[Πληκτρολογήστε κείμενο]

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΠΟΛΙΤΙΣΜΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΣ

Πτυχιακή εργασία: «Σύγκριση των συστημάτων ασφαλείας και ανάλυση των παροχών και των ενεργειών προστασίας χρηστών ανάμεσα στα λειτουργικά συστήματα Android και iOS »

Τσογγίδης Χρήστος
Εξάμηνο:Z
Επιβλέπων καθηγητής: Καλλονιάτης Χρήστος

Ακαδημαϊκό Έτος 2014-2015

Ευχαριστίες

Νιώθω την ανάγκη να εκφράσω την ευγνωμοσύνη μου σε όλα αυτά τα άτομα που είτε άμεσα είτε έμμεσα βοήθησαν στην ολοκλήρωση της πτυχιακής αυτής. Οφείλω, πρωτίστως να ευχαριστήσω από καρδιάς τον επιβλέπων καθηγητή μου κ. Καλλονιάτη Χρήστο, για την βοήθεια και την πρωτοφανή για μένα εμπιστοσύνη. Στο μυαλό μου θα είναι πάντα ο μέντορας μου, ο άνθρωπος που με έκανε να αγαπήσω την ασφάλεια. Μου υπέδειξε μέσα από τις γνώσεις του, τις προτάσεις του και τον ενθουσιασμό του, τι σημαίνει να αγαπάς αυτό που κάνεις, υλοποιώντας το, πάντα με το καλύτερο τρόπο.

Επίσης, θα ήθελα να εκφράσω την αγάπη μου και την ευγνωμοσύνη μου στον πατέρα μου Παναγιώτη, στην μητέρα μου Περιστέρα και στην αδερφή μου Ελισάβετ, για την αμέριστη συμπαράσταση τους, το συνεχές ενδιαφέρον τους και την οικονομική και ψυχολογική ενίσχυση στην προσπάθεια μου να πετύχω και να εκπληρώσω τα όνειρα μου, με πρώτο βήμα της εκπλήρωση της παρούσας πτυχιακής. Χάρη σε αυτούς, γράφω όσα γράφω.

Τον θαυμασμό μου και την ευγνωμοσύνη μου θα ήθελα να εκφράσω στον συμφοιτητή, φίλο, αδερφό και ανταγωνιστή μου κ. Μαυροφίδη Εμμανουήλ. Χάρη στις απεριόριστες ώρες συζήτησης, καθοδήγησης και υποστήριξης όλο αυτό τον καιρό, συνέβαλε με τον τρόπο του στην υλοποίηση της παρούσας πτυχιακής. Τον ευχαριστώ από καρδιάς για την ερευνητική εμπειρία που μου μετέδωσε και για την υπομονή και την κατανόηση που υπέδειξε στις πολλές φορές ανόητες ερωτήσεις μου. Του εύχομαι ότι καλύτερο στην νέα του αρχή και ευελπιστώ στο μέλλον να συνεργαστούμε επαγγελματικά και να συνεχίσουμε αυτές τις απίστευτες συζητήσεις, με γνώμονα την αγάπη μας για την πληροφορική.

Ένα μεγάλο ευχαριστώ θα ήθελα να πω στον ξάδερφο μου Θοδωρή, που με την βοήθεια του, τις φωνές του, την καθοδήγηση του και την εμπειρία του, μου υπέδειξε πως για να γίνεις ο καλύτερος πρέπει να προσπαθήσεις όσο πιο σκληρά γίνεται και πρώτα απ'όλα να το αποδείξεις στον ίδιο σου τον εαυτό.

Ακόμη, θα ήθελα να ευχαριστήσω τον κ.Παυλογεωργάτο Γεράσιμο, που μου επέτρεψε να χρησιμοποιώ τους υπολογιστές του εργαστηρίου του. Η βοήθεια του ήταν πολύ σημαντική για την εκπλήρωση της πτυχιακής μου.

Κλείνοντας, θα ήθελα να εκφράσω την αγάπη μου και να πω ένα τεράστιο ευχαριστώ, στους φίλους μου και τις φίλες μου, που με τη ψυχολογική τους υποστήριξη με βοήθησαν να ολοκληρώσω την παρούσα πτυχιακή.

Χ.Π.Τσογγίδης

Στην οικογένεια μου

Περιεχόμενα

Περίληψη

Μια από τις σημαντικότερες προκλήσεις στο χώρο της τεχνολογίας και συγκεκριμένα των κινητών συσκευών, είναι η ασφάλεια και η ιδιωτικότητα των χρηστών. Όπως συμβαίνει στις περισσότερες επιστήμες, έτσι και η ασφάλεια των κινητών συσκευών ξεκίνησε ως μια οικοτεχνία. Εξελίχθηκε βέβαια από χόμπι σε μια ισχυρή βιομηχανία. Καμία εφαρμογή και κανένα λογισμικό δε μπορεί, όμως, να είναι απόλυτα ασφαλές. Με την έξαρση των κινητών συσκευών στη παγκόσμια αγορά και με την όλο και περισσότερη χρήση τους, ως αντικαταστάτης των προσωπικών υπολογιστών, δημιουργούνται νέα ερωτήματα και νέα προβλήματα στους ερευνητές ασφαλείας. Οι έξυπνες συσκευές είναι ισχυρές φορητές προσωπικές συσκευές που παρέχουν πολλούς τρόπους επικοινωνίας, αναζήτησης πληροφοριών και ψυχαγωγίας. Τα πιο δημοφιλές λογισμικά της αγοράς παγκοσμίως, είναι το Android της Google και το iOS της Apple. Η αρχιτεκτονική των δύο συσκευών βέβαια διαφέρει όσο και το επιχειρηματικό τους μοντέλο. Η Google από την μια, παρέχει δωρεάν υπηρεσίες και έχει κέρδος μέσω των διαφημίσεων που εμπεριέχονται μέσα σε αυτές. Η Apple από την άλλη, χρησιμοποιεί ένα κλειστό μοντέλο, όπου το iOS είναι εσώκλειστο μέσα στις i συσκευές της ομώνυμης εταιρίας. Ακόμη, ο τρόπος με τον οποίο οι δύο εταιρίες προσεγγίζουν την ασφάλεια και τις εφαρμογές για το λογισμικό τους διαφέρει. Η μελέτη των δύο αυτών εταιριών, διαφέρει υποδεικνύοντας τα ακόλουθα προβλήματα-ερωτήματα: α) Μπορεί ο χρήστης να ελέγξει τις εφαρμογές που χρησιμοποιεί ως προς τα δεδομένα που αξιοποιούν; β) Μια εφαρμογή που δεν λειτουργεί σωστά και σύμφωνα με τις προδιαγραφές των δύο εταιριών θα επηρεάσει και τις υπόλοιπες εφαρμογές; γ) Μπορεί ο χρήστης να εμπιστευτεί τους προγραμματιστές των ψηφιακών καταστημάτων των δύο εταιριών; δ) Έιναι τα δεδομένα του χρήστη ασφαλές αν η συσκευή του υποστεί hack, την χάσει ή την κλέψουνε; ε) Μπορεί ο χρήστης να προστατέψει την συσκευή του ενάντια σε μη εξουσιοδοτημένους χρήστες;

Στα πλαίσια της παρούσας πτυχιακής:

- Αναλύθηκε η φιλοσοφία, το επιχειρησιακό μοντέλο, και τα γενικά προβλήματα που αντιμετωπίζουν οι δύο εταιρίες.
- Αναλύθηκαν τα δύο λογισμικά σε βάθος καθώς και οι τεχνικές ασφαλείας που χρησιμοποιούν.
- Αναλύθηκε το μοντέλο UNIX από το οποίο τα δύο λογισμικά έχουν υιοθετήσει τους μηχανισμούς ασφαλείας τους.
- Αναλύθηκαν οι αλλαγές που έχουν κάνει οι δύο εταιρίες σε μηχανισμούς που υιοθετήθηκαν από το μοντέλο UNIX.
- Αναφέρθηκαν προβλήματα που μπορούν να δημιουργηθούν από επιθέσεις μη εξουσιοδοτημένων χρηστών και τους τρόπους με τους οποίους οι δύο εταιρίες έχουν μεριμνήσει για την ασφάλεια των εταιριών.
- Παρουσιάστηκε πίνακας που συγκρίνει τα δύο λογισμικό ως προς τις τεχνικές ασφαλείας τους.
- Αναπτύχθηκε πίνακας που συγκρίνει τα δύο λογισμικά ως προς την παροχή μεθόδων ασφαλείας των χρηστών στο ηλεκτρονικό κατάστημα των δύο εταιριών.
- Παρουσιάστηκε συγκριτικός πίνακας με τα προβλήματα αναβαθμίσεων των δύο εταιριών.
- Παρουσιάστηκε πίνακας σύγκρισης των δύο λογισμικών ως προς το πεδίο επίθεσης των δύο εταιριών από κακόβουλες επιθέσεις τρίτων.

- Παρουσιάστηκε ενιαίος πίνακας σε μορφή infographic με σκοπό την γρήγορη και άμεση ενημέρωση των χρηστών ως προς την ασφάλεια των δύο λειτουργικών συστημάτων.

Abstract

One of the most important challenges in the area of technology, and in particular of mobile devices, is the security and privacy of users. As in most scientific fields, the safety of mobile devices began as a cottage industry, but it certainly evolved from a hobby into a strong industry. However, no application and no software can be perfectly safe. With the mobile devices outbreak in the global market and their increasingly use as a replacement of personal computers, new questions have been created and new problems have arisen for security researchers.

Smart devices are powerful portable personal devices that provide multiple modes of communication, searching for information and entertainment. The most popular software in the market worldwide is Google's Android and iOS of Apple. The architecture of the two devices is of course different as well as their business model. Google on the one hand, provides free services and makes a profit through the advertisements contained within them. Apple on the other, uses a closed model where iOS is enclosed within the iDevices of the homonymous company. In addition, the way in which the two companies approach security and applications for their software differs.

The study of these two companies differs indicating the following problems-questions: a) Can the user control the applications used with regard to the data that he/she takes full advantage of? b) An application that is not working properly and according to the specifications of the two companies will it also affect the rest of the applications? c) Can the user trust the developers of digital stores of both companies? d) Is the data secure if the user's device is hacked, lost or stolen? e) Can the user protect his/her device against unauthorized users?

As part of this dissertation I:

• Analyzed the philosophy, the business model, and the general problems faced by both companies.
• Analyzed both software thoroughly as well as the security techniques used.
• Analyzed the UNIX model from which the two software have adopted their security mechanisms.
• Analyzed the changes that have been made by both companies to mechanisms that were adopted by the UNIX model.
• Reported problems that may arise from attacks by unauthorized users and the ways in which the two companies have ensured safety chains.
• Presented a table comparing the two software regarding their safety techniques.
• Developed a table comparing the two software as regard the provision of security methods of users in the online store of the two companies.
• Presented a comparative table with the problems on upgrades of both companies.
• Presented a comparative table of the two software on the scope of attack of the two companies from malicious attacks from third parties.
• Presented a single table in the format of 'infographic' in order to quickly and directly inform users as to the safety of the two operating systems.

Ευρετήριο Πινάκων

Ευρετήριο Σχημάτων

Ευρετήριο Εικόνων

1.Εισαγωγή

1.1.Οριοθέτηση του προβλήματος

Οι σημερινές κινητές συσκευές επικοινωνίας όπως τα έξυπνα κινητά, τα Tablet καθώς και οι καινούργιες στην αγορά, συσκευές που φοριούνται στα χέρια ή σαν γυαλιά, τα λεγόμενα wearable's, είναι κάτι παραπάνω από συσκευές επικοινωνίας. Είναι μικρού μεγέθους φορητοί προσωπικοί υπολογιστές με περισσότερη ελεύθερη μνήμη και επεξεργαστική ισχύ σε σχέση με τους φορητούς υπολογιστές (Laptop), κοιτώντας στο παρελθόν. Μπορεί να ειπωθεί ότι είναι πλέον κομμάτι της καθημερινής ζωής των ανθρώπων, είτε αυτή διαχωρίζεται σε προσωπική, είτε σε επαγγελματική. Οι πληροφορίες που παρέχουν είναι τόσο κρίσιμες για την καθημερινότητα των ανθρώπων, που πλέον ο καθένας μας έχει τουλάχιστον τρείς συσκευές που επικοινωνούν μεταξύ τους αλλά και με το Διαδίκτυο, ενώ χρησιμοποιούμαι πάνω από έξι online υπηρεσίες καθημερινώς. Ωστόσο, με την αύξηση της χρήσης τέτοιου είδους συσκευών, εγείρουν νέα ζητήματα αλλά και κίνδυνοι ασφαλείας, λόγω της αυξημένης χρήσης πληροφοριών, των νέων αναγκών που προκύπτουν από την συνεχόμενη εξέλιξη της τεχνολογίας καθώς και των απαιτήσεων των χρηστών που ανακύπτουν ως αποτέλεσμα των νέων αγορών που δημιουργούνται από την συνεχόμενη ζήτηση, χρήση και αποθήκευση πληροφοριών. Οι κίνδυνοι που εγείρουν, εκπίπτουν από την μεγάλου όγκου αποθήκευση πληροφοριών στις συσκευές αυτές, δημιουργώντας κινδύνους σε περίπτωση απώλειας της συσκευής, με αποτέλεσμα τον κίνδυνο έκθεσης ευαίσθητων προσωπικών δεδομένων. Οι κίνδυνοι τύπου malware έχουν αυξηθεί μιας και οι συσκευές αυτές συνδέονται απευθείας στο Διαδίκτυο χωρίς τοίχους προστασίας και συστήματα ασφαλείας μεγάλων επιχειρήσεων. Σήμερα, η ασφάλεια των κινητών συσκευών εστιάζεται στον έλεγχο της πρόσβασης μέσω της χρήσης των κλειδαριών συσκευής (Password Codes), καθώς και της συνεχόμενης κρυπτογράφησης δεδομένων τόσο από το λογισμικό όσο και από το Hardware. Ενώ αυτό μπορεί να είναι επαρκής για τους μεμονωμένους χρήστες, είναι ανεπαρκής για την άμυνα απέναντι στους κινδύνους που ενέχουν αναλογικά με τον όγκο δεδομένων και τους χρήστες που ενεργοποιούν καθημερινά μια τέτοια έξυπνη συσκευή. Οι κίνδυνοι όμως, δε σταματούν εδώ. Πολλές από τις επιθέσεις που απειλούν τον προσωπικό υπολογιστή όπως το social-engineering που παρακάμπτουν τα αντι-ιομορφικά συστήματα, λειτουργούν και στις νέες αυτές κινητές συσκευές, καθώς ο ιδιόμορφος τρόπος επίθεσης τέτοιου είδους, δηλαδή μέσω της εξαπάτησης του χρήστη, συνεχώς αυξάνεται, αυξάνοντας παράλληλα και τις επιθέσεις που γίνονται καθημερινά. Παράδειγμα τέτοιας επίθεσης αποτελεί η εξαπάτηση του χρήστη στο να εγκαταστήσει

στην συσκευή του κακόβουλες εφαρμογές. Το πεδίο επίθεσης βέβαια είναι πολύ μεγαλύτερο πλέον, μιας και οι φορητές συσκευές μπορούν όχι μόνο να συνδεθούν στο Διαδίκτυο, προκαλώντας σε περίπτωση επιτυχίας της επίθεσης την συλλογή προσωπικών πληροφοριών συνυπολογίζοντας και την τοποθεσία, αλλά παράλληλα και την χρέωση του λογαριασμού του χρήστη στέλνοντας μηνύματα είτε καλώντας κάποιων αριθμό που χρεώνει. Ακόμη μπορεί να βλάψει μια ολόκληρη επιχείρηση με την έκθεση ηλεκτρονικών μηνυμάτων και γενικότερα ευαίσθητων δεδομένων. Ο ρόλος των κινητών συσκευών και των κοινωνικών δικτύων στις επαναστάσεις στην Αίγυπτο, τη Λιβύη και τη Συρία, ήταν καθοριστικός, με επιθέσεις κακόβουλου λογισμικού και ιούς που απευθύνονται σε συλλογή πληροφοριών καθώς και επιθέσεις τύπου Denial of Service δημιουργούν μεγαλύτερο βάρος στους ώμους των ερευνητών ασφαλείας (Enck, W. : 2011).

1.2.Στόχος της πτυχιακής

Η πτυχιακή αυτή επικεντρώνεται κυρίως στους δύο κολοσσούς των έξυπνων φορητών συσκευών, της Google με το λογισμικό Android και της Apple με τον λογισμικό iOS. Θα ερευνηθεί η ιστορία και η φιλοσοφία των δύο αυτών εταιριών με σκοπό να αντιληφθεί ο απλός χρήστης την οπτική γωνία ασφάλειας που προσεγγίζει η κάθε εταιρία σύμφωνα με το επιχειρηματικό σχέδιο (Business plan) [1]της. Έπειτα θα ελεγχτούν εξονυχιστικά τα μοντέλα ασφαλείας των δύο εταιριών και των στοιχείων που υιοθετούν από το μοντέλο UNIX. Εν κατακλείδι, θα γίνει μια αποτίμηση των μοντέλων αυτών και θα παρατεθεί το ένα μοντέλο ασφαλείας απέναντι στο άλλο ώστε να βγει ένα συμπέρασμα για το πιο θεωρείται το πιο ασφαλές λειτουργικό σύστημα στην αγορά των έξυπνων κινητών συσκευών ανάμεσα στα δύο.

[1] Business plan: Το επιχειρηματικό σχέδιο ορίζεται ως η επίσημη δήλωση των επιχειρηματικών στόχων μιας επιχείρησης, τους λόγους για τους οποίους θεωρεί ότι είναι εφικτοί, και τα σχέδια για την επίτευξή τους. Μπορεί επίσης να περιέχει βασικές πληροφορίες σχετικά με την οργάνωση ή την ομάδα που προσπαθεί να επιτύχει αυτούς τους στόχους.

2. Πλατφόρμα Android

Η λέξη Android χρησιμοποιείται αρκετά στις μέρες μας και με ποικίλους τρόπους. Παρόλο που η λέξη παραπέμπει ακόμη στο ανθρωποειδές ρομπότ, η χρήση της λέξης σημαίνει πολύ περισσότερα σε σχέση με την προηγούμενη δεκαετία. Στον κόσμο των κινητών-φορητών συσκευών, η χρήση του όρου Android παραπέμπει σε μια εταιρία, σε ένα λειτουργικό σύστημα, σε ένα πρότζεκτ ελεύθερου ανοιχτού κώδικα και εν τέλει σε μια ανοιχτή κοινότητα προγραμματιστών με όραμα για μια ελεύθερη και ανοιχτή στο κοινό αγορά λειτουργικών συστημάτων. Εν ολίγοις, ένα τεράστιο οικοσύστημα που περιβάλλει τις κινητές μας συσκευές και όχι μόνο.

2.1 Κατανοώντας τις ρίζες της πλατφόρμας Android

Πολλοί θεωρούν ότι η πλατφόρμα Android έγινε η πιο γνωστή και ευρέως διαδεδομένη πλατφόρμα εν μια νυχτί, η αλήθεια όμως είναι ότι η πλατφόρμα καθώς και όλη η κοινότητα που την περιβάλλει έχουν περάσει από πολλούς σκοπέλους την τελευταία δεκαετία. Ο όρος Android που χρησιμοποιούμε για το λειτουργικό μας σύστημα ξεκίνησε ως όνομα εταιρίας πριν από έντεκα χρόνια, για την ακρίβεια τον Οκτώβριο του 2003 με ιδρυτές τους Andy Rubin, Chris White, Nick Sears και Rich Miner. Σκοπός της εταιρίας, στα πρώιμα στάδια του, ήταν να σχεδιάσουν και να κατασκευάσουν ένα κινητό με ένα λειτουργικό σύστημα που θα μπορούσε να δέχεται πληροφορίες τοποθεσίας, καθώς και τις προτιμήσεις του χρήστη, συνυφασμένα σε ένα προσωπικό λογαριασμό χρήστη. Το 2005 όμως η εταιρία Google, χωρίς την αίγλη που έχει σήμερα, ξεπερνώντας τις δυσκολίες γραφειοκρατίας με το κράτος της Αμερικής και τις ρήτρες ανταγωνισμού, εξαγόρασε την Android A.E. Στη περίοδο που ακολούθησε, η Google άρχισε να ψάχνει για συνεργάτες και να κλείνει συνεργασίες με κατασκευάστριες εταιρίες υλικού (Hardware) και λογισμικού (Software) καθώς και με εταιρίες τηλεπικοινωνιών με σκοπό να ενταχθεί στο τραπέζι της αγοράς των κινητών συσκευών (Drake, J. :2014). Το Νοέμβριο του 2007 ιδρύθηκε η συμμαχία, όπως ονομάζεται στα Ελληνικά, ανοιχτού ακουστικού (Open Handset Alliance[2]) η αλλιώς OHA. Αυτή η συσπείρωση των εταιριών, τριάντα τεσσάρων για την ακρίβεια υπό την ηγεσία της Google, είχε δημιουργήσει ένα μεγάλο κοινό μεταξύ τους (Enck, W. : 2011). Την δέσμευση

[2] OHA (Open Handset Alliance): Το Open Handset Alliance (OHA) είναι μια κοινοπραξία ογδόντα-τεσσάρων επιχειρήσεων για την ανάπτυξη ανοικτών προτύπων για τις κινητές συσκευές. Οι εταιρίες μέλη που συμπεριλαμβάνονται στη συμμαχία είναι η Google, HTC, Sony, Dell, Intel, Motorola, Qualcomm, Texas Instruments, η Samsung Electronics, LG Electronics, η T-Mobile, η Sprint Corporation, η Nvidia, και Wind River Systems.

στην ελεύθερη και ανοιχτή κοινότητα[3]. Επίσης, στόχος της συμμαχίας ήταν να επιταχύνουν τις καινοτομίες στα πλαίσια των κινητών συσκευών, επιτυγχάνοντας συνάμα και μια πιο πλούσια εμπειρία χρήσης για τον χρήστη, σε τιμές προσιτές προς το κοινό. Το 2013 η συμμαχία πλέον έχει ογδόντα τέσσερα μέλη. Η Google έχοντας επιτύχει το στόχο της να δημιουργήσει συμμαχίες, παρουσιάζει για πρώτη φορά το πρώτο της λειτουργικό σύστημα για φορητές συσκευές με το όνομα Android. Παρόλα αυτά, ακόμη δεν παρουσίασε κάποια συσκευή που να χρησιμοποιεί το λειτουργικό της σύστημα. Πέντε χρόνια μετά την εξαγορά της εταιρίας Android, και για την ακρίβεια το 2008, η Google παρουσιάζει και δίνει το λειτουργικό της σύστημα στο ευρύ κοινό. Την πρώτη συσκευή που χρησιμοποιεί το λειτουργικό σύστημα Android, παρουσίασε στις αγορές, το Δεκέμβριο του 2008, η HTC, με κωδικό μοντέλου G1, σηματοδοτώντας μια καινούργια εποχή στις φορητές μας συσκευές. Για να γίνει περισσότερο κατανοητή η φιλοσοφία του λειτουργικού συστήματος Android και στην πορεία να μελετηθούν τα τεχνικά χαρακτηριστικά του καθώς και η ασφάλεια που το περιβάλλει προς όφελος όλως θα γίνει παρακάτω μια ιστορική αναδρομή στα γεγονότα που έκαναν την πλατφόρμα, την πιο διαδεδομένη πλατφόρμα στο κόσμο έως και σήμερα. Αρχικά, πριν την πρώτη επίσημη εμπορική έκδοση του λειτουργικού Android, η πλατφόρμα είχε Alpha και Beta[4] εκδόσεις. Οι εκδόσεις Alpha ήταν διαθέσιμες μόνο για τα μέλη της Google και της συμμαχίας OHA και είχαν κωδικούς ονόματα διάσημων ρομπότ όπως, Astroboy, Bender και του ρομπότ από την ταινία ο πόλεμος των άστρων R2-D2. Η Beta έκδοση κυκλοφόρησε στις 5 Νοεμβρίου του 2007 ημερομηνία που θεωρείται έως και σήμερα ως τα γενέθλια της πλατφόρμας Android. Η πρώτη εμπορική έκδοση (η έκδοση 1.0), κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008 και η επόμενη αναβαθμισμένη έκδοση 1.1 ήταν διαθέσιμη στις 9 Φεβρουαρίου του 2009. Οι παραπάνω δύο εκδόσεις ήταν οι μόνες χωρίς κωδικό όνομα. Στις 30 Απριλίου του 2009, κυκλοφόρησε η έκδοση 1.5 κι έκτοτε, κάθε έκδοση είχε κωδική ονομασία από διάφορα γλυκά, αλφαβητικά (Drake, J. : 2006).

[3] Ανοιχτή κοινότητα (Open Source Community): Ο ανοιχτός κώδικας ως ένα μοντέλο ανάπτυξης προωθεί την καθολική πρόσβαση μέσω της ελεύθερης άδειας για το σχεδιασμό ή προσχέδιο ενός προϊόντος και παράλληλα την καθολική αναδιανομή του εν λόγω σχεδίου ή προσχεδίου, συμπεριλαμβανομένης της περαιτέρω βελτιώσεις της από οποιονδήποτε.

[4] Alpha-Beta:Δοκιμαστικές εκδόσεις του λειτουργικού συστήματος στα πρώιμα στάδια. Είναι σύνηθες αυτές οι εκδόσεις να μην είναι διαθέσιμες στο ευρύ κοινό αλλά μόνο στα εργαστήρια όπου δοκιμάζονται ή σε μια μικρή ομάδα δοκιμαστών (Alpha και Beta testers).

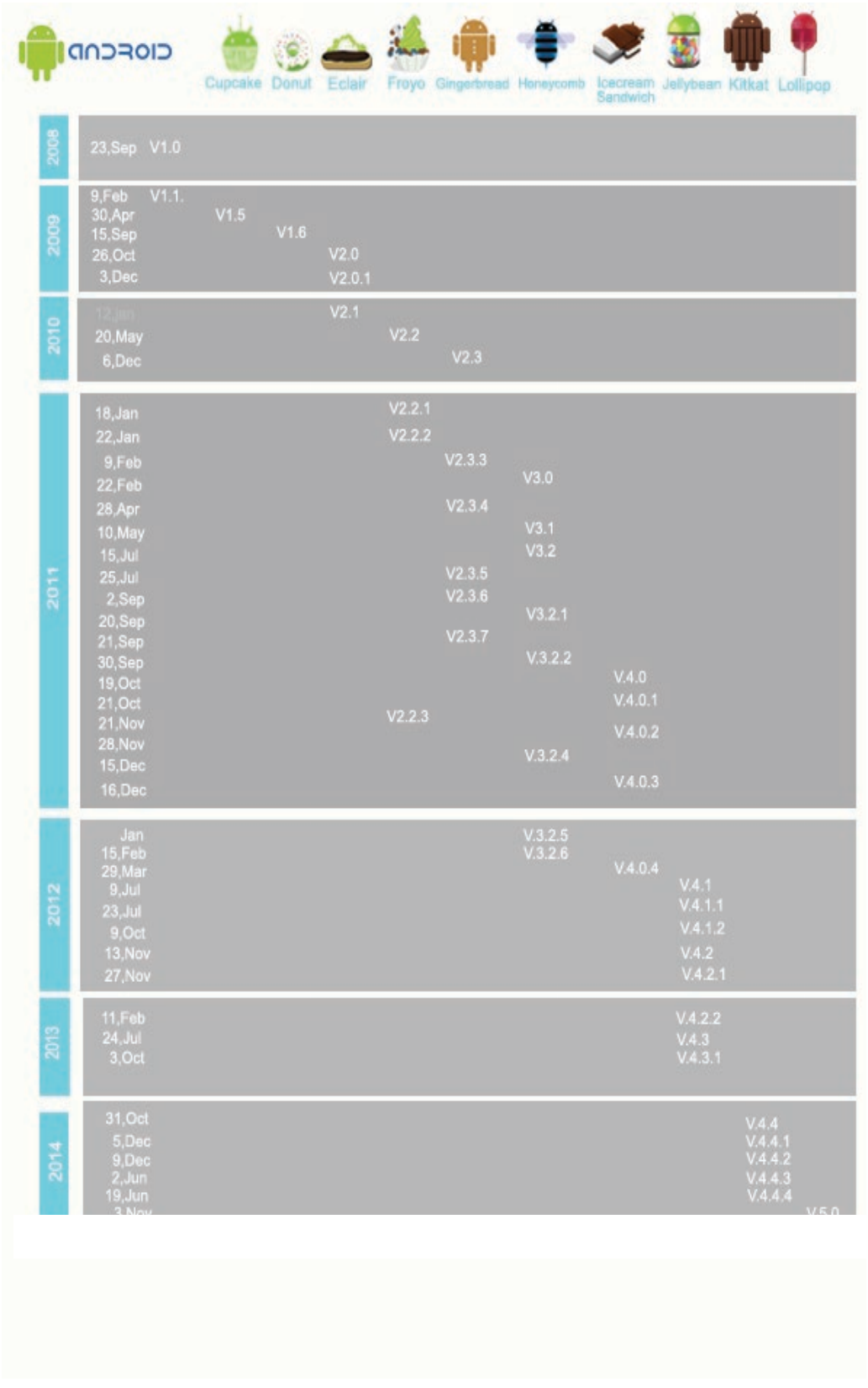
2.2. Εξερευνώντας τις συσκευές που χρησιμοποιούν Android

Με την συνεχώς αυξανόμενη ανάπτυξη του λειτουργικού συστήματος Android, έχει επέλθει και η ταχεία ανάπτυξη συσκευών που χρησιμοποιούν το λειτουργικό αυτό σύστημα. Το λειτουργικό σύστημα σήμερα δεν έχει βέβαια, τους ίδιους στόχους με το 2008 και παρατηρείται τάση εξάπλωσης της πλατφόρμας πέρα από τις παραδοσιακές συσκευές, κινητά και Tablet, σε διάφορες συσκευές όπως την τηλεόραση, τα έξυπνα ρολόγια, δορυφόρους στο διάστημα, τα έξυπνα γυαλιά της Google, καθώς και πρόσφατα την χρήση του λειτουργικού συστήματος σε αυτοκίνητα (Google I/O : 2014). Αναφορικά μερικές από τις κατασκευάστριες εταιρίες συσκευών που χρησιμοποιούν το λογισμικό Android μέχρι και σήμερα είναι η LG, Samsung, HTC, Motorola, Lenovo, Alcatel, Oneplus, Huawei, Oppo, Asus, Gigabyte, Sony, Zte, Nokia (σε κάποιες συσκευές της), η Ελληνική MLS, κ.ο.κ.

2.3 Οι εκδόσεις και οι υποεκδόσεις της πλατφόρμας Android

Στην προηγούμενη ενότητα αναφέρθηκε ότι κάθε έκδοση του λειτουργικού συστήματος Android ονομάζεται με βάση κάποιο γλυκό διακρίνοντας την μία έκδοση από την επόμενη με κριτήριο το πρώτο γράμμα της αλφαβήτου, ξεκινώντας από το γράμμα το Α. Σήμερα, το λειτουργικό σύστημα Android βρίσκεται στην έκδοση 5.0. Lollipop, στα Ελληνικά «γλειφιτζούρι». Όπως ονομάζει η Google τις διάφορες μεγάλες εκδόσεις που κυκλοφορεί στην αγορά, έτσι ονομάζει και τις υποεκδόσεις της. Δηλαδή, τις εκδόσεις που βγαίνουν στα ενδιάμεσα των μεγάλων αναβαθμίσεων με σκοπό τη διόρθωση κάποιου κενού ασφαλείας, λειτουργικών προβλημάτων κ.ο.κ. .Για παράδειγμα, ο κωδικός JOP40D, εξηγείται ως εξής. Το πρώτο γράμμα αντικατοπτρίζει την μεγάλη έκδοση που αυτή τη στιγμή είναι διαθέσιμη στην αγορά, ή την έκδοση που αυτή τη χρονική στιγμή χρησιμοποιεί η συσκευή του εκάστοτε χρήστη (J για την έκδοση Jellybean). Το δεύτερο γράμμα αντικατοπτρίζει τον κωδικό της ομάδας που δούλεψε στην εκάστοτε αναβάθμιση. Το τρίτο γράμμα καθώς και οι επόμενοι δύο αριθμοί αντικατοπτρίζουν μια ημερομηνία. Το τελικό γράμμα αντικατοπτρίζει την περίοδο που χρόνου που δημιουργήθηκε η έκδοση. Ουσιαστικά, η Google διαχωρίζει το χρόνο σε τέσσερις περιόδους. Ιανουάριο-Φεβρουάριο-Μάρτιο ως Α τρίμηνο, Απρίλιο-Μάιο-Ιούνιο ως Β τρίμηνο, Ιούλιο-Αύγουστο-Σεπτέμβριο ως Γ τρίμηνο και Οκτώβριο-Νοέμβριο-Δεκέμβριο ως Δ τρίμηνο. Στο παραπάνω παράδειγμα το γράμμα P αντιπροσωπεύει το τέταρτο τρίμηνο του 2012. Τα δύο νούμερα (40) αντικατοπτρίζουν τις μέρες από την αρχή του εκάστοτε τριμήνου. Το P40 δηλαδή αποκωδικοποιείται στις 10 Νοεμβρίου του 2012

(Drake, J. :2014). Παρακάτω παραθέτεται ο πίνακας με όλες τις εκδόσεις Android από την Έκδοση 1.0 που προαναφέρθηκε έως και την τελευταία έκδοση Android 5.0. Lollipop που παρουσιάστηκε στις 16 Οκτωβρίου του 2014.



Πίνακας 1: *Όλες οι αναβαθμίσεις του λειτουργικού Android ανά χρονολογική σειρα.*

2.4 Σειρά Nexus

Με τον όρο Nexus ορίζεται η σειρά κινητών συσκευών και Tablet υπό την αιγίδα της Google, οι οποίες θεωρούνται ως ναυαρχίδες συσκευές στην κατηγορία τους. Κάθεσυσκευή κατασκευάζεται από διαφορετικούς κατασκευαστές (OEM's[5]) σε στενή σχέσηβέβαια με την Google. Οι συσκευές αυτές πωλούνται με ξεκλείδωτες Sim, δηλαδή μπορούν να χρησιμοποιηθούν με διαφορετικές Sim εταιριών τηλεπικοινωνιών άλλων χωρών, δίνοντας την δυνατότητα σε χρήστες που ταξιδεύουν σε διαφορετικές χώρες με διαφορετικό GSM[6], να αξιοποιούν την υπάρχων συσκευή τους. Επίσης, οι συσκευές αυτές πωλούνται και μέσα από το ηλεκτρονικό κατάστημα της Google, το Google playstore. Η πρώτη συσκευή του προγράμματος Nexus δημιουργήθηκε από την HTC, με το όνομα Nexus ONE, το οποίο παρουσιάστηκε επίσημα στην αγορά τον Ιανουάριο του 2010 και χρησιμοποιούσε το λογισμικό Android 2.1. Éclair. Το συγκεκριμένο μοντέλο ήταν το πρώτο κινητό που αναβαθμίστηκε το λογισμικό του στο τότε καινούργιο Android 2.2. Froyo, το Μάιο του 2010. Η Google, όπως προαναφέρθηκε, συνεργάστηκε με πολλούς κατασκευαστές, για παράδειγμα, Samsung, LG, HTC και την Asus για την κατασκευή κινητών και Tablet. Παρακάτω παραθέτονται όλες οι συσκευές Nexus μέχρι και σήμερα.

Εικόνα 1: *Όλα τα Tablet και Smartphone από την σειρά Nexus της Google*

[5] OEM: (Original equipment manufacturer) ονομάζεται μια επιχείρηση, συνήθως του κλάδου της πληροφορικής , η οποία αγοράζει τα βασικά μέρη ενός υπολογιστή , κινητού κ.ο.κ. έτοιμα και τα συνθέτει με σκοπό την κατασκευή και πώληση ενός ολοκληρωμένου συστήματος τον οποίο πουλάει μαζί με άλλες υπηρεσίες όπως εγγύηση, υποστήριξη, εγχειρίδια χρήσης κ.τ.λ.

[6] GSM: (Global System for Mobile Communications) είναι ένα κοινό Ευρωπαϊκό σύστημα κινητής τηλεφωνίας. Το GSM είναι ένα κυψελοειδές ψηφιακό σύστημα κινητής τηλεφωνίας δεύτερης γενιάς (2G), το οποίο χρησιμοποιεί ηλεκτρομαγνητικά σήματα και την τεχνική πολλαπλής πρόσβασης με διαχωρισμό του διαθέσιμου φάσματος συχνοτήτων σε ένα αριθμό καναλιών και την διαίρεση αυτών σε χρονοθυρίδες για την μετάδοση σημάτων.

Η σειρά Nexus όμως, δεν είναι απλές συσκευές με σκοπό το κέρδος των εταιριών και της Google, αλλά το σχέδιο και τα χαρακτηριστικά που θεωρεί η Google ότι πρέπει να υφίστανται σε μια κινητή συσκευή σήμερα (Google I/O:2014). Οι συσκευές Nexus είναι επίσης το σημείο κατατεθέν, για τις νέες εκδόσεις Android που προωθεί στην αγορά η Google. Οι συσκευές αυτές αναβαθμίζονται κατευθείαν από την Google, λίγο καιρό μετά την παρουσίαση του λογισμικού. Η πλατφόρμα Android λειτουργεί ως πλατφόρμα ανοιχτού κώδικα για τους προγραμματιστές. Οι συσκευές αυτές διατίθενται στην αγορά με ξεκλείδωτο bootloader[7], επιτρέποντας στους προγραμματιστές να μετατρέπουν και να δημιουργούν διάφορες παραλλαγές του λειτουργικού συστήματος Android (για παράδειγμα Cyanogenmod) που μπορούν να τις χρησιμοποιούν. Ακόμη, η Google, στις συγκεκριμένες συσκευές παρέχει εικονικό backup λειτουργικό ταυτόχρονα με το αυθεντικό ή pure[8] λειτουργικό σύστημα της, επιτρέποντας στο χρήστη-προγραμματιστή να επαναφέρει στην πρώιμη κατάσταση το κινητό, όπως δηλαδή το αγόρασε αφαιρώντας όλες τις ρυθμίσεις (tweaks) που μπορεί να είχε αλλάξει. Τελευταίο σπουδαίο χαρακτηριστικό των συσκευών αυτών είναι, ότι χρησιμοποιούν το γνήσιο λογισμικό της Google, προσφέροντας την εμπειρία χρήσης στον χρήστη όπως η Google τη σχεδίασε, τη δοκίμασε και τη παρουσίασε στο κοινό στο Google I/O (Drake, J. : 2006).

2.5 Αναβάθμιση συσκευών στο οικοσύστημα Android

Η αναβάθμιση συσκευών στο οικοσύστημα Android για τις συσκευές nexus γίνεται όπως προαναφέρθηκε σε διάρκεια δεκαπέντε έως τριάντα μέρες από την μέρα παρουσίασης της αναβάθμισης και της διάθεσης της στο ευρύ κοινό. Η Google έχει επιλέξει πλέον να κάνει τις αναβαθμίσεις λογισμικού, παραδείγματος χάρη από την έκδοση Kitkat 4.4 στην έκδοση Lollipop 5.0, που είναι και η τελευταία, διαμέσου του διαδικτύου ή όπως λέγεται στα αγγλικά on the air (OTA). Η Google όμως, πέρα από την αναβάθμιση του λειτουργικού συστήματος, αναβαθμίζει σε μικρά χρονικά διαστήματα και τις μεμονωμένες εφαρμογές-υπηρεσίες που παρέχει στον χρήστη όπως την Gmail εφαρμογή της, χρησιμοποιώντας στην περίπτωση αυτή το ηλεκτρονικό της κατάστημα. Στο σημείο αυτό ο χρήστης, έχει την επιλογή να ορίσει ένα πλήθος εφαρμογών που έχει εγκαταστήσει από το Google play store, όπου αναβαθμίζονται αυτόματα κάθε φορά που

[7] Bootloaders: Ένα μικρό κομμάτι λογισμικού που σκοπό έχει την προετοιμασία της συσκευής για την φόρτωση και εκκίνηση του λογισμικού της συσκευής. Συνήθως ελέγχει την συσκευή για σφάλματα σε επίπεδο hardware ενώ παράλληλα ελέγχει την αλυσίδα των ελέγχων που γίνονται στη συσκευή κατά την εκκίνηση τους.

[8] Pure Android:Είναι το λειτουργικό σύστημα που σχεδιάστηκε,υλοποιήθηκε και δοκιμάστηκε από την Google και δεν έχει δεχτεί τροποποιήσεις από OEMs.

είναι συνδεμένη η συσκευή του στο Διαδίκτυο και υπάρχει διαθέσιμη αναβάθμιση. Η ίδια διαδικασία ισχύει για κάθε εφαρμογή που υπάρχει στο ηλεκτρονικό της κατάστημα. Εξαιρούνται οι εφαρμογές που δεν έχουν το πιστοποιητικό της Google και είναι εγκατεστημένες από τον χρήστη διαμέσου του διαδικτύου και εκτός playstore. Η Google, βέβαια στις προκαθορισμένες ρυθμίσεις της αποτρέπει τον χρήστη από το να εγκαταστήσει τέτοιου είδους εφαρμογές. Εφαρμογές, δηλαδή που δεν έχουν ελεγχτεί από την ίδια. Η Google βέβαια, λόγω της στρατηγικής της σύγκλισης (Drake, J. : 2006) προς την ιδεολογία του ελεύθερου πηγαίου κώδικα επιτρέπει εν γνώσει της την εγκατάσταση μίας εφαρμογή που δεν βρίσκεται στο play store, δίνοντας την δυνατότητα στον χρήστη να εγκαταστήσει την εφαρμογή της επιλογής του, την οποία έχει κατεβάσει από οποιοδήποτε τρίτο ηλεκτρονικό κατάστημα εφαρμογών (3rd party application stores), αρκεί να είναι σε κωδικοποίηση Apk[9].

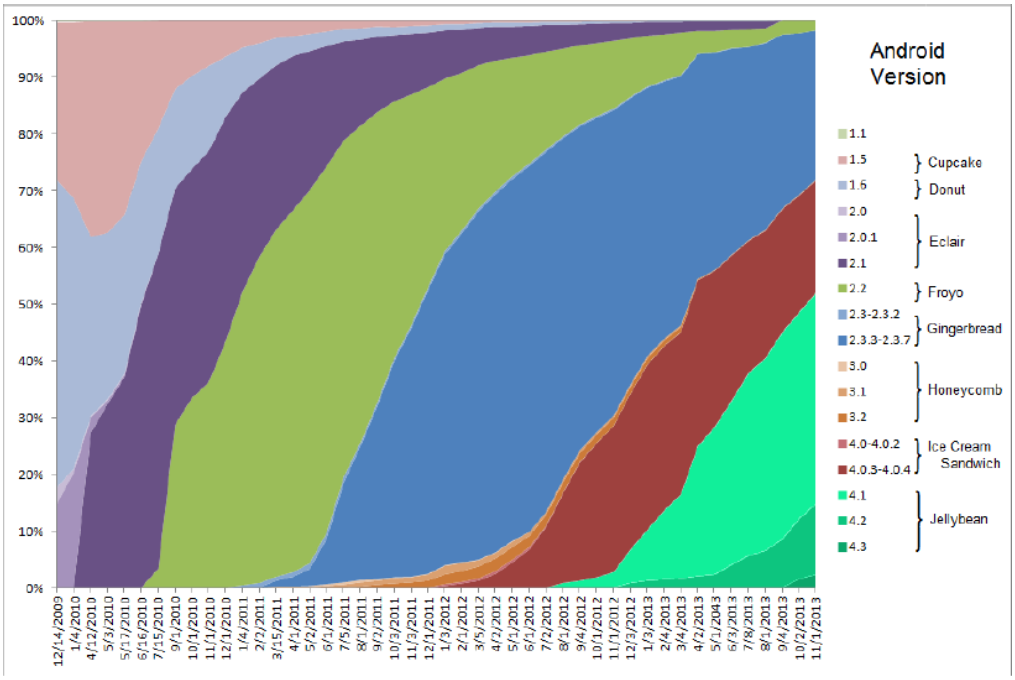
2.6.Προβλήματα ενημέρωσης λογισμικού

Ένα βασικό πρόβλημα όπως αναφέρθηκε και παραπάνω είναι ότι μόνο οι Nexus συσκευές ενημερώνονται σε τακτά διαστήματα από την ίδια την Google. Το πρόβλημα ενέχει στο γεγονός ότι, όταν η Google αντιλαμβάνεται κάποιο κενό ασφαλείας (security breach) ή κάποιο bug[10] στο λογισμικό της, ενημερώνει απευθείας το λογισμικό της, ώστε να αποτρέψει τυχόν επιθέσεις, βελτιώνοντας έτσι και την ποιότητα διάδρασης με το λογισμικό, ενώ παράλληλα μειώνει το φόβο των χρηστών για τυχόν απώλεια ή κλοπή των δεδομένων τους. Με τον τρόπο αυτό δημιουργεί δεσμούς εμπιστοσύνης ανάμεσα στον χρήστη και στην εταιρία παροχής λογισμικού. Το πρόβλημα βέβαια είναι ότι η διαδικασία που προαναφέρθηκε γίνεται μόνο για τις συσκευές Nexus και για κάποιες εκδόσεις κινητών που αναφέρονται ως Google play edition συσκευές, οι οποίες και αυτές με την σειρά τους χρησιμοποιούν το γνήσιο λογισμικό σύστημα της Google. Ο κόσμος όμως δεν αγοράζει μόνο συσκευές Nexus, αλλά αγοράζει στην πλειοψηφία συσκευές άλλων εταιριών που χρησιμοποιούν ένα τροποποιημένο λογισμικό βασισμένο στο Android. Οι εταιρίες αυτές για να αναβαθμίσουν την συσκευή από την επίσημη έκδοση του λογισμικού, κάνουν 90 μέρες και συνήθως αναβαθμίζουν μόνο τις ναυαρχίδες συσκευές τους. Για παράδειγμα η Samsung ανακοίνωσε στις 16 Οκτωβρίου του 2014 ότι θα αναβαθμίσει το λογισμικό του S5 και του Note 4 εντός 90 ημερών από την επίσημη

[9] APK.: (Application) είναι η κωδικοποίηση αρχείου του πακέτου της εφαρμογής που χρειάζεται για να εγκατασταθεί η εφαρμογή στο λειτουργικό android.
[10] Bug: Ένα σφάλμα του λογισμικού. Μπορεί να είναι ένα λάθος, ελάττωμα, αποτυχία, ή ελάττωμα σε ένα πρόγραμμα υπολογιστή ή σε ένα σύστημα που να προκαλεί την παραγωγή μη ορθού ή μη αναμενόμενου αποτελέσματος, ή να συμπεριφέρεται με μη αναμενόμενους τρόπους.

διάθεση του λογισμικού 5.0 Lollipop, αφήνοντας όλες τις υπόλοιπες συσκευές της εταιρίας στο κενό. Παρατηρούμε, ότι το ίδιο μοτίβο αναβαθμίσεων ακολουθούν και οι υπόλοιπες εταιρείες. Βέβαια, πάνω από το 70% των συσκευών που χρησιμοποιούν android σήμερα δεν είναι αναβαθμισμένες στην τελευταία έκδοση Kitkat 4.4, ενώ οι εταιρείες επιμένουν να πουλάνε στην αγορά κινητές συσκευές που χρησιμοποιούν λογισμικό απαρχαιωμένο, για παράδειγμα android 2.2 froyo, και πλήρως απροστάτευτο από κακόβουλες επιθέσεις, με το πρόσχημα ότι είναι φθηνές συσκευές και δε μπορούν να αναβαθμιστούν στην τελευταία έκδοση λόγω δυνατοτήτων σε επίπεδο hardware. Αξίζει να αναφερθεί, ότι η Google πιέζει τις εταιρίες να σταματήσουν να μετατρέπουν το λογισμικό της με δικά του UI[11] και τους προτρέπει να χρησιμοποιούν μόνο το καθαρό λειτουργικό σύστημα που τους προσφέρει, ώστε να αναβαθμίζονται όλες οι συσκευές στην ώρα τους και με το τελευταία διαθέσιμο λογισμικό. Στο παρακάτω γράφημα φαίνεται καθαρά τα όσα αναφέρθηκαν έως τώρα σε επίπεδο αναβαθμίσεων. Τέλος οι προγραμματιστές που δημιουργούν τις εφαρμογές τους και τις διαθέτουν στο playstore, όταν αντιληφθούν κάποιο κενό ασφαλείας, μπορούν να αναβαθμίζουν και τα διορθώνουν τα κενά ασφαλείας στην εφαρμογή τους ταυτόχρονα μέσω του καταστήματος σε όλες τις συσκευές, ανεξαρτήτως του λογισμικού που χρησιμοποιούν.

[11] User Interface (UI): Διεπαφή, διεπιφάνεια ή διασύνδεση (αγγλ. *interface*) ονομάζουμε το σύνορο επικοινωνίας μιας οντότητας (π.χ. το κομμάτι ενός λογισμικού, μια συσκευή υλικού, ένας χρήστης, κτλ.) με το περιβάλλον της. Τείνουμε να ισχυριστούμε ότι η διεπαφή είναι κάτι το αφηρημένο αφού στην ουσία είναι μια περιγραφή του τρόπου με τον οποίο μια οντότητα θα ζητήσει από μια άλλη να επιτελέσει κάποια λειτουργία σε αντίθεση με την ίδια την οντότητα που υλοποιεί την διεπαφή και η οποία συνήθως είναι διαισθητικά πιο 'πραγματική'.



Πίνακας 2: Υιοθέτηση αναβαθμίσεων του λογισμικού από τις συσκευές Android (Creative Commons Attribution-Share Alike 3.0 Unported license[12])

2.7. Back-porting

Με τον όρο Back-Porting ορίζουμε την διόρθωση προβλημάτων που σχετίζονται κυρίως με προβλήματα (bugs) του λογισμικού αλλά και κενά ασφαλείας της τωρινής έκδοσης του λογισμικού σε παλαιότερες εκδόσεις (Drake, J. :2014). Για παράδειγμα η τελευταία γνωστή έκδοση του Android μέχρι τώρα που γράφεται η πτυχιακή αυτή, είναι η έκδοση 5.0.1 kitkat. Εάν έχει υποπέσει στην αντίληψη της Google ότι υπάρχει κάποιο κενό ασφάλειας στην έκδοση 4.0.4 Icecream sandwich, τότε η Google θα διορθώσει το κενό αυτό, καθώς και οποιοδήποτε άλλο πρόβλημα του λογισμικού στην έκδοση 4.4.5 ή 5.0 αφήνοντας απροστάτευτους όσους ακόμη χρησιμοποιούν την έκδοση 4.4.4 και πίσω. Βέβαια, παρόλο που ο φόβος για επιθέσεις σε προβλήματα που αφορούν το Back-porting είναι γνωστές σε όλη την κοινότητα , μέχρι και σήμερα δεν έχει αναφερθεί επίθεση .

2.8. Google Playstore

Το Google playstore είναι το ηλεκτρονικό κατάστημα της Google στο οποίο προσφέρονται εφαρμογές, τόσο της ίδιας της εταιρίας, όσο και τρίτων προγραμματιστών είτε δωρεάν είτε ενός συμβολικού ποσού, προσφέροντας έτσι μια πιο ολοκληρωμένη εμπειρία χρήσης της κινητής μας συσκευής. Διαμέσου του ηλεκτρονικού καταστήματος

[12] *http:// en.wikipedia.org/wiki/File:Android_historical_version_ distribution.png*

προσφέρονται επίσης μικρές αναβαθμίσεις στο λογισμικό καθώς και αναβαθμίσεις τόσο στις εφαρμογές του κινητού μας και παρέχεται η δυνατότητα στο προγραμματιστή μιας εφαρμογής που ανήκει στο κατάστημα, να αναβαθμίζει τις εφαρμογές του με σκοπό την διευκόλυνση του χρήστη. Μοναδική προϋπόθεση για την διάθεση μιας εφαρμογής από τον προγραμματιστή ή την εταιρία είναι να κάνει μια συνδρομή στο πρόγραμμα της Google, πιστοποιώντας τα στοιχεία του και πληρώνοντας το συμβολικό ποσό των τριάντα δολαρίων Αμερικής στην εταιρία. Ο τρόπος που λειτουργεί είναι ιδιαίτερα απλός. Ο χρήστης δημιουργεί ένα κωδικό μέλους στο πρόγραμμα, ανεβάζει την εφαρμογή του, στη πορεία η Google ελέγχει την εφαρμογή για κακόβουλο κώδικα καθώς και την αποδοχή του πρώτου στην πολιτική ορθής χρήσης και παροχής εφαρμογών στο ηλεκτρονικό κατάστημα και έπειτα δίνει ένα μοναδικό κλειδί στον προγραμματιστή της εφαρμογής καθώς και στην ίδια εφαρμογή. Εάν η εφαρμογή περάσει επιτυχώς από όλα τα τεστ της Google τότε «ανεβαίνει» στο κατάστημα και είναι πλέον διαθέσιμη στην μεγάλη κοινότητα του Android.

2.9.Κατασκευαστές CPU

Παρόλο που οι εφαρμογές για το λογισμικό Android είναι εφαρμογές αγνώστου επεξεργαστών δηλαδή μπορούν να «τρέξουν» με οποιαδήποτε συσκευή μιας και οι συσκευές που βρίσκονται στο οικοσύστημα του Android διαφέρουν ως προς το hardware τους, οι εγγενείς εκτελέσιμες βιβλιοθήκες τους είναι κοινές. Οι βιβλιοθήκες αυτές μεταγλωττίζονται για τον εκάστοτε επεξεργαστή της εκάστοτε συσκευής. Το σύστημα Android είναι βασισμένο όπως προαναφέρθηκε στο Linux kernel, το οποίο δε προσφέρει μόνο φορητότητα, αλλά υποστηρίζει και μια πληθώρα επεξεργαστών. Με παρόμοια διαδικασία λειτουργεί και η εγγενής βιβλιοθήκη ανάπτυξης εφαρμογών του Android (Androids Native Development kit -NDK-) προσφέροντας στον προγραμματιστή τη δυνατότητα να δημιουργεί εφαρμογές και όλους τους διαθέσιμους τύπους επεξεργαστών. Λόγω της χαμηλής κατανάλωσης, ο επεξεργαστής βασισμένος στην αρχιτεκτονική Arm είναι πλέον ο πιο διαδεδομένος επεξεργαστής στις κινητές συσκευές. Πρέπει να αναφερθεί ότι οι επεξεργαστές βασισμένες στην αρχιτεκτονική Arm έχουν κατοχυρωμένη μόνο την άδεια πνευματικής ιδιοκτησίας, δίνοντας την δυνατότητα στους διάφορους κατασκευαστές να κατασκευάσουν και να πουλήσουν τους μικροεπεξεργαστές τους ελεύθερα.

2.10. Custom Roms

Όπως προαναφέρθηκε παραπάνω οι εταιρίες κινητών τηλεφώνων σχεδιάζουν και τροποποιούν το λογισμικό Android στις ανάγκες και τις απαιτήσεις που ορίζουν οι ίδιοι για τις συσκευές τους. Με τον ίδιο τρόπο υπάρχουν και εξωτερικά πρότζεκτ (custom firmware[13] projects) γνωστά στο ευρύ κοινό με τον όρο ROMs. Ένα από τα πιο γνωστά εξωτερικά πρότζεκτ βασισμένο στο Android είναι η Cyanogenmod. Μόνο το Δεκέμβριο του 2013 η Cyanogenmod είχε 9.5 εκατομμύρια καινούργιες εγκαταστάσεις και ενεργοποιήσεις του λογισμικού της (Drake, J. : 2006). Τα πρότζεκτ αυτά και κυρίως αυτό του Cyanogenmod είναι ανεπτυγμένα και βασισμένα στην τελευταία πάντα έκδοση του Android όπως το παρουσιάζει η Google με μερικές τροποποιήσεις από μια ελεύθερη και ξέχωρη κοινότητα από αυτήν του καθαρού Android. Τα λογισμικά αυτά συνήθως είναι εκδόσεις βασισμένες στο κώδικα μελών της εκάστοτε κοινότητας και προσφέρουν τσιμπήματα επιδόσεων (performance tweaks), διαμορφωμένο UI βασισμένο στις απαιτήσεις του κοινού της κοινότητας, καθώς και διάφορα άλλα χαρακτηριστικά.

[13] Firmware: Το **υλικολογισμικό** (αγγλ. **firmware**) είναι το λογισμικό των ηλεκτρονικών συσκευών. Το firmware είναι ένα είδος λογισμικού το οποίο είναι γραμμένο σε γλώσσα μηχανής (ή σε συμβολική γλώσσα) και είναι φτιαγμένο αποκλειστικά και μόνο για ένα μοντέλο συσκευής.

3.Πλατφόρμα iOS

Για να γίνει κατανοητή η πλατφόρμα iOS πρέπει πρώτα να γίνει μια αναφορά σε ένα από τα μεγαλύτερα εμπορικά επιτεύγματα την εποχής μας, το iPhone. Το iPhone είναι η κινητή συσκευή που είναι σχεδιασμένη και κατασκευασμένη από την εταιρία Apple inc. στο Πάλο Άλτο της Καλιφόρνιας. Η λέξη iPhone στις μέρες μας είναι συνυφασμένη με την μοναδικότητα, την κομψότητα, την απλότητα και τη λειτουργικότητα. Οι συσκευές αυτές έχουν δημιουργήσει μια παλίρροια συζητήσεων για το αν το iOS είναι καλύτερο από τα Android και για το αν το iPhone είναι καλύτερο από όλα τα υπόλοιπα κινητά της αγοράς, λόγω του ξεχωριστού και μοναδικού ως προς την συσκευή λειτουργικού τους. Όταν γίνεται αναφορά στο iOS παράλληλα γίνεται αναφορά και στο iPhone, μιας και το iPhone είναι η μοναδική κινητή συσκευή που μπορεί να χρησιμοποιήσει το συγκεκριμένο λογισμικό και ο λόγος είναι απλός. Η Apple, αν και εταιρία κυρίως λογισμικού, είχε ασχοληθεί στο παρελθόν με το σχεδιασμό και την δημιουργία Hardware, βλέπε Mac, Lisa, Apple I και Apple II. Ως εκ τούτου, στην Apple πίστευαν και πιστεύουν ότι το λογισμικό τους δε θα μπορούσε να δουλεύει καλύτερα και με το τρόπο που η ίδια η εταιρία θα ήθελε, παρά μόνο στην ίδια συσκευή που αυτοί θα κατασκεύαζαν σύμφωνα με τα πρότυπα που οι ίδιοι θα καθόριζαν (D5,All things Digital : 2007).

3.1.Κατανοώντας τις ρίζες της πλατφόρμας iOS

Οι «φίλοι» της τεχνολογίας κάθε φορά που αγοράζουν μια καινούργια συσκευή αναρωτιούνται πόσο έχει προχωρήσει η τεχνολογία στην αναβάθμιση των δυνατοτήτων της συσκευής, από την φωτογραφική κάμερα, την οθόνη, έως τις δυνατότητες που προσφέρει το λειτουργικό και οι εφαρμογές που προσφέρονται στα ηλεκτρονικά καταστήματα των εταιριών παραγωγής του λογισμικού. Για όσους όμως ενδιαφέρονται για την ασφάλεια της συσκευής και των εφαρμογών της, η βασική ερώτηση που προκύπτει πάντα κατά την αγορά μιας τέτοιας συσκευή είναι το κατά πόσο ασφαλής είναι η λειτουργία του λογισμικού της συσκευής. Για να ερευνηθεί και στην πορεία να αναλυθούν οι μέθοδοι και οι τεχνικές ασφάλειας σε ένα από τα διασημότερα, σήμερα, λειτουργικά συστήματα της αγοράς, πρέπει πρώτα να γίνει ιστορική αναδρομή στην εταιρία που το κατασκεύασε καθώς και στο ίδιο το λογισμικό, αναζητώντας την φιλοσοφία πίσω από το τελικό προϊόν. Το λειτουργικό σύστημα iOS είναι ένα λειτουργικό σύστημα για κινητές συσκευές που έχει αναπτυχθεί από την εταιρία Apple Inc. και

μπορεί να χρησιμοποιηθεί μόνο σε κινητές συσκευές της ομώνυμης εταιρίας. Το λειτουργικό σύστημα IOS, αξίζει να σημειωθεί, ότι σήμερα χρησιμοποιείται ως το κύριο λειτουργικό σύστημα σε συσκευές που ξεπερνούν τα 900 εκατομμύρια παγκοσμίως (Hoog,A : 2011). Το λειτουργικό σύστημα παρουσιάστηκε το 2007 για το iPhone, στην πρώτη απόπειρα της Apple να εισέλθει δυναμικά στην αγορά των κινητών συσκευών. Στη πορεία όμως αναπτύχθηκε δυναμικά στο χώρο των κινητών συσκευών και τροποποιήθηκε και για άλλες συσκευές της Apple όπως το iPod touch που παρουσιάστηκε τον Σεπτέμβριο του 2007, το iPad που παρουσιάστηκε τον Ιανουάριο του 2010, καθώς και για το iPad mini που παρουσιάστηκε το Νοέμβριο του 2012 και την δεύτερη γενιά Apple το Σεπτέμβριο του 2010. Τον Ιούνιο του 2014, η Apple παρουσίασε άλλη μια συσκευή που χρίζει προσοχής στην τεχνολογική κοινότητα και χρησιμοποιεί το λειτουργικό σύστημα iOS, το Apple iWatch. Η διεπαφή του χρήστη του IOS βασίζεται στην έννοια του άμεσου χειρισμού, με την χρήση πολλαπλών (multi-touch gestures) κινήσεων. Ο χειρισμός γίνεται από μια οθόνη αφής με την χρήση τόσο φυσικών κουμπιών στην συσκευή, όσο και ψηφιακών κουμπιών και ρυθμιστικών διακοπτών, όπως για παράδειγμα το «τσίμπημα» (εντός εισαγωγικών) της οθόνης, την περιστροφή των δαχτύλων κ.ο.κ.. Το λειτουργικό σύστημα IOS χρησιμοποιεί κατά κάποιο τρόπο το ίδιο γενικό πλαίσιο (Framework[14]) με το λειτουργικό σύστημα για υπολογιστές OS X της ίδιας εταιρίας. Η Apple, ως κατασκευάστρια λειτουργικών συστημάτων, δεν δίνει άδεια χρήσης του λειτουργικού της συστήματος σε τρίτους και επιτρέπει την χρήση των λειτουργικών της συστημάτων μόνο για τις συσκευές που η ίδια σχεδιάζει, κατασκευάζει και πουλάει στην αγορά. Τα εργαλεία δημιουργίας του UI του λειτουργικού συστήματος IOS είναι το Cocoa touch παρά το OS X cocoa ώστε να παρέχει τα απαραίτητα εργαλεία για την κατασκευή και ανάπτυξη της διεπιφάνειας του χρήστη παρά των εφαρμογών που ο χρήστης θα χρησιμοποιήσει. Η βασική διαφορά ανάμεσα στα δύο γενικά πλαίσια λογισμικού είναι ότι οι κλάσεις του UI και των APIs διαφέρουν ανάμεσα στα δύο. Για παράδειγμα αντί να υπάρχει η κλάση NSTextField, στο πλαίσιο του Cocoa touch έχουμε την UITextField. Πολλές από τις κλάσεις που έχουν τα δύο πλαίσια έχουν κοινές δυνατότητες παρόλα αυτά δεν έχουν κοινά ονόματα αλλά μπορούν να επιτύχουν κοινά αποτελέσματα με λίγες μόνο αλλαγές. Ακόμη, πολλές από τις κλάσεις που είναι

[14] Γενικό πλαίσιο (Framework):Στον προγραμματισμό των ηλεκτρονικών υπολογιστών, το γενικό πλαίσιο του λογισμικού ορίζεται ως η περίληψη της γενικής φόρμας ανάπτυξης του λογισμικού γύρω από την οποία μπορεί ο χρήστης έχοντας κάποιες γενικές λειτουργίες να αλλάξει επιλεκτικά γράφοντας πρόσθετο κώδικα, παρέχοντας έτσι συγκεκριμένες εφαρμογές λογισμικού.

διαθέσιμες στο Cocoa δεν υπάρχουν στο Cocoa Touch. Παρόλα αυτά και τα δύο πλαίσια καθώς και τα APIs γράφονται σε Objective-C. Για το λόγο αυτό, παρόλο που μοιράζονται τα δύο λογισμικά κοινά εργαλεία για την κατασκευή τους, δεν μπορούν να χρησιμοποιήσουν κοινές εφαρμογές. Εν κατακλείδι, η πλατφόρμα IOS σήμερα θεωρείται μια όμορφη σχεδιαστικά. απλή προς την χρήση και γρήγορη, ως προς την ανταπόκριση της, λειτουργική πλατφόρμα.

3.2.Εξερευνώντας τις συσκευές iOS

Για να μπορέσουμε να κατανοήσουμε επαρκώς την φιλοσοφία του λειτουργικού συστήματος και να εντρυφήσουμε στο κομμάτι της ασφάλειας που αφορά τόσο τις μεθόδους που ακολούθησε η εταιρία για την προστασία των δεδομένων των χρηστών, όσο και για τις ενέργειες που οφείλει ο χρήστης να πραγματοποιεί, πρέπει να εξερευνηθούν οι συσκευές που μπορούν να αξιοποιήσουν το λογισμικό αυτό. Για ένα διάστημα τα κινητά τηλέφωνα χρησιμοποιούνταν από το κοινό μόνο για επικοινωνία με την μορφή τηλεφώνου. Η τεχνολογία όμως εξελίχθηκε απροσδόκητα δημιουργικά και στην σκηνή της κινητής τηλεφωνίας παρουσιάστηκαν καινούργιες ανάγκες. Το κοινό πλέον χρησιμοποιούσε το κινητό του όχι μόνο ως τηλέφωνο αλλά και για μηνύματα, για να βλέπουν το ημερολόγιο τους, να δημιουργούν εκδηλώσεις, υπενθυμίσεις, να αποθανατίζουν στιγμές διαμέσου φωτογραφιών και βίντεο, να μοιράζονται πληροφορίες με τρίτους διαμέσου του Διαδικτύου, να παρακολουθούν τα ηλεκτρονικά τους μηνύματα (Emails) και να περιηγούνται στο Διαδίκτυο. Είναι γνωστό ότι σήμερα, περίπου 5 δισεκατομμύρια άνθρωποι χρησιμοποιούν αυτό που ονομάζουμε έξυπνη κινητή συσκευή και ο αριθμός αυτός συνεχώς αυξάνεται (Hoog,Strzempka:2011). Στη μάχη των συσκευών αυτών, η εταιρία Apple εισήλθε τον Ιούνιο του 2007 με την παρουσίαση του τότε πρωτοποριακού iPhone 1 που χρησιμοποιούσε το λειτουργικό σύστημα iPhone OS 1.0. Το τότε λειτουργικό στο iPhone 1 επέτρεπε στον χρήστη πέρα από τα συνηθισμένα, να καλεί και να δέχεται τηλεφωνήματα, να αναπαράγει μουσική, να βλέπει βίντεο, να μπορεί να διαδρά με την συσκευή χωρίς φυσικά κουμπιά αλλά διαμέσου της αφής απευθείας πάνω στην οθόνη και να «σερφάρει» στο Διαδίκτυο μέσω wifi hotspot. Από την πρώτη κιόλας παρουσίαση του πρώτου iPhone η εταιρία παρουσίασε έως σήμερα μια πληθώρα συσκευών γνωστά και ως iDevices, με πρωτοπόρο το iPhone. Μέχρι σήμερα η Apple έχει παρουσιάσει 8 εκδόσεις λογισμικών σε συνολικά δέκα γενιές iPhone , πέντε γενιές-μοντέλα iPod touch , εννιά γενιές-μοντέλα iPad , τρείς γενιές iTv και πρόσφατα, από τον

Σεπτέμβριο του 2014, την πρώτη γενιά ψηφιακών έξυπνων ρολογιών iWatch. Παρακάτω παραθέτονται πίνακες με τις συσκευές που χρησιμοποιούν το λογισμικό IOS.

iPhone Devices

Model	iPhone 1st Gen	iPhone 3G	iPhone 3GS	iPhone 4	iPhone 4s	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6+
Pictures										
Initial release OS	iOS 1.0	iOS 2.0	iOS 3.0	iOS 4.0(GSM) iOS 4.2.2 (CDMA)	iOS 5.0	iOS 6.0	iOS 7.0	iOS 7.0	iOS 8.0	iOS 8.0
Highest Supported OS	iOS 3.1.3	iOS 4.2.1	iOS 6.1.6	iOS 7.1.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2

Πίνακας 3: Όλα τα μοντέλα iPhone με τις αρχικές και πιο πρόσφατες εκδόσεις του λογισμικου iOS που χρησιμοποιούν

iPod Devices

Model	iPod 1st Gen	iPod 2nd Gen	iPod 3rd Gen	iPod 4th Gen	iPod 5th Gen
Pictures					
Initial release OS	iOS 1.0	iOS 2.0	iOS 3.1.1	iOS 4.1	iOS 6.0
Highest Supported OS	iOS 3.1.3	iOS 4.2.1	iOS 5.1.1	iOS 6.1.6	iOS 8.2

Πίνακας 4: Όλα τα μοντέλα iPod με τις αρχικές και πιο πρόσφατες εκδόσεις του λογισμικου iOS που χρησιμοποιούν

iPad Devices

Model	iPad1st Gen	iPad 2	iPad 3rd Gen	iPad 4th Gen	iPad Air	iPad Air 2	iPad Mini	iPad Mini 2	iPad Mini 3
Pictures									
Initial release OS	iOS 3.2	iOS 4.2.1	iOS 5.1	iOS 6.0(WIFI) iOS 6.0.1 (Cell)	iOS 7.0(WIFI) iOS 7.0.3(CELL)	iOS 8.1	iOS 6.0	iOS 7.0.3	iOS 8.1
Highest Supported OS	iOS 5.1.1	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2	iOS 8.2

Πίνακας 5: Όλα τα μοντέλα iPad με τις αρχικές και πιο πρόσφατες εκδόσεις του λογισμικού iOS που χρησιμοποιούν

3.3. Εκδόσεις της πλατφόρμας

Από την στιγμή παρουσίασης της πρώτης κινητής συσκευής από την εταιρία Apple με το λειτουργικό σύστημα IOS η εταιρία έχει συνεχίσει να παρουσιάζει αναβαθμίσεις στο λειτουργικό της σύστημα έως και σήμερα, με τελευταία αναβάθμιση στις 20 Οκτωβρίου του 2014 με την έκδοση 8.1., προσφέροντας ένα πιο ασφαλές κι εύκολο στην χρήση λειτουργικό σύστημα. Η ιστορία των εκδόσεων του λειτουργικού συστήματος ξεκινάει τον Ιούνιο του 2007 και συγκεκριμένα στις 29 Ιουνίου, με την παρουσίαση του πρώτου λειτουργικού της συστήματος στην τότε ολοκαίνουργια και καινοτόμα κινητή της συσκευή iPhone 1. Η τελική αναβάθμιση της συσκευής αυτής ήταν η έκδοση 1.1.5 η οποία δόθηκε στο ευρύ κοινό λίγες μέρες πριν την παρουσίαση της έκδοση 2.0. Όπως αναφέραμε προηγουμένως στις 11 Ιουνίου του 2008 η Apple δίνει στη διάθεση του κοινού την έκδοση iPhone OS 2.0. με αναβαθμίσεις έως την υποέκδοση 2.2.1. Βέβαια δεν ήταν μέχρι τον Ιούνιο του 2010 που πρώτο ονομάστηκε το λογισμικό IOS με τον αριθμητικό κωδικό 4.0., έως τότε το λειτουργικό ονομαζόταν iPhone OS. Τον Ιούνιο του 2009 η Apple δίνει στο ευρύ κοινό την ολοκαίνουργια αναβαθμισμένη έκδοση iPhone OS 3.0 με αναβαθμίσεις για το υπάρχον iPhone 2 μέχρι την υποέκδοση 3.1.3 με ημερομηνία παρουσίασης στις 2 Φεβρουαρίου του 2010. Το τότε ολοκαίνουργιο iPod Touch μια συσκευή παρόμοια με το iPhone, με την διαφορά ότι δεν είχε την δυνατότητα να καλεί και να δέχεται κλήσεις λόγω απουσίας Sim, ερχόταν απευθείας με την έκδοση 3.1.3. Την

ίδια χρονιά παρουσιάστηκε το τότε ολοκαίνουργιο Tablet της εταιρίας Apple, με το όνομα iPad, αξιοποιώντας το ειδικά διαμορφωμένο λογισμικό iOS που όμως αναβαθμίστηκε μόνο για το iPad σε 3.2. και αργότερα αναβαθμίστηκε στην έκδοση 3.2.2. Τον Ιούνιο του 2010 έρχεται η μεγάλη αλλαγή του λογισμικού, με το λογισμικό πλέον να μην ονομάζεται iPhone OS αλλά iOS και τον αριθμητικό κωδικό 4.0. Το λειτουργικό αυτό σύστημα αρχικά ήταν διαθέσιμο μόνο για το iPhone και το iPod Touch. Το ολοκαίνουργιο τότε λογισμικό iOS 4.0. και η ομάδα της Apple που το παρουσίασε, ανακοίνωσε ότι είχε πάνω από 1500 καινούργια API's για του προγραμματιστές που ενδιαφέρονταν να δημιουργήσουν εφαρμογές για το οικοσύστημα της Apple, ανάμεσα στα API αυτά ήταν και το πολύ αναμενόμενο χαρακτηριστικό των πολλαπλών ενεργειών (Multitasking feature). Η δεύτερη γενιά iPod Touch και το καινούργιο iPhone 3G ανακοινώθηκαν με την έκδοση 4.2.1. Αυτό το λειτουργικό σύστημα, από την έκδοση 4.2 και μετά, φέρνει στου χρήστες την δυνατότητα πολλαπλών ανοιχτών εφαρμογών κάνοντας τις εναλλαγές από εφαρμογή σε εφαρμογή ένα μόνο κλικ στην οθόνη. Η τελική υποέκδοση του iOS 4 είναι η υποέκδοση 4.3. που παρουσιάστηκε για όλες τις κινητές συσκευές της Apple. Στις 6 Ιουνίου του 2011, η Apple δίνει μια μικρή γεύση στο κοινό της με την παρουσίαση της έκδοσης IOS 5.0, παρουσιάζοντας ταυτόχρονα το διαμορφωμένο της IOS 4.4.beta, για την φορητή της συσκευή για τηλεοράσεις Apple TV, καθώς και το τότε ολοκαίνουργιο και ακόμη σε δοκιμαστικό στάδιο iCloud. Η τελική παρουσίαση του λογισμικού συμπεριλάμβανε και την υπηρεσία iMessage, μια υπηρεσία δωρεάν αποστολής ηλεκτρονικών μηνυμάτων για συσκευές που «τρέχουν» το λογισμικό 5.0, καθώς και ένα ολοκαίνουργιο σύστημα ειδοποιήσεων και άλλα πολλά λιγότερο σημαντικά χαρακτηριστικά. Η έκδοση 5.0 είχε μόνο τρεις αναβαθμίσεις: την υποέκδοση 5.0.1 , την 5.1 και την 5.1.1. η οποία γίνονταν όλες over the air[15] όπως ακριβώς και στα Android χωρίς την αναγκαία σύνδεση της συσκευής στον προσωπικό υπολογιστή του χρήστη. Στις 11 Ιουνίου στο διεθνές σεμινάριο προγραμματιστών της Apple (WWDC), η εταιρία παρουσιάζει την έκδοση 6.0. Στο σημείο αυτό αξίζει να σημειωθεί ότι οι προηγούμενες συσκευές iPhone 1, 2, 3G καθώς και το iPod 1ης, 2ης, και 3ης γενιάς καθώς και το iPad 1ης γενιάς, δεν μπορούσαν πλέον να δεχτούν την αναβαθμισμένη έκδοση καθιστώντας πλέον τις συσκευές άχρηστες για το κοινό. Η εταιρία στο ίδιο μοτίβο παρουσιάζει την νέα

[15] Over the air: ως OTA ορίζονται οι μέθοδοι διανομής αναβαθμίσεων λογισμικού, ρυθμίσεων καθώς και κλειδιών κρυπτογράφησης (σε ορισμένες περιπτώσεις), σε συσκευές όπως κινητά τηλέφωνα κ.ο.κ. Στις περιπτώσεις των κινητών συσκευών , η διανομή OTA του λογισμικού γίνεται χωρίς τη χρήση προσωπικού υπολογιστή παρά μόνο με τη σύνδεση της ίδιας της συσκευής που θα αναβαθμιστεί, στο Διαδίκτυο.

έκδοση για τις συσκευές iPhone 3gs, iPad 2, iPod Touch 4rth Generation και μετέπειτα. Αργότερα στις 12 Σεπτεμβρίου του 2012, η Apple ανακοινώνει το καινούργιο iPhone 5 μαζί με το ανασχεδιασμένο iPod Touch 5ης γενιάς. Τελικά, δίνει την αναβάθμιση στο ευρύ κοινό στις 19 Σεπτεμβρίου του ίδιου χρόνο, είτε over-the-air, είτε μέσω της εφαρμογής iTunes. Στις 10 Ιουνίου του 2013 στο προγραμματισμένο πλέον ετήσιο συνέδριο προγραμματιστών της Apple ανακοινώθηκε η έκδοση 7.0., ενώ η διάθεση του λογισμικού στο ευρύ κοινό έγινε στις 10 Σεπτεμβρίου του 2013 παρουσιάζοντας παράλληλα την καινούργια συσκευή της iPhone 5c και 5s. Με την ανακοίνωση αυτή η Apple για μια ακόμη φορά σταμάτησε την υποστήριξη για τις παλαιότερες συσκευές iPhone 3GS και iPod touch 4ης γενιάς. Η τελική αναβάθμιση της έκδοσης αυτής για το Iphone 4 ήταν η έκδοση 7.1.2 αφήνοντας έτσι άλλη μια συσκευή της ομώνυμης εταιρίας στο νεκροταφείο των κινητών. Τελευταία και πιο πρόσφατη έκδοση του λογισμικού είναι η έκδοση 8.1 που ανακοινώθηκε στις 2 Ιουνίου του 2014 στο ετήσιο συνέδριο προγραμματιστών της εταιρίας και παρουσιάστηκε στις 9 Σεπτεμβρίου του 2014 στην εκδήλωση του καινούργιου iPhone 6 και 6+ ενώ δόθηκε στην διάθεση του κοινού η ολοκληρωμένη έκδοση στις 17 Σεπτεμβρίου. Στο ίδιο μοτίβο της εταιρίας, σταμάτησε η υποστήριξη και η αναβάθμιση στην τελευταία αυτή έκδοση για το iPhone 4. Η μόνη συσκευή που δέχθηκε πέντε συνεχόμενες αναβαθμίσεις και δεν έχει διακοπεί ακόμη η υποστήριξη και η αναβάθμιση του λογισμικού της, είναι το iPad 2. Στον παρακάτω πίνακα διακρίνονται ξεκάθαρα όλες οι αναβαθμίσεις του λογισμικού.

Legend: Discontinued Current Beta

Version	Build	Release date	Highest version for		
3.1.3	7E18	02 Feb,2010	Iphone (1st Gen)	Ipod touch (1st Gen)	
4.2.1	8C148	22 Nov,2010	Iphone 3G	Ipod touch (2nd Gen)	
5.1.1	9B206	07 May,2012		Ipod touch(3rd Gen)	Ipad (1st Gen)
6.1.6	10B500	21 Feb,2014	Iphone 3GS	Ipod touch(4th Gen)	
7.1.2	11D257/58	30 Jun,2014	Iphone 4		
8.2	12D508	09 Mar,2015	Iphone 4s,5,5c Iphone 5s,6,6+	Ipod touch(5th Gen)	Ipad 2, 3rd-4th (Gen) Ipad Mini,Air,Air 2
8.3	12F61	24 Mar,2015	Iphone 4s,5,5c,5s Iphone 6,6+	Ipod touch(5th Gen)	Ipad 2,3rd-4th (Gen) Ipad Mini,Air,Air 2

Πίνακας 6: με όλες τις εκδόσεις iOS και τα μοντέλα που υποστηρίζουν τις εκάστοτε εκδόσεις

3.4.Η εφαρμογή iTunes

Το iTunes είναι μια εφαρμογή για τον υπολογιστή η οποία λειτουργεί ταυτόχρονα, ως μέσω αναπαραγωγής τραγουδιών και ταινιών, ως βιβλιοθήκη πολυμέσων, ως εφαρμογή διαχείρισης κινητών λειτουργικών συστημάτων κι ως μια ψηφιακή αγορά για φωτογραφίες, βίντεο, μουσική όλων των ειδών, τηλεοπτικές εκπομπές, ταινίες και φυσικά εφαρμογές για τις συσκευές που χρησιμοποιούν το λειτουργικό σύστημα iOS. Η πιο πρόσφατη αναβαθμισμένη έκδοση του iTunes είναι το iTunes 12, που παρουσιάστηκε στις 16 Οκτωβρίου του 2014 και είναι διαθέσιμη για το λειτουργικό σύστημα που χρησιμοποιούν οι υπολογιστές της Apple OS X v10.7.5 κι αργότερα, καθώς και για τους υπολογιστές που χρησιμοποιούν το λειτουργικό σύστημα Windows και συγκεκριμένα από την έκδοση XP και μεταγενέστερα.

3.5.Αναβάθμιση συσκευών

Η Apple, όπως όλες οι υπόλοιπες εταιρίες λογισμικών, σε τακτά χρονικά διαστήματα αναβαθμίζει το λογισμικό των συσκευών της. Στις αναβαθμισμένες αυτές εκδόσεις συνήθως παρουσιάζονται καινούργια χαρακτηριστικά (features) στο λογισμικό, διορθώνουν τυχόν προβλήματα των παλαιοτέρων εκδόσεων, καλύπτουν κενά ασφαλείας και συνάμα αυξάνουν τα μέτρα προστασίας κατά των κακόβουλων επιθέσεων από τρίτους. Ο χρήστης έχει την επιλογή, όταν υπάρχει διαθέσιμη καινούργια έκδοση του λογισμικού, είτε να παραμείνει στην ίδια έκδοση, είτε να αναβαθμίσει τη συσκευή του στην τελευταία έκδοση του διαθέσιμου λογισμικού που του προσφέρεται. Η αναβάθμιση γίνεται μέσω της εφαρμογής iTunes στον υπολογιστή του χρήστη και συνήθως είναι μια εύκολη διαδικασία χωρίς μεγάλες διαδικασίες και βήματα. Αναλυτικά: Μόλις η συσκευή συνδεθεί και αναγνωριστεί από τον υπολογιστή, ο χρήστης πατάει το κουμπί αναβάθμιση και η συσκευή αναβαθμίζεται κατευθείαν στην τελευταία έκδοση που είναι διαθέσιμη. Τα iPhone συγκεκριμένα είναι χωρισμένα σε δυο μέρη ως προς τον αποθηκευτικό τους χώρο (Partitions). Ο ένας περιέχει τα αρχεία του χρήστη και ο άλλος περιέχει το λογισμικό. Λόγω των δύο αυτών partitions η συσκευή αναβαθμίζεται χωρίς να επηρεάζονται τα δεδομένα του χρήστη. Όλες οι εκδόσεις και υποεκδόσεις του λειτουργικού συστήματος iOS αναφέρονται αναλυτικά στην υποενότητα 2.3.

3.6.Back-Porting

Με τον όρο Back-Porting ορίζουμε τη διόρθωση προβλημάτων που σχετίζονται κυρίως με προβλήματα (bugs) του λογισμικού αλλά και κενά ασφαλείας της τωρινής έκδοσης του λογισμικού σε παλαιότερες εκδόσεις. Παραδείγματος χάρη, η τελευταία γνωστή έκδοση του iOS μέχρι τώρα που γράφεται η έρευνα αυτή, είναι η έκδοση 8.1. Όπως και στην Google έτσι και η Apple, αν αντιληφθεί ότι υπάρχει κάποιο κενό ασφαλείας σε παλαιότερες εκδόσεις, τότε διορθώνει τα κενά αυτά είτε με απευθείας αναβάθμιση είτε με κάποιο Patch αφήνοντας όμως τις παλαιότερες συσκευές, που δεν υποστηρίζουν το λειτουργικό σύστημα 8.1 και δεν έχουν αναβαθμιστεί σε αυτήν, απροστάτευτες. Η Apple όμως σε σχέση με την Google κρατά σε ποσοστό 90% τα Tablet της ενημερωμένα στην τελευταία πάντα έκδοση, ενώ στα κινητά της κρατά πολύ καλή στάση απέναντι στους καταναλωτές της αναβαθμίζοντας τα iPhones στην τελευταία έκδοση iOS 8.1 ακόμη και κινητά κατασκευασμένα μέχρι και πέντε χρόνια πίσω. Ο λόγος που η Apple κρατά βέβαια τις συσκευές που παράγει δεν γίνεται με σκοπό πρωτίστως την ασφάλεια των συσκευών της και αυτό μπορεί να παρατηρηθεί σε όλες τις συσκευές

της, αλλά το business plan της εταιρίας να προσφέρει την καλύτερη εμπειρία χρήσης χωρίς bottlenecks[16] στο hardware των συσκευών της και κυρίως χωρίς bugs που καθυστερούν το λογισμικό της.

3.7.Apple app store

Το ψηφιακό κατάστημα της Apple για της συσκευές της ομώνυμης εταιρίας έχει κατασκευαστεί και διατηρείται από την ίδια την εταιρία από το 2009 έως σήμερα με σύνολο περισσότερες από 1,3 εκατομμύρια εφαρμογές. Η υπηρεσία επιτρέπει στους χρήστες των συσκευών iDevices να κατεβάζουν εφαρμογές φτιαγμένες ειδικά για το λογισμικό IOS. Το ψηφιακό κατάστημα υπάρχει διαθέσιμο και στην εφαρμογή iTunes στον ηλεκτρονικό υπολογιστή. Υπάρχουν δύο τρόποι για να αγοράσει και να κατεβάσει κάποιος τις εφαρμογές που θέλει στην κινητή του συσκευή. Είτε διαμέσου του iTunes, είτε απευθείας στο κινητό του. Οι εφαρμογές που δημιουργούνται για τις συσκευές iDevices είναι εφαρμογές που εκμεταλλεύονται τα μοναδικά χαρακτηριστικά των συσκευών αυτών, όπως ανιχνευτές κίνησης κ.ο.κ.. Η διαδικασία με την οποία μπορεί κάποιος να δημιουργήσει μια εφαρμογή για τις συσκευές αυτές και να τις ανεβάσει στο ηλεκτρονικό κατάστημα της Apple είναι απλός. Αρχικά κάνει συνδρομή στο πρόγραμμα της Apple δημιουργώντας ένα μοναδικό όνομα χρήστη και ένα μοναδικό κωδικό. Η επόμενη φάση αφορά την συμπλήρωση των αληθινών του στοιχείων και την επικύρωση αυθεντικότητας τους από την Apple. Τέλος πληρώνει ένα χρηματικό ποσό των 100 δολαρίων Αμερικής. Εφόσον συμφωνήσει στους όρους χρήσης της πολιτικής ασφαλείας και πώλησης ψηφιακών εφαρμογών της Apple, του παρέχεται το NDK της Apple. Στη συνέχεια, εάν δημιουργήσει επιτυχώς την εφαρμογή του και την ανεβάσει στο ηλεκτρονικό κατάστημα, η Apple ελέγχει την εφαρμογή για κακόβουλο κώδικα κι όταν περάσει όλα τα τεστ της Apple, εκ των οποίων ένα είναι το εάν η εφαρμογή προσβάλει φυσικά πρόσωπα κ.ο.κ, τότε παρέχει στην εφαρμογή ένα μοναδικό κωδικό και την πιστοποίηση της Apple. Μετά από όλες αυτές τις φάσεις η εφαρμογή πλέον θα βρίσκεται διαθέσιμη στο ηλεκτρονικό κατάστημα για του χρήστες. Εάν σε οποιαδήποτε χρονική στιγμή η εφαρμογή παρουσιάσει κακόβουλο λογισμικό, η Apple αμέσως κατεβάζει την εφαρμογή από το ηλεκτρονικό της κατάστημα και αμέσως μηνύει τον προγραμματιστή ή την εταιρία ανάπτυξης της εφαρμογής.

[16] Bottleneck:Η συμφόρηση, αναφέρεται επί λεξεί στο στενό τμήμα του μπουκαλιού που βρίσκεται συνήθως στην κορυφή. Στη μηχανικη, ο όρος χρησιμοποιείται για να περιγραφεί ένα φαινόμενο όπου η απόδοση ή η ικανότητα ενός ολόκληρου συστήματος περιορίζεται από ένα ενιαίο ή μικρό αριθμό εξαρτημάτων ή πόρων.

4.Εισαγωγή στο μοντέλο ασφαλείας των σύγχρονων κινητών συσκευών

Η επόμενη γενιά των λειτουργικών συστημάτων δε πρόκειται να είναι σε επιτραπέζιους ή φορητούς υπολογιστές, αλλά σε μικρές κινητές συσκευές που ο χρήστης θα «κουβαλάει» καθημερινά μαζί του. Εκτιμάται ότι στα επόμενα χρόνια κάθε χρήστης θα έχει πάνω από πέντε συσκευές οι οποίες θα επικοινωνούν μεταξύ τους, εκ των οποίων μερικές θα έχει μαζί του ο χρήστης καθημερινά. Όσο τα χρόνια περνάνε και η τεχνολογία αναπτύσσεται, τόσο παρατηρείται η αναγκαιότητα ο πηγαίος κώδικας του λογισμικού να είναι ανοιχτός και προσβάσιμος για όλους. Η κοινότητα του Open Source έχει ανοίξει νέους ορίζοντες και αγορές για νέα προγράμματα, ενώ η σύνδεση των συσκευών μεταξύ τους αλλά και με το Διαδίκτυο, επιτρέπει μεγαλύτερη ενοποίηση με τις υπάρχουσες ηλεκτρονικές υπηρεσίες, αλλά και με τις υπηρεσίες που θα δημιουργηθούν στο μέλλον. Ωστόσο, όσο αυξάνονται οι πληροφορίες που σήμερα ο χρήστης αξιοποιεί και συλλέγει στις προσωπικές του κινητές συσκευές καθώς και οι υπηρεσίες που υποστηρίζουν και παρέχουν οι συσκευές του, τόσο αυξάνονται και οι πιθανότητες των ευπαθειών τις οποίες μπορεί να κινδυνεύει ένα σύστημα (Halbronn, C. & Sigwald, J. : 2010). Σήμερα αποτελεί πρώτη προτεραιότητα ο σχεδιασμός και η ασφάλεια των υποδομών ενός λειτουργικού συστήματος που μπορεί να χρησιμοποιηθεί με ασφάλεια. Τα κύρια χαρακτηριστικά-απαιτήσεις της ασφάλειας οποιουδήποτε λογισμικού παραμένουν τα ίδια με παλαιότερα, παρόλο που οι τεχνικές μπορεί να προσαρμόζονται στις ανάγκες τόσο της εποχής, με βάση τις ηλεκτρονικές συσκευές των αγορών, όσο και του λογισμικού και της φιλοσοφίας των εταιριών που τα σχεδιάζουν και τα παρέχουν στην αγορά. Αναφορικά, τα κύρια χαρακτηριστικά ασφάλειας οποιουδήποτε λογισμικού που πρέπει να αποζητά ο χρήστης είναι η εμπιστευτικότητα (Confidentiality), η ακεραιότητα (Integrity) και η διαθεσιμότητα (Availability). Πιο συγκεκριμένα, με τον όρο εμπιστευτικότητα (Confidentiality) ορίζουμε την ιδιότητα ότι οι πληροφορίες δεν γίνονται διαθέσιμες και δεν αποκαλύπτονται σε μη εξουσιοδοτημένους χρήστες, οντότητες και διαδικασίες. Με τον όρο ακεραιότητα (Integrity) ορίζουμε την ιδιότητα της προστασίας, της ορθότητας και της πληρότητας ενός αγαθού και της αποφυγής μη εξουσιοδοτημένης τροποποίησης του. Τέλος, ως διαθεσιμότητα (Availability) ορίζουμε την ιδιότητα ενός αγαθού να είναι διαθέσιμο προς χρήστη όταν ζητείται από μια εξουσιοδοτημένη οντότητα (Gritzalis, D.:2004). Το μοντέλο ασφάλειας των λειτουργικών συστημάτων, συμπεριλαμβανομένων και των κινητών συσκευών όπως του Android και του iOS είναι πολύπλοκο και βασισμένο στο καλά μελετημένο μοντέλο ασφαλείας Unix, το οποίο θα

μελετήσουμε παρακάτω, ώστε να μπορεί να καλύψει όλες τις ανάγκες ασφαλείας της σημερινής εποχής.

4.1. Μοντέλο UNIX

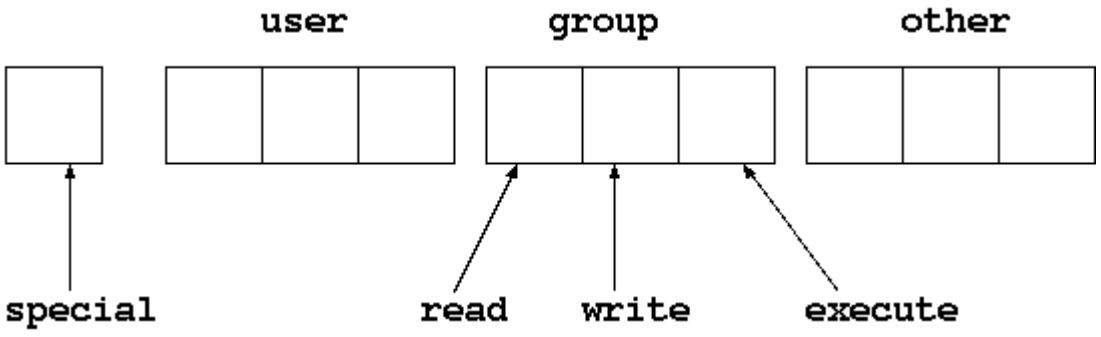
Ως μοντέλο UNIX ορίζουμε το μοντέλο ασφαλείας ενός λειτουργικού συστήματος UNIX ή ενός λειτουργικού συστήματος που υιοθετεί χαρακτηριστικά από το μοντέλο UNIX (Curry, D. A. : 1992). Αναφορικά, το μοντέλο του UNIX προσδίδει διαφορετικές άδειες ανά χρήστη (file permission), μερική ή ολική απομόνωση των διαφορετικών διαδικασιών ανά εφαρμογή και ανά χρήστη (process isolation), μηχανισμούς επικοινωνίας μεταξύ των διαφόρων διαδικασιών (inter-process communication) καθώς και δυνατότητα τροποποιήσεις του kernel[17] ώστε να μπορεί να προσαρμοστεί στις ανάγκες του λειτουργικού συστήματος. Παρακάτω θα αναλυθούν τα κύρια χαρακτηριστικά του μοντέλου UNIX καθώς και επιπρόσθετων χαρακτηριστικών που έχουν προσθέσει οι δύο εταιρίες Google και Apple, στο λειτουργικό τους σύστημα, ενώ θα αναλυθούν περαιτέρω στο κεφάλαιο 4-5 εξονυχιστικά τα μοντέλα των δύο αυτών λειτουργικών συστημάτων.

4.1.1. Δικαιώματα φακέλων (File permission)

Ένας από τους μηχανισμούς άμυνας απέναντι σε ανεπιθύμητους εισβολείς, είναι ο μηχανισμός των δικαιωμάτων χρήσης που προσφέρονται από το σύστημα αρχείων (file system). Αυτά τα δικαιώματα (permissions), επιτρέπουν στους χρήστες, την πρόσβαση σε διαφόρους φακέλους και καταλόγους του συστήματος με δικαίωμα όχι μόνο για ανάγνωση των αρχείων αλλά εγγραφής, αναζήτησης και εκτέλεσης (ισχύει για αρχεία τα οποία μπορούν να εκτελεσθούν από την συσκευή) (Curry, D. A. : 1992). Τα δικαιώματα αρχείων όπως ονομάζονται επιτρέπουν σε συγκεκριμένα προγράμματα να έχουν ξεχωριστά δικαιώματα διαχείρισης, αυτά του διαχειριστή (super-user permissions). Τα δικαιώματα διακρίνονται στο μοντέλο του Unix σε τρεις διακριτές κατηγορίες-πεδία, σε αυτές του απλού χρήστη (user), ανά ομάδες (group) και σε κάποιες περιπτώσεις κατηγορίες συγκεκριμένου σκοπού (others). Κάθε αρχείο ή κατάλογος έχει τριών ειδών δικαιώματα που συνδέονται με αυτό. Ένα σετ δικαιωμάτων για τον χρήστη στον οποίο ανήκει το αρχείο (UID), ένα σετ δικαιωμάτων για τους χρήστες που ανήκουν στην ομάδα (GID) και ένα σετ για όλους τους υπόλοιπους χρήστες (the "world permission"). Κάθε σετ δικαιωμάτων περιλαμβάνει τρία ίδια bits πρόσβασης τα οποία διαχειρίζονται το

[17] Kernel:Στην πληροφορική το Kernel είναι το πρόγραμμα εκείνο που διαχειρίζεται της αιτήσεις inputs και outputs του λογισμικού και τις οποίες στη συνέχεια μεταφράζει σε επεξεργάσιμα δεδομένα, ως οδηγίες για τον κεντρικό επεξεργαστή.

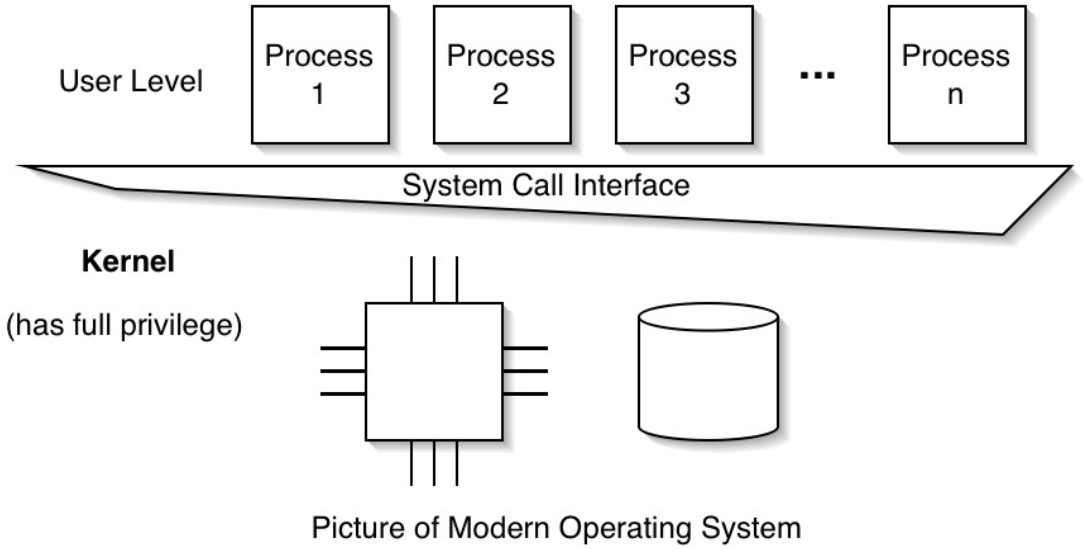
δικαίωμα εγγραφής (write), το δικαίωμα ανάγνωσης (read) και το δικαίωμα εκτέλεσης (execute). Το δικαίωμα ανάγνωσης δουλεύει με τον εξής τρόπο. Αν επιλεχθεί το αρχείο ή ο κατάλογος, τότε μπορεί να διαβαστεί από τον χρήστη. Στην περίπτωση ενός καταλόγου, η πρόσβαση ανάγνωσης επιτρέπει σε έναν χρήστη να δει τα περιεχόμενα του καταλόγου, δηλαδή τα ονόματα των αρχείων που περιλαμβάνονται σ' αυτό, αλλά να μην έχουν πρόσβαση σε αυτά. Το δικαίωμα εγγραφής από την άλλη λειτουργεί με τον εξής τρόπο. Αν επιλεχθεί, το αρχείο ή ο κατάλογος τότε μπορεί να εγγραφούν νέα δεδομένα σε αυτό επιτρέποντας δηλαδή την τροποποίηση τους. Στην περίπτωση ενός καταλόγου, δικαίωμα εγγραφής συνεπάγεται την δυνατότητα να δημιουργηθούν, να διαγραφούν και να μετονομαστούν αρχεία. Σημειώνεται ότι η δυνατότητα να αφαιρεθεί ένα αρχείο δεν ελέγχεται από την παραπάνω ιεραρχία δικαιωμάτων αλλά από την συγκεκριμένη άδεια που έχει το ίδιο το αρχείο, ενώ το δικαίωμα εκτέλεσης έχει την εξής χρήση. Αν επιλεχθεί, το αρχείο ή ο κατάλογος μπορούν να εκτελεστούν (μπορεί επίσης να βρεθεί σε αναζήτηση). Στην περίπτωση ενός αρχείου, το δικαίωμα εκτέλεσης δίνει στον χρήστη την άδεια να τρέξει ένα πρόγραμμα που εμπεριέχεται στον φάκελο αυτό. Εκτελώντας μεταγλωττισμένα προγράμματα (compiled binary programs) προϋποθέτουν μόνο να έχει ο χρήστης το δικαίωμα εκτέλεσης στον φάκελο που βρίσκεται το πρόγραμμα, ενώ για να μπορέσει να εκτελέσει κάποιο shell script προϋποθέτει να έχει το δικαίωμα τόσο της ανάγνωσης όσο και το δικαίωμα εκτέλεσης. Υπάρχει βέβαια και ένα τέταρτο σετ των τριών bit που υποδεικνύουν ειδικά χαρακτηριστικά που σχετίζονται με το αρχείο το οποίο ονομάζεται set-user-id. Αν επιλεχθεί, αυτά τα bit ελέγχουν το set-user-id status του φακέλου. Όταν ένα πρόγραμμα εκτελείται, τότε εκτελείται με τα δικαιώματα του χρήστη στον οποίο ανήκει το πρόγραμμα.



Σχήμα 1: *Τα τρία διακριτά πεδία δικαιωμάτων του μοντέλου UNIX*

4.1.2. Μηχανισμός απομόνωσης διεργασιών (Process Isolation)

Με τον όρο απομόνωση διεργασιών (process isolation) ορίζουμε τα διαφορετικά επίπεδα τεχνολογιών σε επίπεδο λογισμικού και hardware που δουλεύουν μαζί με σκοπό την προστασία των διαφορετικών διεργασιών ενός λειτουργικού συστήματος (Curry, D. A. : 1992). Πιο συγκεκριμένα, εάν υποθέσουμε ότι έχουμε την διεργασία Α και την διεργασία Β. Ο μηχανισμός απομόνωση διεργασιών προστατεύει την διεργασία Β και αντίστροφα από την διεργασία Α όταν μια από της δύο εγγράφουν καινούργια στοιχεία ή εκτελούν κάποιες ενέργειες. Η μέθοδος αυτή μπορεί να υλοποιηθεί με τον χώρο εικονικών διευθύνσεων (virtual address space), όπου η εικονική διεύθυνση της διεργασίας Α θα είναι διαφορετική από την εικονική διεύθυνση Β με αποτέλεσμα η Α να μην επηρεάζει την Β.

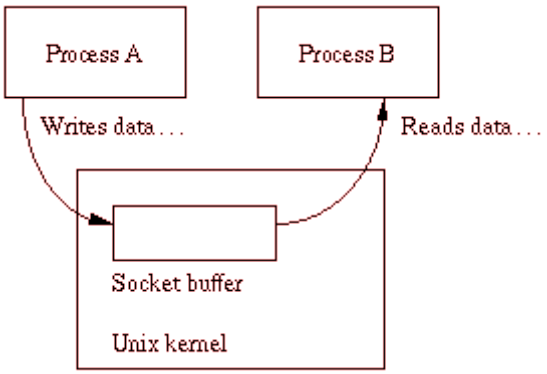


Σχήμα 2: *Λειτουργίας του μηχανισμού απομόνωσης διεργασιών*(**www.read.cs.ucla.edu**)

4.1.3.Inter processes communication

Σε ένα σύστημα που χρησιμοποιεί την διαδικασία απομόνωσης (process isolation), μια διεργασία μπορεί ακόμη να έχει κάποια περιορισμένη και ελεγχόμενη αλληλεπίδραση με κάποια άλλη διεργασία. Για να γίνει όμως η επικοινωνία μεταξύ των δύο διεργασιών επιτρεπτή, θα πρέπει και οι δύο διεργασίες να αποδεχτούν από κοινού την συνεργασία-επικοινωνία τους διαμέσου καναλιών επικοινωνίας (inter-process communication ή αλλιώς IPC channels).Τέτοια κανάλια μπορεί να είναι η κοινή μνήμη του συστήματος για τις διεργασίες, διαμέσου τοπικών υποδοχών ή διαμέσου του Διαδικτύου

(Curry, D. A. : 1992). Σε αυτό το σχήμα, το σύνολο σχεδόν της μνήμης διαδικασιών, είναι εξολοκλήρου απομονωμένο από άλλες διεργασίες, εκτός της μνήμης-μεταβλητών που επιτρέπει την είσοδο πληροφοριών από τις συνεργαζόμενες διεργασίες.

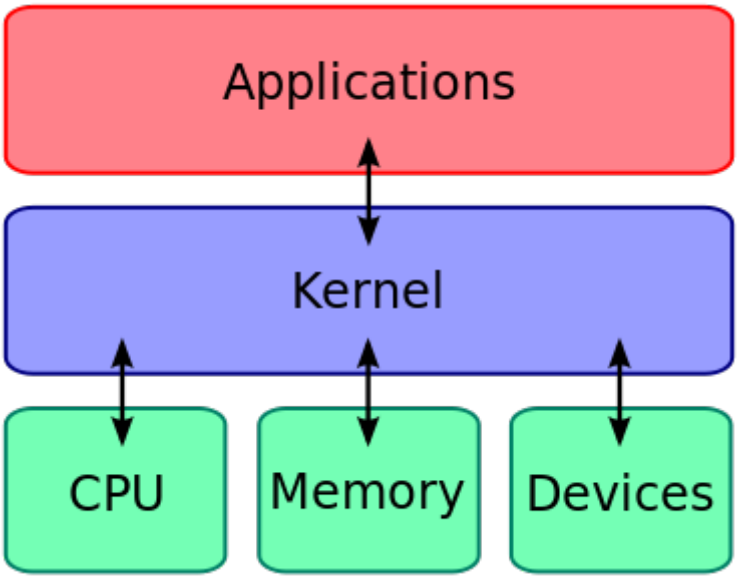


Σχήμα 3: *Επεξήγηση της λειτουργίας ενδοεπικοινωνίας ανάμεσα στις εφαρμογές*

4.1.4.Kernel

Στην πληροφορική με την ονομασία «kernel» ορίζουμε ένα πρόγραμμα για υπολογιστή που διαχειρίζεται τα αιτήματα των εισόδων και εξόδων από το λογισμικό και τα μεταφράζει σε εντολές επεξεργασίας δεδομένων για την κεντρική μονάδα επεξεργασίας καθώς και άλλων ηλεκτρονικών εξαρτημάτων του υπολογιστή. Το kernel σήμερα αποτελεί θεμελιώδης κομμάτι των σύγχρονων λειτουργικών συστημάτων. Το kernel εκτελεί τα καθήκοντα του, όπως για παράδειγμα, η εκτέλεση των διαδικασιών και ο χειρισμός καταστάσεων μέσα στις οποίες διακόπτονται λειτουργίες μιας διαδικασίας σε ένα χώρο του kernel (kernel space), ενώ οτιδήποτε μπορεί ένας χρήστης να κάνει σε καθημερινή βάση όπως τηλεφωνήματα ή την σύνταξη ενός κειμένου, με ένα πρόγραμμα επεξεργασίας κειμένου, γίνονται στον χώρο του χρήστη (user space). Αυτός ο διαχωρισμός έχει γίνει με σκοπό τα δεδομένα του χρήστη να μην παρεμβάλλονται με τα δεδομένα του kernel με αποτέλεσμα το σύστημα να μην μειώνει την απόδοση του καθιστώντας το ίδιο το σύστημα πιο σταθερό. Μπορεί να ειπωθεί ότι το kernel κάνει τρεις βασικές διεργασίες, που αφορούν τον επεξεργαστή, την μνήμη και την είσοδο και έξοδο διαφόρων συσκευών. Ο επεξεργαστής, για παράδειγμα, είναι υπεύθυνος για την εκτέλεση των προγραμμάτων. Το kernel ουσιαστικά είναι ο διαμεσολαβητής ανάμεσα στο πρόγραμμα (λογισμικό) και στον επεξεργαστή (hardware). Δουλειά του kernel είναι να αποφασίζει ανά πάσα στιγμή ποιο ή ποιες διεργασίες θα εκτελεσθούν, με ποια σειρά

θα εκτελεσθούν και πως θα κατανεμηθούν ανά πυρήνα (συνήθως κάθε πυρήνας μπορεί να εκτελεί ένα πρόγραμμα κάθε φορά). Επίσης το kernel ελέγχει την κατανομή της μνήμης ενός συστήματος. Σε ένα λειτουργικό σύστημα πολλά διαφορετικά προγράμματα συνήθως ζητάνε άδεια για να διαχειριστούν την μνήμη του συστήματος, ενώ τείνουν να ζητάνε περισσότερη μνήμη από ότι το ίδιο το σύστημα μπορεί να προσφέρει. Δουλειά του kernel είναι να αποφασίζει ποια πεδία μνήμης μπορεί κάθε διεργασία να αξιοποιήσει, καθώς και να βρίσκει λύσεις όταν το σύστημα δεν έχει να προσφέρει την ποσότητα μνήμης που ένα πρόγραμμα ζητάει. Τέλος συγχρονίζει και διαχειρίζεται συσκευές εισόδου όπως οθόνες, πληκτρολόγια κ.ο.κ. δημιουργώντας κανάλια επικοινωνίας μεταξύ των συσκευών και των προγραμμάτων (IPC).

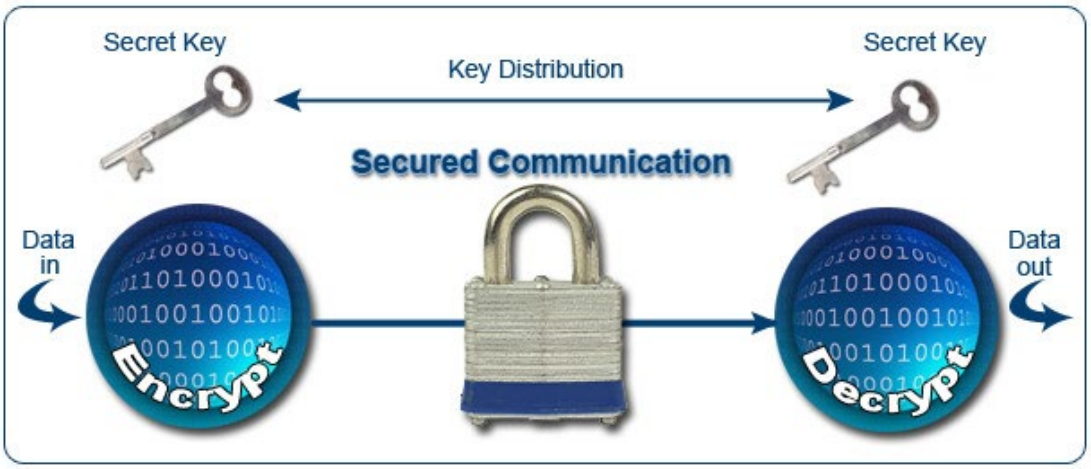


Σχήμα 4: *Αλληλοσύνδεση μεταξύ των εφαρμογών, του kernel και του hardware.(http://osarena.net/sites/default/files/old-wp/2013/04/kernel.png)*

4.1.5. Cryptography

Ο όρος κρυπτογραφία προέρχεται από τις ελληνικές λέξεις κρυπτός που σημαίνει μυστικός ή κρυφός καθώς και την λέξη γραφείν που σημαίνει γράφω. Η κρυπτογραφία είναι η τεχνολογία προστασίας πληροφοριών η οποία, μετατρέποντας (encrypting) την πληροφορία σε διάφορες μορφές μη αναγνώσιμες προς κάποιο μη εξουσιοδοτημένο χρήστη προσφέρει την δυνατότητα απόκρυψης και προστασίας των πληροφοριών από τρίτους δίνοντας την δυνατότητα μόνο σε εξουσιοδοτημένους χρήστες οι οποίοι έχουν αυτό που ονομάζεται μυστικό κλειδί την δυνατότητα να αποκρυπτογραφήσουν την συγκεκριμένη πληροφορία (decrypting). Σκοπός της

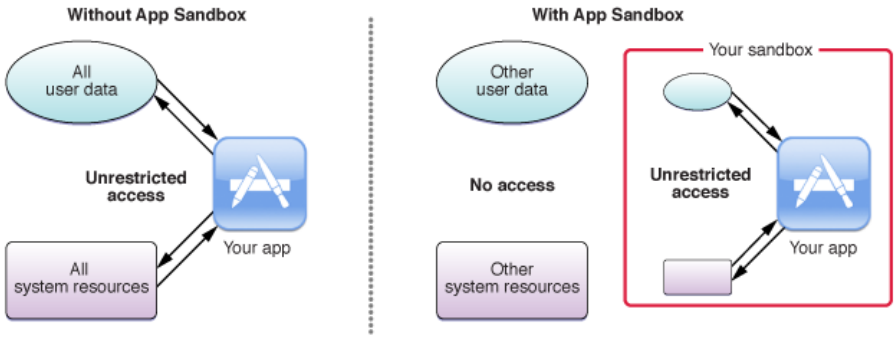
κρυπτογραφίας είναι η προστασία πληροφοριών κυρίως από ανεπιθύμητους αναγνώστες στην αυθαίρετη ανάγνωση και τροποποίηση προστατευμένων και ευαίσθητων πληροφοριών.



Σχήμα 5: *Συμμετρική κρυπτογράφηση δεδομένων.*
(http://www.nucrypt.net/images/encrypt_overview.jpg)

4.1.6. Sandboxing

Με τον όρο «Sandbox» στην ασφάλεια πληροφοριακών συστημάτων ορίζεται ο μηχανισμός ασφάλειας για τον διαχωρισμό των προγραμμάτων που εκτελούνται (Curry, D. A. : 1992). Συνηθίζεται να χρησιμοποιείται για κώδικα που δεν έχει δοκιμαστεί και δε υπάρχει γνώση για τις επιπτώσεις του σε ένα σύστημα, για μη εγκεκριμένες εφαρμογές (3rd party applications) καθώς και για χρήστες που δεν έχουν κάποιο δικαίωμα χρήσης (untrusted users). Το Sandbox παρέχει ένα αυστηρά ελεγχόμενο διαμοιρασμό των πόρων του συστήματος για προγράμματα που φιλοξενούνται στο σύστημα.



Σχήμα 6 : Λειτουργία των εφαρμογών μέσα στα πλαίσια του Sandbox και εκτός.[18]

4.2. Rooting

Με τον όρο «Rooting» ορίζουμε την διαδικασία που επιτρέπει στους χρήστες έξυπνων κινητών, Tablet και γενικά όποιων άλλων συσκευών χρησιμοποιούν το λογισμικό «Android Mobile Operating System» να αποκτήσουν πρόσβαση με προνόμια χρήστη γνωστό και ως "root access". Το λογισμικό Android χρησιμοποιεί Linux kernel, το οποίο προσδίδει κοινές ιδιότητες στο μοντέλο ασφάλειας του λογισμικού με το UNIX. Αναφορικά, το μοντέλο του UNIX προσδίδει διαφορετικές άδειες ανά χρήστη, μερική ή ολική απομόνωση των διαφορετικών διαδικασιών ανά εφαρμογή και ανά χρήστη, μηχανισμούς επικοινωνίας μεταξύ των διαφόρων διαδικασιών, καθώς και δυνατότητα τροποποιήσεις του kernel ώστε να μπορεί να προσαρμοστεί στις ανάγκες του λειτουργικού συστήματος (Curry, D. A. : 1992). Το ίδιο μοντέλο μπορεί να βρεθεί και στο λειτουργικό σύστημα των Linux καθώς και σε οποιοδήποτε άλλο λειτουργικό σύστημα χρησιμοποιεί στοιχεία από UNIX όπως το λογισμικό FreeBSD και το OSX της Apple. Ο κύριος λόγος που ένας χρήστης θα ξεκλειδώσει ή αλλιώς θα κάνει root την συσκευή του είναι για να σπάσει τους φραγμούς που η κατασκευάστρια εταιρία ή πολλές φορές ο πάροχος των συγκεκριμένων συσκευών βάζει με σκοπό να περιορίσει και ταυτόχρονα να προστατέψει τον χρήστη.

4.2.1 Τα οφέλη του Rooting

Συγκεκριμένα το rooting επιτρέπει: α) την αλλαγή εξολοκλήρου του λογισμικού που έχει εγκατασταθεί στην συσκευή, β) την χρησιμοποίηση εφαρμογών οι οποίες για να λειτουργήσουν χρειάζονται δικαιώματα διαχειριστή (administrator-level permissions), γ)

18 *https://developer.apple.com/library/mac/documentation/Security/Conceptual/AppSandboxDesignGuide/Art/about_sandboxing.png*

την αύξηση των αποδόσεων του επεξεργαστή, δ) την τροποποίηση των τάσεων του ρεύματος που χρησιμοποιεί η συσκευή, ε) την τροποποίηση ή και την δυνατότητα διαγραφής αρχείων του συστήματος, στ) τη διαγραφή εφαρμογών που ο πάροχος έχει εγκαταστήσει και οι οποίες δε διαγράφονται και τέλος τη διαχείριση χαμηλού επιπέδου προσβάσιμων στοιχείων του λειτουργικού συστήματος, όπως ο επανακαθορισμός των εισόδων αφής. Γενικά, είναι ορθό να οριστεί το rooting ως η δυνατότητα του χρήστη να διαχειριστεί με πλήρη ελευθερία την συσκευή του, διαμέσου της απόκτησης δικαιωμάτων διαχειριστή.

4.2.2. Η διαδικασία του Rooting

Το rooting βέβαια αποτελούσε μια ιδιαίτερα εξειδικευμένη ενέργεια που δε μπορούσε να εκτελέσει ο απλός χρήστης. Με το πέρασμα των χρόνων η διαδικασία αυτή έχει γίνει απλούστερη επιτρέποντας και χρήστες χωρίς καθόλου γνώσεις σε Unix συστήματα να «ρουτάρουν» την συσκευή τους με μεγάλο ποσοστό επιτυχίας. Αναφορικά το 2012-2013 για να «ρουτάρει» κάποιος χρήστης την συσκευή του έπρεπε να έχει πρόσβαση στο Διαδίκτυο για να κατεβάσει την εφαρμογή στον υπολογιστή του, η οποία θα έκανε root την Android συσκευή, με αποτέλεσμα ο χρήστης να αποκτήσει δικαιώματα διαχειριστή, ενώ θα έπρεπε να έχει βασικές γνώσεις αρχιτεκτονικής του λειτουργικού ώστε να μπορεί να εκμεταλλευτεί τις δυνατότητες που προσφέρει το root, στο έπακρο. Η σελίδα xda-developer.com αυτή τη στιγμή προσφέρει οδηγίες για rooting και για εγκαταστάσεις custom roms για πάνω από εκατόν πενήντα συσκευές της αγοράς συμπεριλαμβανομένων των ναυαρχίδων κινητών όλων των εταιριών. Σήμερα, η διαδικασία έχει γίνει πολύ απλή και δε χρειάζεται πλέον να συνδεθεί η συσκευή στον υπολογιστή παρά μόνο στο Διαδίκτυο όπου ο χρήστης θα κατεβάσει μια εφαρμογή η οποία σε ελάχιστο χρόνο θα προσδώσει στον χρήστη, δικαιώματα διαχειριστή και θα ξεκλειδώσει και το δικαίωμα εγκατάστασης άλλης έκδοση λογισμικού (unlock bootloader). Την εφαρμογή αυτή προσφέρουν εταιρίες όπως η Cyanogenmod που δημιουργεί διάφορες παραλλαγές του λειτουργικού Android. Το παράδοξο βέβαια είναι ότι την εφαρμογή αυτή μπορεί κάποιος να την κατεβάσει δωρεάν από το ηλεκτρονικό κατάστημα της Google. Τέλος, με την απόκτηση δικαιωμάτων διαχειριστή, ο χρήστης κατεβάζει αυτόματα με την ολοκλήρωση της διαδικασίας του root στην συσκευή του μια εφαρμογή (superuser app) η οποία λειτουργεί ως διαμεσολαβητής ανάμεσα στον χρήστη και στο λειτουργικό ώστε να κάνει κλήσεις (System Calls) σε εφαρμογές με δικαιώματα διαχειριστή. Παρόλο που το rooting στα πλαίσια του κινήματος ελεύθερου ανοιχτού

πηγαίου κώδικα θεωρείται επιτρεπτό, πολλές εταιρίες καλύπτουν στην εγγύηση των συσκευών τους, συσκευές οι οποίες έχουν τροποποιηθεί. Βέβαια η πλήρη άγνοια σε βασικές γνώσεις του λειτουργικού συστήματος του Android καθώς και οι διαφορές ανάμεσα στις συσκευές μπορεί να οδηγήσει τον χρήστη στην προσπάθεια του να «ρουτάρει» την συσκευή του, στο να την καταστρέψει (Brick).

4.2.3. Κίνδυνοι του Rooting

Όλες αυτές οι εφαρμογές που δίνουν δικαιώματα διαχειριστή καθώς και όλες οι παραλλαγές των διαφορετικών μη αυθεντικών λειτουργικών συστημάτων βασισμένες στον κώδικα της Google ενέχουν κινδύνους τόσο ασφάλειας του λειτουργικού συστήματος, όσο και σωματικών βλαβών. Παραδείγματος χάρη, να ανέβει η τάση ρεύματος που δέχεται η μπαταρία, με αποτέλεσμα η μπαταρία είτε να καταστραφεί είτε να εκραγεί στην τσέπη του ιδιοκτήτη. Το rooting είναι μια διαδικασία που ο χρήστης πολλές φορές χωρίς να γνωρίζει τις επιπτώσεις και τους κινδύνους που κρύβονται πίσω από την απόλυτη ελευθερία χρήσης, θα δοκιμάσει στην προσωπική του συσκευή. Χρήζει προσοχής ότι το rooting δε μπορεί να καταγραφεί, ούτε και υπάρχουν γνωστοί τρόποι ώστε η Google να ελέγξει τους χρήστες της με σκοπό να τους απαγορεύσει να «ρουτάρουν» τις συσκευές τους.

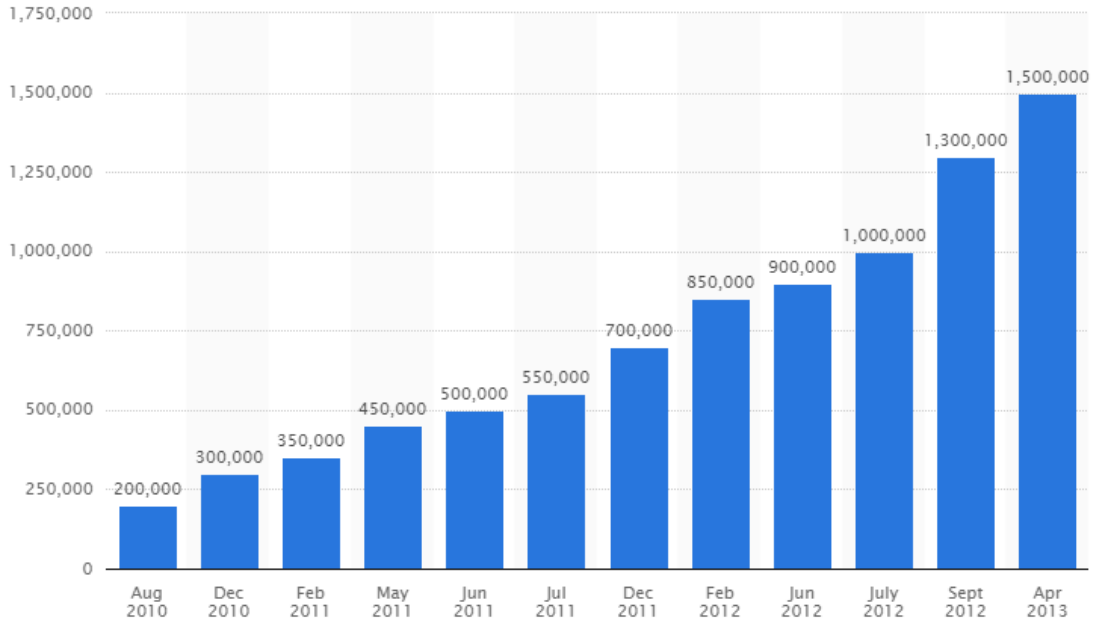
4.2.4. Έλεγχος του Rooting

Ο μόνος τρόπος να ελεγχθεί το root είναι στατιστικά, βάσει των συσκευών που χρησιμοποιούν Android και τα οποία είναι αφενός ενεργά και αφετέρου έχουν υποστεί κάποιο είδος τροποποίησης, καθώς και πόσες φορές έχουν κατεβάσει οι χρήστες από το ηλεκτρονικό κατάστημα της Google εφαρμογές που λειτουργούν μόνο με δικαιώματα διαχειριστή. Στο πρώτο πίνακα που παραθέτεται, φαίνεται πόσες φορές έχει κατεβάσει κάποιος από την σελίδα της Cyanogenmod το τροποποιημένο λειτουργικό τους σύστημα.

Type	Total
Official Installs	5,528,273
Unofficial Installs	4,485,423
Total Installs	**10,013,696**
Last 24 Hours	12,430

Εικόνα 2: Επίσημα στοιχεία διανομής του OS της Cyanogenmod (Androidcommunity.com)

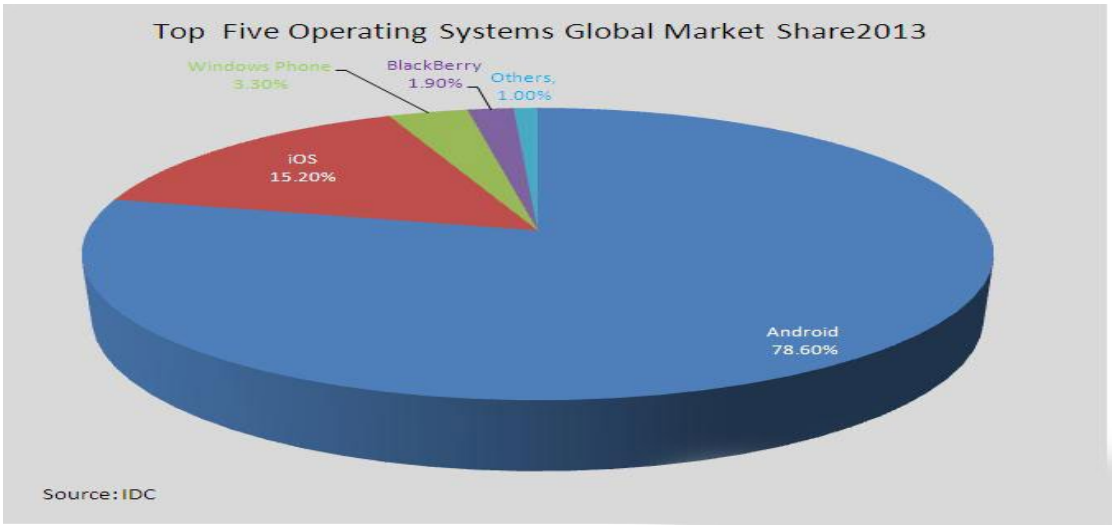
Εν συνεχεία παραθέτεται ένα γράφημα με στατιστικά στοιχεία βασισμένα στις καθημερινές ενεργοποιήσεις κινητών με Android από την ίδια την Google.



Πίνακας 7: Μηνιαίες ενεργοποιήσεις του λειτουργικού Android κατά την περίοδο 2010-2013[19]

Επιπρόσθετα, ο ιδρυτής και CEO της εταιρίας της Google ο Larry Page στο ετήσιο συνέδριο της Google (Google I/0 2013), ανέφερε ότι από το 2008 που δόθηκε για πρώτη φορά στο ευρύ κοινό το λειτουργικό σύστημα, μέχρι το 2013 έχουν πάνω από 900 εκατομμύρια ενεργοποιήσεις κινητών με το λειτουργικό τους σύστημα. Σήμερα ο αριθμός βέβαια είναι πολύ μεγαλύτερος μιας και το 78.6% της αγοράς στις κινητές συσκευές χρησιμοποιεί συσκευή με λειτουργικό android , όπως φαίνεται και παρακάτω.

[19]Source:Google.Android.com



Πίνακας 8: Προτιμήσεις κοινού ως προς την αγορά συσκευών με βάση τα λειτουργικά συστήματα τους.(IDC)

Επιπλέον, εφαρμογές όπως το Titanium Backup, το Rom manager και το Set CPU που χρειάζονται δικαιώματα διαχειριστή βρίσκονται στην θέση 3, 4 και 9 των πιο δημοφιλών εφαρμογών που κατεβάζει ο χρήστης "επί πληρωμή" με το καθένα να τους να έχει πάνω από 10 εκατομμύρια downloads στο ηλεκτρονικό κατάστημα. Συγχωνεύοντας όμως όλες τις εφαρμογές αυτές μαζί καθώς και τους χρήστες οι οποίοι κατεβάζουν και χρησιμοποιούν κάποιο τροποποιημένο λογισμικό της Google, έχοντας βέβαια στο νου ότι υπάρχουν πάνω από 900 εκατομμύρια συσκευές ενεργές που χρησιμοποιούν το λειτουργικό Android εικάζουμε ότι μόνο το 0.5-2.5% των χρηστών «ρουτάρουν» την συσκευή τους. Ποσοστό αρκετά ενθαρρυντικό ως προς τον τομέα της ασφάλειας, εάν βέβαια αληθεύει. Εν κατακλείδι, η ασφάλεια των προσωπικών δεδομένων εξαρτάται τόσο από το λειτουργικό σύστημα όσο κι από τους ίδιους τους χρήστες. Εάν υπάρξει κάποιος ιός μέσα στο λειτουργικό σύστημα επειδή τροποποιήθηκε κάποια λειτουργία του κινητού που δεν επιτρεπόταν από τον κατασκευαστή να τροποποιηθεί ή εγκαταστάθηκε κάποια παραλλαγή του λειτουργικού συστήματος με αρχεία που κατέβηκαν από το Διαδίκτυο και τα οποία δείχνοντας τυφλή εμπιστοσύνη εμπεριείχαν κάποιο κομμάτι κώδικα που είτε εμπεριείχε keylogger[20] και έστελνε κωδικούς, τηλέφωνα, στοιχεία πιστωτικής κάρτας κ.ο.κ ή έστελνε μηνύματα σε άγνωστα νούμερα ή ακόμη έστελνε την τοποθεσία της συσκευής σε κάποιο χρήστη χωρίς την άδεια του ιδιοκτήτη της συσκευής, τότε η ευθύνη ανήκει στον χρήστη και όχι στην εταιρία. Είναι σημαντικό να αναφερθεί

[20] Keylogger: Είναι ένα κομμάτι κατασκοπευτικού λογισμικού (μπορεί να θεωρηθεί είτε λογισμικό είτε Spyware) το οποίο έχει την δυνατότητα να καταγράφει τις πληκτρολογήσεις του χρήστη.

ότι τόσο το rooting όσο και το jailbreak που θα αναλυθεί παρακάτω πυροδότησε την αρχή μιας νέας αγοράς κινητών τηλεφώνων.

4.2.5. Νέες μορφές αγορών χάρη στο Rooting

Το κίνημα του Open source έγινε ισχυρότερο και πλέον ικανοποιεί μια ομάδα καταναλωτών οι οποίοι αγοράζουν κινητές συσκευές με σκοπό να τις "ρουτάρουν" δημιουργώντας νέες εφαρμογές, με νέες προϋποθέσεις, χωρίς όρια με μόνο γνώμονα την φαντασία τους. Παράλληλα, χάρη στο rooting δημιουργήθηκαν εταιρίες όπως η Cyanogenmod και αναπτύχθηκαν διάφορα μοντέλα λογισμικού βασισμένα στο Android δημιουργώντας λογισμικό για συσκευές εταιριών κολοσσών στο χώρο των κινητών συσκευών που σε άλλη περίπτωση δε θα μπορούσαν να δοκιμάσουν και ταυτόχρονα να διαμοιράσουν το προϊόν τους. Εν κατακλείδι το rooting αφαιρείται αυτόματα με την αναβάθμιση του λογισμικού. Μόνο σε περίπτωση που η συσκευή χρησιμοποιεί custom rom τότε χρειάζεται να διαγραφούν όλα τα αρχεία και να εγκατασταθεί το αυθεντικό λογισμικό με τα αρχεία από το AOSP που δίνει η Google ή εάν υπάρχει διαθέσιμο εικονικό backup του αυθεντικού λογισμικού, να το επαναφερθεί διαγράφοντας τη custom rom.

4.3. Jailbreak

Μέχρι στιγμής αναλύθηκε το rooting για τις Android συσκευές, τα πλεονεκτήματα και τα μειονεκτήματα που αναφέρθηκαν. Το «Jailbreak» παρόλο που σαν όρος και σαν διαδικασία προηγήθηκε του rooting (rooting 2013 - jailbreak 2008), γίνεται για περίπου τους ίδιους λόγους με αυτούς που προαναφέρθηκαν. Το Jailbreak που στα ελληνικά μεταφράζεται ως «δραπετεύω από την φυλακή» ξεκίνησε για αυτόν ακριβώς τον λόγο. Να δραπετεύσει ο χρήστης από τα δεσμά που βάζει η Apple στο λογισμικό της ώστε να πειραματιστεί αφενός σε νέες εφαρμογές με νέες δυνατότητες αξιοποιώντας πλήρως τόσο το λογισμικό όσο και το υλικό της συσκευής. Από την άλλη, για τους ερευνητές ασφαλείας το Jailbreak αποτέλεσε ανάγκη για έρευνα μιας και το λογισμικό από την ίδια την Apple είναι κλειδωμένο με πολλά επίπεδα ασφαλείας που δεν επιτρέπει κώδικα χωρίς ψηφιακή υπογραφή της Apple στην συσκευή τους και μαζί με το sandbox που ουσιαστικά εμποδίζει το δίαυλο επικοινωνίας ανάμεσα στις εφαρμογές καθιστά σχεδόν αδύνατο το debug οποιασδήποτε εφαρμογής. Αυτό βέβαια αποτελεί εμπόδιο στους ερευνητές ασφαλείας οι οποίοι αξιολογούν τα επίπεδα ασφάλειας του λογισμικού και των συσκευών στο σύνολο τους. Το Jailbreak, είναι η διαδικασία άρσης των περιορισμών τόσο του υλικού (hardware) όσο και του software στο λογισμικό iOS (Dai Zovi,D : 2011).

Επειδή και το iOS όπως και το λογισμικό Android είναι βασισμένα στο καλά μελετημένο μοντέλο ασφαλείας του Unix, το Jailbreak έχει πολλά κοινά στοιχεία και αποτελέσματα με το rooting για τα android. Η διαδικασία μέσω εφαρμογών, με πιο γνωστό το επονομαζόμενο redsn0w app, κάνει jailbreak την συσκευή iOS, δίνοντας στον χρήστη δικαιώματα-προνόμια διαχειριστή (root access) για το iOS αρχείο του συστήματος και για τα αρχεία διαχείρισης της συσκευής επιτρέποντας το «κατέβασμα» και την εγκατάσταση διαφόρων εφαρμογών, επεκτάσεις λογισμικού (extensions)[21] καθώς και θεμάτων που δεν μπορεί ο χρήστης να βρει στο ηλεκτρονικό κατάστημα της Apple. Θα μπορούσε κανείς να πει ότι το Jailbreak είναι μια μορφή κλιμάκωσης προνομιών (Serion,N : 2010).

4.3.1. Περιορισμοί του Jailbreak

Έχοντας βέβαια κάνει ο χρήστης jailbreak την συσκευή του, δε σημαίνει ότι δε μπορεί να συνεχίζει να κατεβάζει εφαρμογές από την apple ούτε ότι δε θα μπορεί να χρησιμοποιεί το iTunes και γενικά οποιαδήποτε εφαρμογή δίνει η apple για χρήση στις συσκευές της. Ένας ακόμη λόγος που ο χρήστης κάνει Jailbreak είναι πιθανώς για να παρακάμψει το φράγμα των παρόχων που κλειδώνουν τις συσκευές (ultrasn0w) ώστε να μπορούν να χρησιμοποιηθούν μόνο στο δικό τους δίκτυο. Στην περίπτωση της Apple όμως αποτελεί αναγκαίο κακό το ξεκλείδωμα του κινητού διαμέσου του jailbreak μιας και η Apple ακόμη και με κωδικό προγραμματιστή από την ίδια την Apple δε σου επιτρέπει να δοκιμάσεις κώδικα στις συσκευές της. Παρόλο που οι χρήστες έχουν καταφέρει να κάνουν jailbreak τις συσκευές τους, για πολλά χρόνια πολλές από τις διαφορετικές εκδόσεις του iOS δίνουν στο jailbreak νέες δυνατότητες και νέα χαρακτηριστικά. Ο κύριος λόγος, είναι η ποιότητα του jailbreak που εξαρτάται σχεδόν εξολοκλήρου στις ευπάθειες του ίδιου του λογισμικού. Συνήθως ευπάθειες που αξιοποιούνται από τους χρήστες μια φορά, γίνονται άμεσα γνωστές στην Apple και συνήθως με μεγάλη ταχύτητα της διορθώνει κλείνοντας τις λεγόμενες δικλείδες ασφαλείας στις επόμενες αναβαθμισμένες εκδόσεις του λειτουργικού (Serion,N : 2010). Για το λόγο αυτό για κάθε νέο αναβαθμισμένο λογισμικό που παρουσιάζει η Apple χρειάζονται να βρεθούν νέες ευπάθειες ώστε να επιτύχουν οι χρήστες το jailbreak.

[21] Επεκτάσεις λογισμικού (Extensions): ως επέκταση λογισμικού ορίζεται ένα κομμάτι λογισμικού που διευρύνει τις ικανότητες του βασικού λογισμικού.Ο όρος αυτός πολλές φορές ταυτίζεται λανθασμένα με τον όρο Plug-in.

4.3.2. Διαφορετικές κατηγορίες Jailbreak

Στο σημείο αυτό, έχοντας ήδη αναφερθεί στην ανάγκη να βρεθεί κάποια ευπάθεια στο λειτουργικό σύστημα ώστε να υπάρξει επιτυχημένο ξεκλείδωμα της συσκευής, πρέπει να διευκρινιστεί ότι ανάλογα με την ευπάθεια που βρίσκουν κάθε φορά οι χρήστες στο λειτουργικό εξαρτάται εξολοκλήρου και η ποιότητα του jailbreak έχοντας ως αποτέλεσμα την διάρκεια που θα μπορεί ο χρήστης να κρατήσει μια συσκευή με jailbreak. Γι' αυτό το λόγο το Jailbreak διαχωρίζεται σε δύο βασικές κατηγορίες. Το λεγόμενο Tethered Jailbreak και το Untethered jailbreak.

4.3.3 Tethered Jailbreak

Το Tethered Jailbreak είναι ουσιαστικά ένα Jailbreak το οποίο θα εξαφανισθεί με την επανεκκίνηση της συσκευής. Η συσκευή, που έχει υποστεί Jailbreak χρειάζεται κάποια μορφής επανατροποποίησης-rejailbreak μετά από κάθε επανεκκίνηση. Αυτό κυρίως σημαίνει ότι η συσκευή πρέπει να είναι συνδεμένη στον ηλεκτρονικό υπολογιστή κάθε φορά που η συσκευή κλείνει ή ανοίγει. Εξαιτίας της χρήσης καλωδίων USB κάθε φορά που ενεργοποιείται η συσκευή βγαίνει και ο όρος Tethered. Με πιο τεχνικούς όρους εάν η ευπάθεια βρεθεί σε κάποιο σημείο κάποιου κώδικα που ενέχει προνόμια , ένα tethered Jailbreak θα μπορούσε να αποτελείται από την εκμετάλλευση μιας μόνο ευπάθειας. Ένα απλό παράδειγμα είναι το limera1n bootrom exploit που χρησιμοποιείται κυρίως για το Jailbreak των εκδόσεων iOS 4 και iOS 5 του λογισμικού. Ένα ακόμη παράδειγμα εκμετάλλευσης ευπαθειών ώστε να αποκτήσει ο χρήστης προνόμια διαχειριστή θα ήταν να βρει μια ευπάθεια στο kernel driver του iOS για την USB σύνδεση στον υπολογιστή. Βέβαια μια τέτοια ευπάθεια ακόμη δεν έχει βρεθεί μέχρι και σήμερα.

4.3.4. Untethered Jailbreak

Από την άλλη υπάρχει και το λεγόμενο Untethered Jailbreak που ορίζεται ως μια κατάσταση εκμετάλλευσης μια επίμονης ευπάθειας του λογισμικού που δε θα χαθεί ακόμη και αν η συσκευή κάνει επανεκκίνηση. Ονομάζεται Untethered ακριβώς επειδή δε χρειάζεται κάθε φορά που η συσκευή κάνει επανεκκίνηση να κάνει ο χρήστης ξανά jailbreak. Για το λόγο αυτό μπορεί κανείς με ασφάλεια να ονομάσει το Untethered Jailbreak την καλύτερη δυνατή μορφή απόκτησης δικαιωμάτων διαχειριστή. Το bootchain ή στα Ελληνικά η αλυσίδα εκκίνησης είναι οι διάφορες διαδικασίες που γίνονται κάθε φορά που η συσκευή ανοίγει (μπορεί να παρομοιαστεί με το Bios του προσωπικού υπολογιστή). Είναι άρα λογικό να ειπωθεί ότι το Untethered jailbreak είναι αρκετά

δυσκολότερο στο να επιτευχθεί μιας και πρέπει να βρεθούν ευπάθειες σε πολύ συγκεκριμένα σημεία της αλυσίδας εκκίνησης του λειτουργικού.

4.3.5.Ανασκόπηση του Jailbreak

Στο παρελθόν ήταν πολύ ευκολότερο να βρεθεί κάποια ευπάθεια μιας και στις πρώιμες εκδόσεις του λειτουργικού υπήρχαν πολλές και μεγάλες ευπάθειες στο λειτουργικό επιτρέποντας την εκμετάλλευση τους από τους χρήστες. Όσο όμως το λειτουργικό ωριμάζει τόσο αυτές οι ευπάθειες χάνονται. Με την εξαφάνιση αυτών των ευπαθειών και με την συνεχή αναβάθμιση του μοντέλου ασφάλειας του λογισμικού, γίνεται πιο δύσκολη η διαδικασία του Jailbreak. Η Apple σε σύγκριση με την Google είναι μια εταιρία που κλείνει τον πηγαίο κώδικα της, παρόλο που δίνει κάποια από τα Repos[22] , όπως ονομάζει τον κώδικα της, ελεύθερο στο κοινό, συνηθίζει να τα ανεβάζει με μεγάλη καθυστέρηση πολλές φορές και δύο εκδόσεις πίσω. Αυτή την στιγμή παρόλο που το λογισμικό βρίσκεται στην έκδοση 8.3, η Apple δεν ανεβάζει στην επίσημη σελίδα της τα Repos του νέου λογισμικού, ενώ πρέπει να σημειωθεί ότι έχει σταματήσει στην έκδοση 6 του iOS. Με στοιχεία που παρατέθηκαν παραπάνω στον πίνακα 3 φαίνεται ξεκάθαρα ότι η Apple και το λογισμικό iOS διατηρούν ένα ποσοστό στην αγορά της τάξεως του 16% κάνοντας το αυτόματα το δεύτερο πιο δημοφιλές λογισμικό της αγοράς. Το Jailbreak βέβαια είναι μια πιο δύσκολη διαδικασία όπως αναφέρθηκε παραπάνω από το Rooting και γι' αυτό αξίζει να δοθεί ιδιαίτερη προσοχή στα ποσοστά των χρηστών που κάνουν Jailbreak την συσκευή τους. Στο παρακάτω post ενός πολύ μεγάλου Hacker εν ονόματι Cyril Cattiax ο οποίος έχει ανακαλύψει πολλές ευπάθειες στο bootrom[23] του λογισμικού iOS αναφέρει ότι 14,051,500 μοναδικές συσκευές χρησιμοποιούν το app Cydia σε συσκευές iOS 6.x εκδόσεις ενώ είκοσι τρία εκατομμύρια μοναδικοί χρήστες χρησιμοποιούν την εφαρμογή Cydia ανεξαρτήτως εκδόσεων.

[22] Software Repository (Repos): Είναι ο αποθηκευτικός χώρος στον οποίο αποθηκεύονται πακέτα λογισμικού και τα οποία μπορούν να χρησιμοποιηθούν στον ηλεκτρονικό υπολογιστή, έξυπνη συσκευή κ.ο.κ.

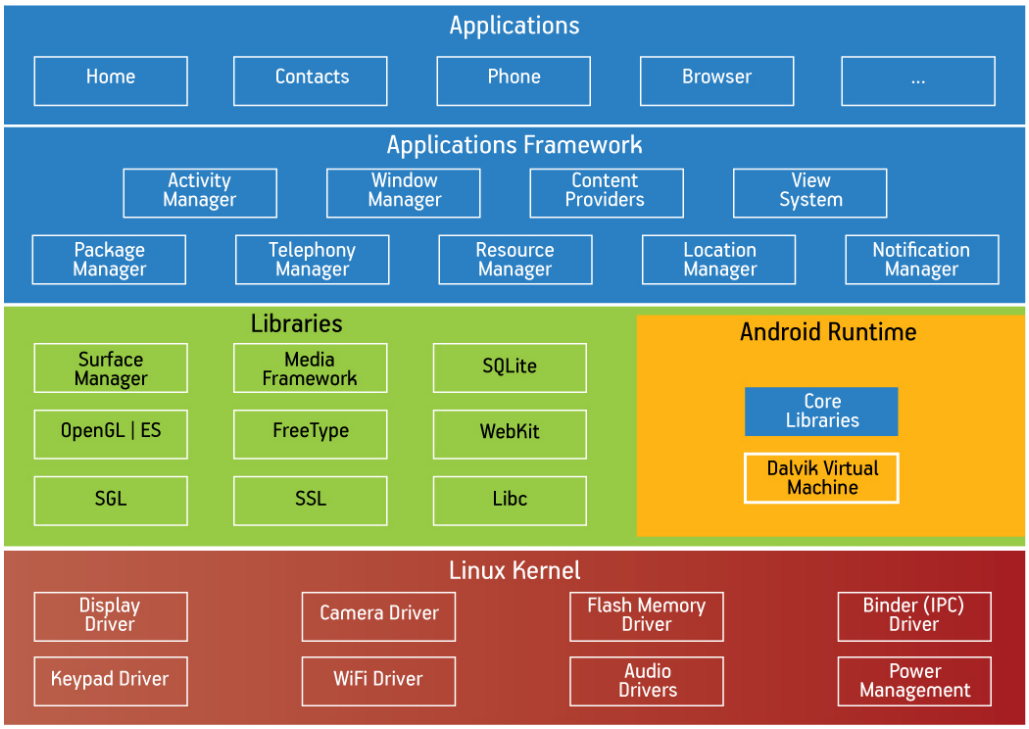
[23] Bootroms:Είναι ο πρώτος μηχανισμός που «τρέχει» κατά την εκκίνηση της συσκευής. Το Bootrom έχει μόνο δυνατότητα ανάγνωσης. Αν βρεθεί ευπάθεια σε επίπεδο Bootrom θεωρείται μεγάλη επιτυχία μιας και η Apple για να κλείσει τα κενά ασφαλείας θα πρέπει να βγάλει ολόκληρο Patch που σχετίζεται με το Hardware και συνήθως είναι αρκετά χρονοβόρο.

Εικόνα 3: Tweet από έναν από τους πρώτους χάκερ που έκανε Jailbreak, iOS συσκευή[24]

[24] *Twitter.com/prod2g*

5. Ο σχεδιασμός και η αρχιτεκτονική του μοντέλου ασφάλειας του Android

Το λειτουργικό σύστημα Android βασίζεται σε πολλαπλούς καλά μελετημένους ελεγκτικούς μηχανισμούς που εναρμονίζονται μεταξύ τους με σκοπό τον έλεγχο και την ενδυνάμωση της ασφάλειας του λογισμικού, για την προστασία των προσωπικών δεδομένων του χρήστη. Όπως όλα τα σύγχρονα λειτουργικά συστήματα, πολλοί από αυτούς τους μηχανισμούς επικοινωνούν/αλληλεπιδρούν μεταξύ τους, ανταλλάσσοντας πληροφορίες για διάφορα θέματα όπως εφαρμογές και χρήστες, αντικείμενα (άλλες εφαρμογές, αρχεία, συσκευές), καθώς και διεργασίες/λειτουργίες οι οποίες πρέπει να πραγματοποιηθούν όπως η ανάγνωση, εγγραφή και διαγραφή ενός αρχείου (Brunette, E : 2009). Πολλές φορές η επιβολή αυτών των μηχανισμών ελέγχου και προστασίας μπορεί να υπάρχουν και να λειτουργούν στην καρδιά του λογισμικού χωρίς να συμβεί κάποιο περιστατικό. Αποτελούν, βέβαια την ασπίδα προστασίας του λογισμικού συστήματος απέναντι σε κακόβουλες επιθέσεις προστατεύοντας συνεχώς και αδιαλείπτως τον χρήστη από κακόβουλες επιθέσεις και σενάρια εκμετάλλευσης/κατάχρησης της συσκευής του από μη εξουσιοδοτημένους χρήστες. Η γενική αρχιτεκτονική του λειτουργικού Android, έχει πολλές φορές παρομοιαστεί ως "Java για Linux" (Drake,J : 2014). Ωστόσο, η παρομοίωση-ορισμός της πλατφόρμας είναι ολίγον παραπλανητική, αφού δεν αποδίδει με απόλυτη δικαιοσύνη την πολυπλοκότητα και την αρχιτεκτονική της πλατφόρμας. Η αρχιτεκτονική της πλατφόρμας Android, διαχωρίζεται σε πέντε βασικά στρώματα, που περιλαμβάνουν τις εφαρμογές Android, το Android Framework, την εικονική μηχανή (virtual machine) Dalvik, το user-space native code και το Linux kernel. Στον σχήμα παρακάτω εμφανίζεται η ολοκληρωμένη μορφή των πέντε βασικών στρωμάτων που σαν ολότητα διαμορφώνουν το λειτουργικό σύστημα Android. Το Linus kernel που βρίσκεται στον πυρήνα του λειτουργικού συστήματος, αποτελεί τον βασικό πυρήνα του μηχανισμού προστασίας των εφαρμογών (sandboxing), όπως θα αναλυθεί και στην συνέχεια. Η αρχιτεκτονική του λειτουργικού συστήματος βασίζεται στο μοντέλο ασφαλείας του UNIX όπως αναφέρθηκε στο κεφάλαιο 3, αλλά προσδίδει μεγάλες αλλαγές στο πηγαίο κώδικα του kernel, πολλές από τις οποίες έχουν δικούς τους μηχανισμούς ασφαλείας. Το κεφάλαιο αυτό, ερευνά τον σχεδιασμό και την αρχιτεκτονική του μοντέλου ασφαλείας του λειτουργικού συστήματος Android.



Πίνακας 9: Παρουσίαση των βασικών στρωμάτων που συντελούν το λειτουργικό σύστημα Android.(http://tutorials4android.com/wp-content/uploads/2015/03/Android-Architecture.jpg)

5.1.Κατανοώντας τα όρια ασφάλειας και την επιβολή μηχανισμών ελέγχου στο λειτουργικό Android.

Με τον όρο, όρια ασφαλείας, ορίζουμε συγκεκριμένες περιοχές του συστήματος, όπου το επίπεδο ασφαλείας και εμπιστοσύνης διαφέρει από λειτουργία σε λειτουργία. Ένα πολύ καλό παράδειγμα αποτελούν τα διακριτά όρια επικοινωνίας-ασφάλειας μεταξύ του χώρου όπου ενεργεί το kernel (kernel space) και του χώρου όπου ενεργεί ο χρήστης (user-space). Ο κώδικας στο χώρο όπου ενεργεί το kernel έχει το δικαίωμα να εκτελεί χαμηλού επιπέδου διεργασίες στο Hardware, ενώ μπορεί ταυτόχρονα να διαχειρίζεται όλη την φυσική μνήμη του συστήματος. Από την άλλη ο χώρος στον οποίο ενεργεί ο χρήστης, δε μπορεί να έχει πρόσβαση σε όλη την μνήμη του συστήματος εξαιτίας των ορίων που επιβάλει στο σύστημα η κεντρική μονάδα επεξεργασίας της συσκευής (CPU). Το λειτουργικό σύστημα Android χρησιμοποιεί δύο διακριτά, αλλά παράλληλα συνεργαζόμενα μοντέλα δικαιωμάτων. Στο χαμηλό επίπεδο όπως ονομάζεται, το Linux kernel επιβάλλει δικαιώματα χρήστη και ομάδων (users-group permissions). Αυτό το μοντέλο επιβολής δικαιωμάτων έχει κληρονομηθεί από το μοντέλο ασφάλειας των Linux, ενισχύοντας την πρόσβαση στο αρχείο των πόρων του συστήματος (file system entries), καθώς και σε άλλους ειδικούς πόρους του λειτουργικού συστήματος Android. Το μοντέλο αυτό αναφέρεται συνήθως ως το Sandbox του συστήματος Android. Το δεύτερο μοντέλο που είναι εκτεθειμένο στους χρήστες, όταν

εγκαθιστούν εφαρμογές, ορίζει όρια ασφαλείας μέσω διακριτών δικαιωμάτων, που περιορίζουν τις δυνατότητες των εφαρμογών. Κάποιες από τις άδειες από το δεύτερο μοντέλο που αναφέρθηκαν παραπάνω, διαχωρίζουν τις εφαρμογές σε συγκεκριμένου τύπου χρήστες (users), ανά ομάδες (group) καθώς και ανά δυνατότητες στο υποκείμενο λειτουργικό σύστημα.

5.2 Android's Sandbox

Το λειτουργικό σύστημα Android βασισμένο στο μοντέλο που κληρονομεί από τα Linux συστήματα φέρνει μαζί του μια άκρως μελετημένη Unix-like διαδικασία απομόνωσης διεργασιών καθώς και την αρχή των διαφόρων προνομιών, που διαχωρίζεται στους λιγότερο προνομιούχους χρήστες και στους περισσότερο προνομιούχους χρήστες (Drake,J : 2014). Συγκεκριμένα, η ιδέα ότι μια διεργασία τρέχει παράλληλα αλλά ξεχωριστά από κάθε άλλη διεργασία χωρίς να παραβιάζει η μία την άλλη, χρησιμοποιώντας τον χώρο μνήμης της άλλης διεργασίας κ.ο.κ. Ως εκ τούτου, ένα μεγάλο μέρος του Sandbox του λειτουργικού συστήματος Android στηρίζεται σε μερικές βασικές έννοιες, όπως: α) την πρότυπη διαδικασία απομόνωσης του Linux, β) μοναδικές ταυτότητες χρήστη που προσδίδονται για τις περισσότερες εφαρμογές (UIDs), γ) περιορισμένα δικαιώματα του αρχείου συστήματος (file system permissions). Το λειτουργικό σύστημα Android βέβαια, μοιράζεται και τα αναγνωριστικά UID/group ID (GID) που κληρονομεί από τα Linux αλλά δεν εμπεριέχουν τα παραδοσιακά *passwd* και *group* αρχεία ως πηγή διαπιστευτηρίων των χρηστών και της ομάδας (Drake,J : 2014). Αντ'αυτού, το λειτουργικό σύστημα Android ορίζει ένα χάρτη ονομάτων σε μοναδικούς χρήστες, γνωστά και ως Android IDs (AIDs). Η αρχική χαρτογράφηση των AIDs, περιέχει καλά κλεισμένες στατικές καταχωρήσεις, για προνομιούχους και κρίσιμες για το σύστημα χρήστες, όπως το system User/Group. Από την έκδοση 4.1 του λειτουργικού συστήματος Android, η Google πρόσθεσε μια μεγάλη γκάμα επιπρόσθετων AIDs, για πολλούς διαφορετικούς χρήστες καθώς και απομονωμένες διεργασίες χρηστών για καλύτερη απομόνωση του Chrome. Παρακάτω παραθέτονται μερικές από τις μοναδικές ταυτότητες που προσδίδει το λειτουργικό σύστημα σε διάφορες διεργασίες. Περισσότερα στοιχεία για τις διακριτές ταυτότητες που προσδίδει το λειτουργικό σύστημα Android μπορούν να βρεθούν με το path:/system/core/include/private/android_filesystem_config.h στα αρχεία του λειτουργικού συστήματος από το AOSP.

```
#define AID_ROOT             0  /* traditional unix root user */

#define AID_SYSTEM        1000  /* system server */

#define AID_RADIO         1001  /* telephony subsystem, RIL */
#define AID_BLUETOOTH     1002  /* bluetooth subsystem */
...
#define AID_SHELL         2000  /* adb and debug shell user */
#define AID_CACHE         2001  /* cache access */
#define AID_DIAG          2002  /* access to diagnostic resources */

/* The 3000 series are intended for use as supplemental group id's only.
 * They indicate special Android capabilities
that the kernel is aware of. */
#define AID_NET_BT_ADMIN  3001  /* bluetooth: create any socket */
#define AID_NET_BT        3002  /* bluetooth: create sco,
                                   rfcomm or l2cap sockets */
#define AID_INET          3003  /* can create AF_INET and
                                   AF_INET6 sockets */
#define AID_NET_RAW       3004  /* can create raw INET sockets */
...
#define AID_APP          10000  /* first app user */

#define AID_ISOLATED_START 99000 /* start of uids for fully
                                    isolated sandboxed processes */
#define AID_ISOLATED_END   99999 /* end of uids for fully
                                    isolated sandboxed processes */
#define AID_USER        100000 /* offset for uid ranges for each user */
```

Εικόνα 4: Directory εγγραφών των χαρτογραφημένων χρηστών AID(Android Hacker's Handbook)

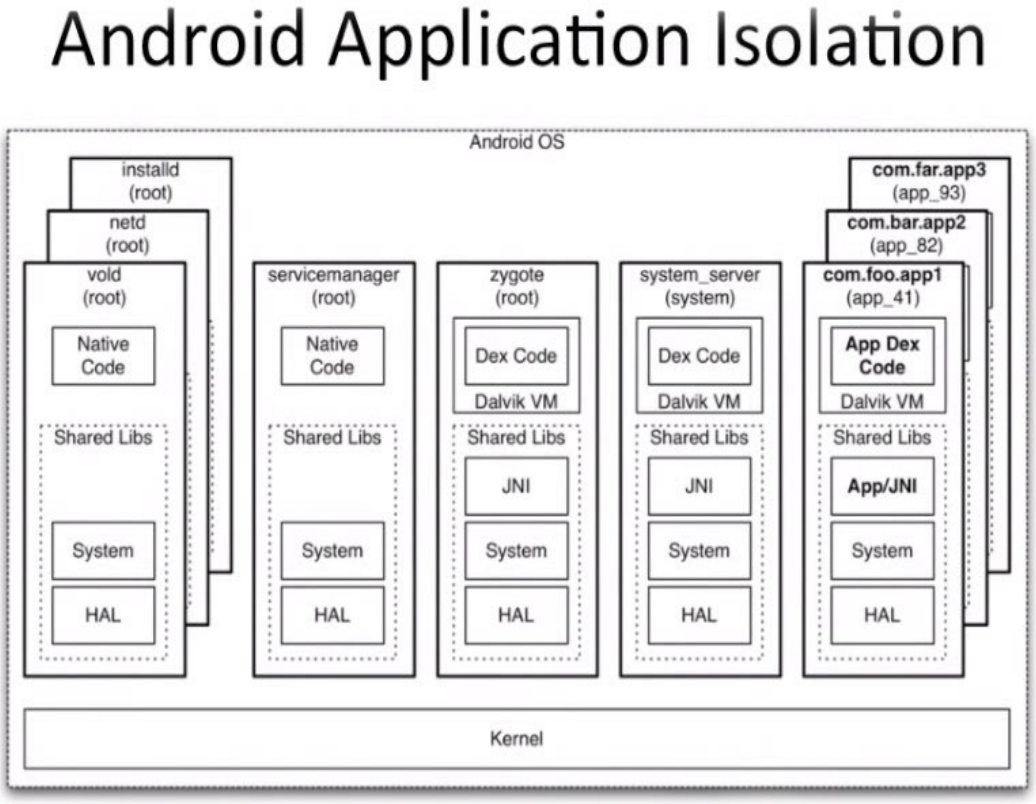
Επιπρόσθετα στα AIDs, το λειτουργικό σύστημα Android, ομαδοποιεί κάποιες διεργασίες προσδίδοντας σε αυτές μια ταυτότητα ομάδας (group ID) επιτρέποντας στις διεργασίες που ανήκουν στην ίδια ομάδα διεργασιών να αξιοποιούν πόρους, άλλων διεργασιών που βρίσκονται στην ίδια ομάδα. Για παράδειγμα, αν μια εφαρμογή ανήκει στην ομάδα sdcard_rw , τότε του επιτρέπεται στην διεργασία να εγγράφει και να διαβάζει στον κατάλογο /sdcard , ενώ όποια ομάδα δεν ανήκει στο GID αυτό δε μπορεί να αξιοποιήσει τους πόρους του καταλόγου αφού περιορίζεται από το Sandboxing του Android (Drake,J : 2014).Τέλος, όταν μια εφαρμογή εκτελείται, η ταυτότητα χρήσης και η ταυτότητα ομάδας στην οποία ανήκει εγγράφονται στις νέες διεργασίες που εκτελούνται στο λειτουργικό σύστημα. Εκτελώντας μια εφαρμογή κάτω από την μοναδική ταυτότητα χρήστη καθώς και την ταυτότητα ομάδας που ανήκει, δίνει την δυνατότητα στο λειτουργικό σύστημα να ενδυναμώσει τους περιορισμούς που έχει σε χαμηλού επιπέδου διεργασίες όπως το Kernel, ενώ παράλληλα επιτρέπει την επικοινωνία μεταξύ των εφαρμογών του συστήματος. Ουσιαστικά, αυτή είναι και η ουσία του Sandbox του λειτουργικού συστήματος του Android. Παρακάτω παρατίθεται ένα στιγμιότυπο από τις εισόδους

δικαιωμάτων που δέχεται το λειτουργικό σύστημα σε μια κινητή συσκευή HTC ONE V . Στα αριστερά αναγράφονται οι διακριτές ταυτότητες των εφαρμογών που βρίσκονται εγκατεστημένες στην συσκευή , ενώ δίπλα ακριβών αναγράφεται ο κωδικός κάθε εφαρμογής που εκτελείται με τη γνωστή από τα Linux εντολή start 4089 (για την πρώτη εφαρμογή κ.ο.κ).

```
app_16    4089  1451  304080 31724 ... S com.htc.bgp
app_35    4119  1451  309712 30164 ... S com.google.android.calendar
app_155   4145  1451  318276 39096 ... S com.google.android.apps.plus
app_24    4159  1451  307736 32920 ... S android.process.media
app_151   4247  1451  303172 28032 ... S com.htc.lockscreen
app_49    4260  1451  303696 28132 ... S com.htc.weather.bg
app_13    4277  1451  453248 68260 ... S com.android.browser
```

Εικόνα 5: Στιγμιότυπο εισόδων δικαιωμάτων του λειτουργικού συστήματος στη συσκευή HTC ONE(Android Hacker's Handbook)

Στο παρακάτω σχήμα παρουσιάζεται η διαδικασία απομόνωσης των διαφόρων διεργασιών.



Σχήμα 7: Διαδικασία απομόνωσης διαφορετικών διεργασιών (com.far.app(i[25]))

[25] (i): όπου i οι διαφορετικές εφαρμογές σε αύξοντα αριθμό.

5.3 Σύστημα δικαιωμάτων του λειτουργικού συστήματος Android

Το σύστημα δικαιωμάτων του λειτουργικού συστήματος Android, είναι πολύπλευρο. Υπάρχουν δικαιώματα για εφαρμογές που αξιοποιούν οι προγραμματιστές για να φτάσουν στα κατώτερα στρώματα του λειτουργικού συστήματος, τα λεγόμενα API, επίσης υπάρχουν δικαιώματα για τα αρχεία συστήματος, καθώς και δικαιώματα IPC δηλαδή δικαιώματα για επικοινωνία μεταξύ των διαφόρων εφαρμογών του συστήματος (Felt, A : 2011). Τα δικαιώματα αυτά όπως αναφέρθηκαν, διαχωρίζονται σε υψηλού επιπέδου και είναι χαρτογραφημένα στο σύστημα με σκοπό να μπορούν να διενεργούν στα χαμηλότερα στρώματα που δεν έχουν απευθείας πρόσβαση οι εφαρμογές, όπως την εκκίνηση του Bluetooth ή την δόνηση του κινητού. Για να μπορεί το ίδιο το λειτουργικό σύστημα να καταλαβαίνει ποια εφαρμογή έχει υψηλού επιπέδου δικαιώματα και ποια είναι αυτά τα δικαιώματα, η Google δίνει την δυνατότητα στον προγραμματιστή να αναγράφει ποια δικαιώματα ζητάει από τον χρήστη να δώσει έγκριση για να γνωρίζει και το λειτουργικό σύστημα ότι επιτρέπεται να χρησιμοποιεί τους συγκεκριμένους πόρους. Τα δικαιώματα αυτά αναγράφονται σε ένα αρχείο .xml που ονομάζεται το Μανιφέστο του λειτουργικού συστήματος Android (AndroidManifest.xml) και εγκαθίστανται μαζί με την εφαρμογή στην διαδρομή /data/system/packages.xml. Παρακάτω παρουσιάζεται ένα στιγμιότυπο που δείχνει τις καταχωρήσεις στο αρχείο package.xml που εμπεριέχει τόσο την ταυτότητα του χρήστη (User ID) καθώς και τα δικαιώματα που ζητάει η εφαρμογή Chrome .

```
<package name="com.android.chrome"
codePath="/data/app/com.android.chrome-1.apk"
nativeLibraryPath="/data/data/com.android.chrome/lib"
flags="0" ft="1422a161aa8" it="1422a163b1a"
ut="1422a163b1a" version="1599092" userId="10082"
installer="com.android.vending">
<sigs count="1">
<cert index="0" />
</sigs>
<perms>
<item name="com.android.launcher.permission.INSTALL_SHORTCUT" />
<item name="android.permission.NFC" />
...
<item name="android.permission.WRITE_EXTERNAL_STORAGE" />
<item name="android.permission.ACCESS_COARSE_LOCATION" />
...
<item name="android.permission.CAMERA" />
<item name="android.permission.INTERNET" />
...
</perms>
</package>
```

Εικόνα 6: Στιγμιότυπο καταχωρήσεων στο αρχείο package.xml(Android Hacker's Handbook)

5.3.1 Android's Cryptography

Το λειτουργικό σύστημα Android χρησιμοποιεί την λεγόμενη κρυπτογράφηση δημοσίου κλειδιού (Public key cryptography) ή αλλιώς την κρυπτογράφηση ασύμμετρου κλειδιού (Asymmetric Cryptography) για πολλούς σκοπούς που σχετίζονται με τις εφαρμογές. Η βασική ιδέα είναι ότι ο αποστολέας και ο παραλήπτης δε μοιράζονται ένα κοινό μυστικό κλειδί όπως στην περίπτωση συμμετρικού κλειδιού, αλλά διαθέτουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες. Συγκεκριμένα κάθε χρήστης διαθέτει δύο κλειδιά κρυπτογράφησης: το ένα ονομάζεται ιδιωτικό κλειδί (private key) και το άλλο δημόσιο κλειδί (public key). Τα δύο αυτά κλειδιά (ιδιωτικό και δημόσιο) έχουν μαθηματική σχέση μεταξύ τους (Drake,J : 2014). Εάν το ένα χρησιμοποιηθεί για την κρυπτογράφηση κάποιου μηνύματος, τότε το άλλο χρησιμοποιείται για την αποκρυπτογράφηση αυτού. Η επιτυχία αυτού του είδους κρυπτογραφικών αλγορίθμων βασίζεται στο γεγονός ότι η γνώση του δημόσιου κλειδιού κρυπτογράφησης δεν επιτρέπει με κανέναν τρόπο τον υπολογισμό του ιδιωτικού κλειδιού κρυπτογράφησης. Αρχικά, το λειτουργικό σύστημα Android, χρησιμοποιεί ένα ειδικό κλειδί που το ονομάζει κλειδί πλατφόρμα για να "υπογράψει" προ-εγκατεστημένα πακέτα εφαρμογών. Εφαρμογές που έχουν το συγκεκριμένο κλειδί μπορούν να έχουν δικαιώματα χρήστη του συστήματος (system-user, privilleges). Εν συνεχεία, οι εφαρμογές τρίτων όπως ονομάζονται έχουν κλειδιά με την ψηφιακή υπογραφή μεμονωμένων προγραμματιστών. Και στις δύο περιπτώσεις το λειτουργικό σύστημα έχει αυτού του είδους τα κλειδιά ώστε να αποτρέπει μη εξουσιοδοτημένες αναβαθμίσεις εφαρμογών.

5.3.2 Δικαιώματα API

Τα δικαιώματα API, συμπεριλαμβάνουν τα δικαιώματα εκείνα που χρησιμοποιούνται για τον έλεγχο προσβασιμότητας σε υψηλού επιπέδου διεργασίες (high-level functionality) που βρίσκονται εντός του Android πλαισίου (framework) καθώς και σε πλαίσια τρίτων κατασκευαστών (Drake,J : 2014). Ένα απλό παράδειγμα API δικαιωμάτων είναι το READ_PHONE_STATE, το οποίο το λειτουργικό δέχεται ως δικαίωμα για ανάγνωση-πρόσβαση στην κατάσταση του τηλεφώνου. Μια εφαρμογή η οποία αιτείται για αυτό το δικαίωμα και του οποίου το αίτημα γίνεται μεταγενέστερα αποδεκτό, θα είναι σε θέση να καλεί μια ποικιλία μεθόδων που σχετίζονται με ερωτήματα πληροφοριών του τηλεφώνου (Felt, A : 2011). Μέθοδοι οι οποίες σχετίζονται με την ίδια κλάση που στο συγκεκριμένο παράδειγμα που θίχτηκε είναι η κλάση TelephonyManager. Οι μέθοδοι

αυτοί μπορεί να είναι η getDeviceSoftwareVersion που επιτρέφει στον χρήστη την έκδοση του λειτουργικού συστήματος της συσκευής ,η getDeviceID που επιστρέφει την μοναδική ταυτότητα της συσκευής που σχετίζεται με τον χρήστη που της ανήκει κ.ο.κ.
Όπως αναφέρθηκε και νωρίτερα, κάποια από τα δικαιώματα API αντιστοιχούν σε δικαιώματα μηχανισμών επιβολής διεργασιών σε επίπεδο kernel που θεωρούνται low level permissions. Για παράδειγμα, αν μια εφαρμογή αιτηθεί το δικαίωμα για πρόσβαση στο Διαδίκτυο δηλαδή αιτηθεί το δικαίωμα INTERNET και γίνει αποδεκτό από το λειτουργικό, τότε αυτό σημαίνει ότι το UID της εφαρμογής θα εγγραφεί ως μέλος της ομάδας inet (GID 3003). Έχοντας γίνει πλέον η εφαρμογή μέλος της ομάδας inet, τότε παρέχεται στον χρήστη η δυνατότητα να ανοίξει και τις λειτουργίες με τα δικαιώματα AF_INET καθώς και τα sockets AF_INET6, οι οποίες είναι απαραίτητες για την να έχει η εφαρμογή στην πορεία πρόσβαση σε υψηλού επιπέδου API διεργασίες, όπως το να μπορεί να δημιουργήσει ένα αντικείμενο από την κλάση HttpURLConnection που χρειάζεται για να έχει πρόσβαση στο Διαδίκτυο μέσω κάποιου περιηγητή.

5.3.3. Δικαιώματα αρχείων συστήματος (File System Permissions)

Κάθε εφαρμογή που εκτελείται στο λειτουργικό σύστημα Android όπως ειπώθηκε και παραπάνω περιορίζεται από τον μηχανισμό ελέγχου εφαρμογών Sandbox το οποίο υποστηρίζεται στενά από τα δικαιώματα αρχείων συστήματος του UNIX. Οι εφαρμογές έχοντας μοναδική ταυτότητα χρήστη καθώς και μοναδική ταυτότητα συμμετοχής σε μια ομάδα διεργασιών, by default, έχουν πρόσβαση μόνο για τις αντίστοιχες διαδρομές αποθήκευσης δεδομένων τους στα αρχεία του συστήματος. Αν μια εφαρμογή δημιουργήσει φακέλους τότε και οι φάκελοι αυτοί θα έχουν τα κατάλληλα δικαιώματα. Παρακάτω παραθέτετε ένας κατάλογος δεδομένων μιας εφαρμογής που υποδεικνύει την ιδιοκτησία των φακέλων από την εφαρμογή twitter καθώς και τα δικαιώματα της συγκεκριμένης εφαρμογής, των υποκαταλόγων της, καθώς και των φακέλων που έχουν οριστεί μόνο για την ταυτότητα UID και των μελών GID στις οποίες ανήκει.

```
root@android:/data/data/com.twitter.android # ls -lR

.:
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 cache
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 databases
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 files
lrwxrwxrwx install  install           2013-10-22 18:16 lib ->
/data/app-lib/com.twitter.android-1
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 shared_prefs

./cache:
drwx------ u0_a55   u0_a55            2013-10-17 00:07
com.android.renderscript.cache

./cache/com.android.renderscript.cache:

./databases:
-rw-rw---- u0_a55   u0_a55     184320 2013-10-17 06:47 0-3.db
-rw------- u0_a55   u0_a55       8720 2013-10-17 06:47 0-3.db-journal
-rw-rw---- u0_a55   u0_a55      61440 2013-10-22 18:17 global.db
-rw------- u0_a55   u0_a55      16928 2013-10-22 18:17 global.db-journal

./files:
drwx------ u0_a55   u0_a55            2013-10-22 18:18
com.crashlytics.sdk.android

./files/com.crashlytics.sdk.android:
-rw------- u0_a55   u0_a55         80 2013-10-22 18:18
5266C1300180-0001-0334-EDCC05CFF3D7BeginSession.cls

./shared_prefs:
-rw-rw---- u0_a55   u0_a55        155 2013-10-17 00:07 com.crashlytics.prefs.
xml
-rw-rw---- u0_a55   u0_a55        143 2013-10-17 00:07
com.twitter.android_preferences.xml
```

Εικόνα 7: Κατάλογος δεδομένων της εφαρμογής twitter και των δικαιωμάτων στους φακέλους της.

5.3.4 Δικαιώματα IPC

Ως δικαιώματα IPC (Inter process communication), ονομάζονται εκείνα τα δικαιώματα που σχετίζονται άμεσα με την επικοινωνία μεταξύ των εφαρμογών, αν και υπάρχει μια επικάλυψη με τα δικαιώματα API (Drake,J : 2014). Η δήλωση και επιβολή αυτών των αδειών μπορεί να ανακύπτουν σε διάφορα επίπεδα, συμπεριλαμβανομένης της εκτέλεσης της εφαρμογής, λειτουργίες των βιβλιοθηκών του συστήματος, ή απ'ευθείας επικοινωνία των διεργασιών μέσα στην ίδια την εφαρμογή. Συγκεκριμένα, τα δικαιώματα αυτά ισχύουν για τα μείζον συστατικά των εφαρμογών Android που είναι χτισμένα πάνω στον μηχανισμό του λειτουργικού συστήματος Android, το λεγόμενο Binder.

5.4.Εφαρμογές Android

Για να γίνει κατανοητή η διαδικασία αξιολόγησης, επίθεσης και προστασίας της ασφάλειας των εφαρμογών Android, πρέπει πρώτα να γίνει κατανοητό το πώς δημιουργούνται αυτές οι εφαρμογές. Αυτή η ενότητα ερευνά τα κομμάτια ασφαλείας καθώς και σχετικές εφαρμογές του Android, όπως η εφαρμογή runtime και την υποστήριξη της από του μηχανισμούς επικοινωνίας IPC. Οι εφαρμογές χωρίζονται σε δύο κατηγορίες. Στις προεγκατεστημένες εφαρμογές και στις εφαρμογές που εγκαθιστά ο χρήστης στην συσκευή του. Στις προεγκατεστημένες εφαρμογές υπάγονται οι εφαρμογές της Google, οι εφαρμογές των διαφόρων εταιριών κατασκευής κινητών όπως το Touchwiz της Samsung με μια γκάμα εφαρμογών της ίδιας της εταιρίας, εφαρμογές από τους διαφόρους παρόχους, όπως για παράδειγμα η εφαρμογή της Cosmote σε συσκευές που ο ίδιος ο πάροχος πουλάει στους καταναλωτές καθώς και εφαρμογές όπως το ημερολόγιο, οι επαφές, τα μηνύματα κ.ο.κ. Μερικές από αυτές τις εφαρμογές μπορεί να έχουν αυξημένα δικαιώματα ή δυνατότητες και ως εκ τούτου χρήζουν ιδιαιτέρου ενδιαφέροντος. Τα πακέτα αυτών των εφαρμογών βρίσκονται στους καταλόγους με path /system/app. Από την άλλη, υπάρχουν εφαρμογές που εγκαθιστά ο ίδιος ο χρήστης στην συσκευή, είτε από κάποιο app store όπως αυτό της Google που είναι προεγκατεστημένο στην συσκευή ή χειροκίνητα, κατεβάζοντας τα αρχεία της εφαρμογής (.obb) καθώς και το αρχείο εγκατάστασης (.apk) από το Διαδίκτυο, που γίνεται με δύο τρόπους. Είτε μέσω pm install δηλαδή μέσω της ίδιας της συσκευής είτε μέσω κάποιου προγράμματος στον υπολογιστή, adb install. Αυτές οι εφαρμογές καθώς και οι αναβαθμίσεις για τις προεγκατεστημένες εφαρμογές βρίσκονται στους καταλόγους με path: /data/app.

5.5. Android Framework

Η «κόλλα» που ενώνει τις εφαρμογές με τους μηχανισμούς εκτέλεσης τους, ονομάζεται Android Framework. Τα πλαίσια αυτά παρέχουν κομμάτια-πακέτα καθώς και τις κλάσεις του λειτουργικού συστήματος Android στους προγραμματιστές ώστε να εκτελούν κάποιες διεργασίες. Μια τέτοια διεργασία μπορεί να περιλαμβάνει, την είσοδο σε κοινές αποθήκες δεδομένων του συστήματος, μεταφορά μηνυμάτων μεταξύ των εφαρμογών, καθώς και την διαχείριση στοιχείων σε επίπεδο διεπιφάνειας χρήστη (UI). Τα κοινά πλαίσια πακέτων που βρίσκονται μέσα στο λειτουργικό σύστημα, είναι αυτά που έχουν το πρόθεμα android.* όπως το android.content ή το android.telephony. Το λειτουργικό σύστημα παρέχει επίσης πολλές κλάσεις από την γλώσσα προγραμματισμού java, καθώς και πακέτα τρίτων κατασκευαστών, όπως βιβλιοθήκες

Apache, καθώς και το SAX XML parser (Drake,J : 2014). Ακόμη, συμπεριλαμβάνουν τις υπηρεσίες που χρησιμοποιούνται για τη διαχείριση πολλών από των διεργασιών που παρέχονται μέσα στην κλάση. Τα αυτοαποκαλούμενα στελέχη ελέγχου ξεκινάν με την κλάση system_server (θα αναλυθεί εκτενέστερα, στη συνέχεια, στην ενότητα Zygote) μετά από την εκκίνηση του συστήματος. Παρακάτω παραθέτονται κάποια από αυτά τα στελέχη καθώς και ο ρόλος τους στα πλαίσια του συστήματος.

FRAMEWORK SERVICE	DESCRIPTION
Activity Manager	Manages Intent resolution/destinations, app/activity launch, and so on
View System	Manages views (UI compositions that a user sees) in activities
Package Manager	Manages information and tasks about packages currently and previously queued to be installed on the system
Telephony Manager	Manages information and tasks related to telephony services, radio state(s), and network and subscriber information
Resource Manager	Provides access to non-code app resources such as graphics, UI layouts, string data, and so on
Location Manager	Provides an interface for setting and retrieving (GPS, cell, WiFi) location information, such as location fix/coordinates
Notification Manager	Manages various event notifications, such as playing sounds, vibrating, flashing LEDs, and displaying icons in the status bar

Πίνακας 10: Διεργασίες του πλαισίου του λογισμικού Android(Android Hacker's Handbook)

5.6. Zygote

Μια από τις πρώτες διεργασίες που εκτελούνται όταν μια συσκευή Android εκκινείτε είναι η διεργασία Zygote. Η διεργασία αυτή είναι υπεύθυνη για να εκκινήσει επιπρόσθετες διεργασίες καθώς και να φορτώσει τις απαραίτητες βιβλιοθήκες που χρησιμοποιούνται από τα πλαίσια Android που αναφέρθηκαν παραπάνω. Μετά την ολοκλήρωση των παραπάνω η διεργασία Zygote λειτουργεί ως φορτωτής αρχείων για κάθε Dalvik διεργασία κάνοντας αντίγραφα του εαυτού του. Αυτός ο μηχανισμός αντιγραφής του εαυτού του επιτρέπει στο σύστημα να εξυπηρετεί όλες τις διεργασίες ταυτόχρονα καθώς και τις εφαρμογές γρηγορότερα, αφού δε θα χρειάζεται να επαναλαμβάνετε η ακριβές διαδικασία φόρτωσης. Σε δεύτερη φάση, ο μηχανισμός της διεργασίας Zygote ενεργοποιεί την διεργασία system_server, η οποία με την σειρά της ενεργοποιεί όλες τις βασικές υπηρεσίες του λειτουργικού οι οποίες λειτουργούν με αυξημένα προνόμια κάτω από την κλάση system. Τέλος, η κλάση system_server ενεργοποιεί όλες τις διεργασίες πλαισίου που παρουσιάστηκαν στον πίνακα 23. Για να γίνει κατανοητό το πόσο σημαντική είναι η διεργασία της κλάσης system_server, αν η

διεργασία αυτή σταματήσει για κάποιο λόγο τότε όλη η συσκευή αυτόματα επανεκινείται. Η διεργασία Zygote επίσης παρέχει τις απαραίτητες βιβλιοθήκες καθώς και επικοινωνία μεταξύ των εφαρμογών μέσω RPC και IPC.

5.7. Kernel

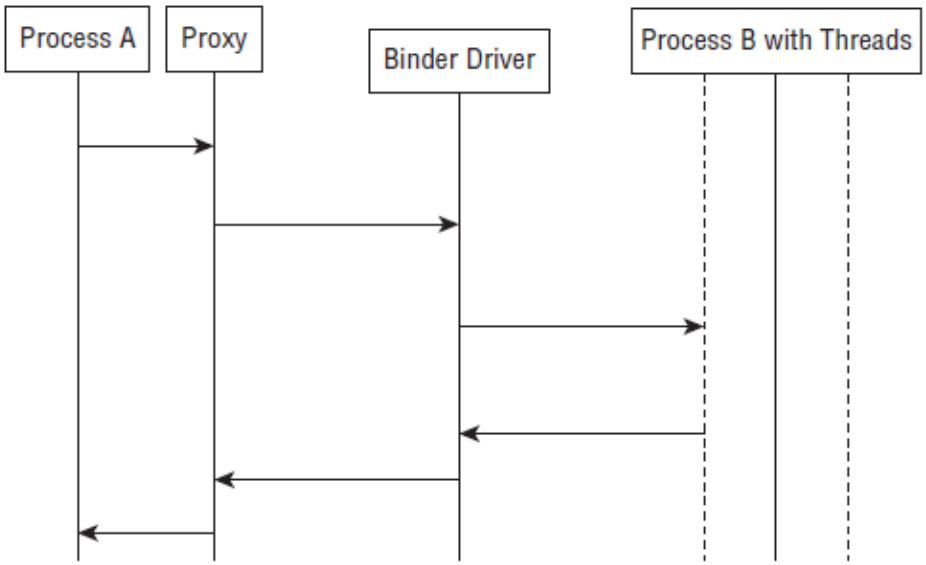
Παρόλο που το λειτουργικό σύστημα είναι βασισμένο στο kernel πυρήνα του Linux, η Google παρουσίασε ένα τροποποιημένο δέντρο υπηρεσιών του kernel. Λέγεται ότι πάνω από 250 διαφορετικές τροποποιήσεις έχει υποστεί το kernel του Linux, που περιλαμβάνουν τροποποιήσεις στο σύστημα διαχείρισης αρχείων, στην επικοινωνία της συσκευή με το Διαδίκτυο, στην διαχείριση μνήμης κ.ο.κ.. Οι αλλαγές αυτές φαίνονται παρακάτω στον πίνακα 24 .

KERNEL CHANGE	DESCRIPTION
Binder	IPC mechanism with additional features such as security validation of callers/callees; used by numerous system and framework services
ashmem	Anonymous Shared Memory; file-based shared memory allocator; uses Binder IPC to allow processes to identify memory region file descriptors
pmem	Process Memory Allocator; used for managing large, contiguous regions of shared memory
logger	System-wide logging facility
RAM_CONSOLE	Stores kernel log messages in RAM for viewing after a kernel panic
"oom" modifications	"Out of memory"-killer kills processes as memory runs low; in Android fork, OOM kills processes sooner than vanilla kernel, as memory is being depleted
wakelocks	Power management feature to keep a device from entering low-power state, and staying responsive
Alarm Timers	Kernel interface for `AlarmManager`, to instruct kernel to schedule "waking up"
Paranoid Networking	Restricts certain networking operations and features to specific group IDs
timed output / gpio	Allows user-space programs to change and restore GPIO registers after a period of time
yaffs2	Support for the yaffs2 flash file system

Πίνακας 11: Αλλαγές στο Linux kernel στο λογισμικό σύστημα Android(Android Hacker's Handbook)

Ίσως η πιο σημαντική από τις αλλαγές που έγιναν στο kernel και αξίζει να αναφερθεί είναι το binder. Το οποίο είναι ο μηχανισμός ενδοεπικοινωνίας IPC που αναφέθηκε και παραπάνω. Η διεργασία αυτή ως μηχανισμός βασίζεται στην αρχιτεκτονική του μοντέλου client-server. Παρακάτω παρουσιάζεται ένα σχήμα που αναδεικνύει την χρήση

του Binder για την ενδοεπικοινωνία μεταξύ δύο διαφορετικών διεργασιών στο ίδιο το σύστημα.



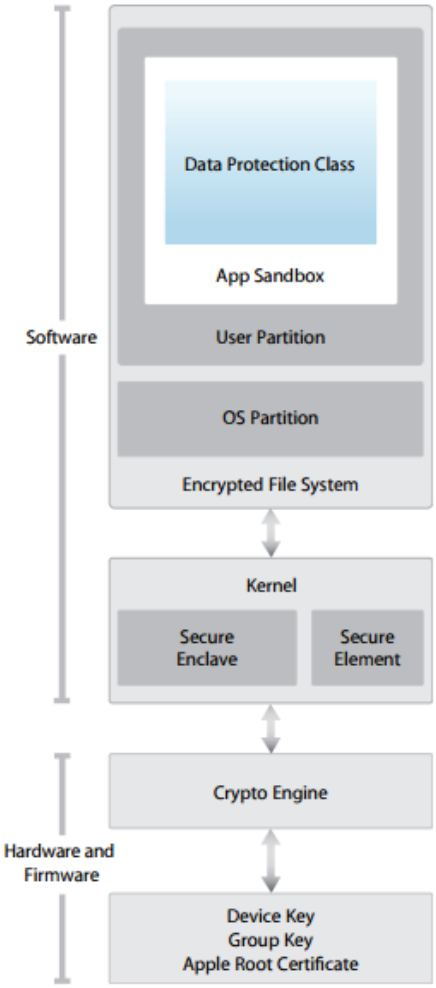
Σχήμα 8: *Λειτουργία ενδοεπικοινωνίας δια μέσου του Binder μεταξύ δύο διεργασιών.*

6. Ο σχεδιασμός και η αρχιτεκτονική του μοντέλου ασφάλειας του iOS

Ο όρος ασφάλεια κινητών συσκευών όπως είναι γνωστός σήμερα στις αγορές δεν συγχρονίζεται με τις πραγματικές ανάγκες της εποχής περί ασφάλειας και ιδιωτικότητας (Seriot,N: 2010). Για πολλούς, ο όρος ασφάλεια δεν σχετίζεται με την πολυπλοκότητα των επιθέσεων αλλά με την μείωση των επιφανειών επίθεσης[26] κυρίως με την χρήση μιας μόνο κουλτούρας ασφαλείας με σκοπό την προστασία ολόκληρου του λογισμικού. Στην περίπτωση του iOS, αυτή η κουλτούρα απαρτίζεται από ένα σετ κλάσεων προστασίας δημιουργημένες από τον κατασκευαστή, την Apple, με σκοπό την παροχή συγκεκριμένων μεθόδων κρυπτογράφησης κωδικών, προστασίας της ιδιωτικότητας των δεδομένων του χρήστη, ασφάλειας των δεδομένων, κ.ο.κ (Dai Zovi,D: 2011). Το μοντέλο αυτό που ακολουθεί η Apple παρόλο που έχει προκαλέσει μια σειρά από αλληλένδετες εξελίξεις στον χώρο της ασφάλειας, το γενικότερο αποτέλεσμα της εξάρτησης ολόκληρης της ασφάλειας του λογισμικού στο μοντέλο αυτό, έχει δημιουργήσει αντίθετα αποτελέσματα στην ασφάλεια των εφαρμογών του ίδιου του λογισμικού (Egele,M: 2011). Οι εφαρμογές έχουν γίνει λιγότερο πολύπλοκες με αποτέλεσμα την παραβίαση των συστημάτων ασφαλείας και την δημιουργία ευπάθειας για όλες τις εφαρμογές του λογισμικού σε περίπτωση που βρεθεί ευπάθεια στο μοντέλο ασφαλείας της κλειστής κουλτούρας που ακολουθεί η Apple. Βέβαια όπως προ ειπώθηκε παραπάνω η Apple έχει επιδείξει πολλές καλές ιδέες στους μηχανισμούς ασφαλείας του λογισμικού της για την προστασία των εφαρμογών για iOS συσκευές, αλλά σε κάθε στάδιο αποδυναμώνονται από τις κρίσιμες ευπάθειες. Επειδή οι περισσότεροι κατασκευαστές λογισμικού λειτουργούν μέσα σε αυτή την κουλτούρα, κάθε φορά που η Apple βρίσκεται σε κίνδυνο παραβίασης, το ίδιο βρίσκονται και οι κατασκευαστές των εφαρμογών για το iOS. Όπως συμβαίνει με τις περισσότερες κουλτούρες τέτοιου είδους όπως της Apple, αν ένας μηχανισμός ασφάλειας αποτύχει, τότε θα αποτύχει με εξολοκλύρου (Dai Zovi,D: 2011). Πολλές αδυναμίες ασφαλείας που βασίζονται στις iOS συσκευές έχουν προκύψει τα τελευταία χρόνια, αφήνοντας στο App Store περίπου μισό εκατομμύριο εφαρμογές να εκτίθενται σε μια σειρά από θέματα ευπάθειας ασφαλείας, που κυρίως κληρονομούνται από την επαναχρησιμοποίηση του κώδικα του κατασκευαστή. Αυτό βέβαια δεν αποτελεί νέο πρόβλημα. Από την εισαγωγή των

[26] Επιφάνεια επίθεσης (Attack Surface): Η επιφάνεια επίθεση ενός λογισμικού περιβάλλοντος είναι το άθροισμα των διαφόρων σημείων (οι «φορείς της επίθεσης»), όπου ένας μη εξουσιοδοτημένος χρήστης (ο "εισβολέας") μπορεί να προσπαθήσει να εισάγει δεδομένα ή να εξάγει δεδομένα από ένα περιβάλλον χωρίς εξουσιοδότηση από τον ιδιοκτήτη.

συστημάτων κρυπτογράφησης για επιχειρήσεις και άλλα χαρακτηριστικά ασφαλείας στο iOS, έχουν βρεθεί πολλές ατέλειες που χρησιμοποιούνται για την προστασία των ιδιωτικών δεδομένων, βάζοντας τα δεδομένα χρηστών και των εταιριών, για εκατομμύρια συσκευές σε κίνδυνο. Δυστυχώς, η νομοθεσία περί πνευματικών δικαιωμάτων των Ηνωμένων Πολιτειών κατέστησε δύσκολο να εκθέσει πολλά από αυτά τα κενά ασφαλείας. Η Apple έλαβε μια επιθετική νομική στάση κατά το άνοιγμα ιδιωτικών APIs της συσκευής, ενώ προσπάθησε σε μεγάλο βαθμό να καταστείλει την τρέχουσα ερευνητική κοινότητα, υποστηρίζοντας ότι μέθοδοι όπως το jailbreaking ήταν παράνομες, και αποτελούν παραβίαση των δικαιωμάτων πνευματικής ιδιοκτησίας (Dai Zovi,D: 2011). Το λειτουργικό σύστημα iOS βασίζεται σε δύο επίπεδα. Στο λειτουργικό σύστημα και στο Hardware-Firmware, όπως φαίνεται παρακάτω:



Πίνακας 12: Διάγραμμα αρχιτεκτονικής ασφαλείας του λειτουργικού συστήματος iOS[27]

[27] Source:*https://www.apple.com/business/docs/iOS_Security_Guide.pdf*

Ο διαχωρισμός του λειτουργικού συστήματος σε δύο στρώσεις έγινε με σκοπό την επιτυχία μεγαλύτερων ποσοστών ασφάλειας του λειτουργικού. Στα ανώτερα στρώματα το λειτουργικό διαχωρίζεται σε δυο διακριτά partitions[28], αυτό του χρήστη και αυτό του λειτουργικού συστήματος, τα οποία και τα δύο κρυπτογραφούνται. Στη συνέχεια υπάρχει το kernel βασισμένο στο μοντέλο UNIX που παρέχει του απαραίτητους οδηγούς και τις απαραίτητες άδειες χρήσης για χαμηλού επιπέδου διεργασίες. Έπειτα σε επίπεδο υλικού υπάρχει ο πυρήνας κρυπτογράφησης κάθε συσκευής που θα αναλυθεί παρακάτω η χρήση του, ενώ στο τέλος της αλυσίδας υπάρχει ο μηχανισμός ψηφιακής υπογραφής της Apple καθώς και οι άδειες χρήσης. Όλοι αυτοί οι μηχανισμοί ασφαλείας απαρτίζουν το λειτουργικό σύστημα iOS (Hoog,A: 2011).

6.1. Κατανοώντας τα όρια ασφάλειας και την επιβολή μηχανισμών ελέγχου στο λειτουργικό iOS.

Η Apple στο μοντέλο ασφαλείας της έχει ενσωματώσει τέσσερα στρώματα προστασίας στο λειτουργικό της σύστημα με σκοπό την προστασία τόσο των χρηστών όσο και των δεδομένων τους (Seriot,N: 2010). Αναφορικά τα τέσσερα στρώματα λειτουργούν ανεξάρτητα αλλά αλληλένδετα μεταξύ τους δημιουργώντας το μοντέλο ασφαλείας του iOS. Η ασφάλεια των δεδομένων, δηλαδή τεχνικές με σκοπό την προστασία των δεδομένων που αποθηκεύονται στην συσκευή ακόμη και σε περιπτώσεις κλοπής της συσκευής. Η ασφάλεια δικτύων, δηλαδή εργαλεία που κρυπτογραφούν τα δεδομένα που εξέρχονται από την συσκευή κατά την μετάδοση τους στο Διαδίκτυο και τέλος η ασφάλεια των εφαρμογών, που επιτυγχάνεται μέσα από μηχανισμούς που προστατεύουν το λειτουργικό σύστημα και απομονώνουν τις εφαρμογές την ώρα που εκτελούνται.

6.1.1. Ασφάλεια συσκευής

Οι μηχανισμοί ασφαλείας που χρησιμοποιεί η Apple βοηθούν στο να διασφαλιστεί ότι η συσκευή του χρήστη δεν μπορεί να χρησιμοποιηθεί από μη εξουσιοδοτημένο πρόσωπο. Ο πιο κοινός μηχανισμός ασφαλείας της συσκευής είναι κλείδωμα PIN της συσκευής ή ο κωδικός πρόσβασης. Η Apple επιτρέπει και προτρέπει τις

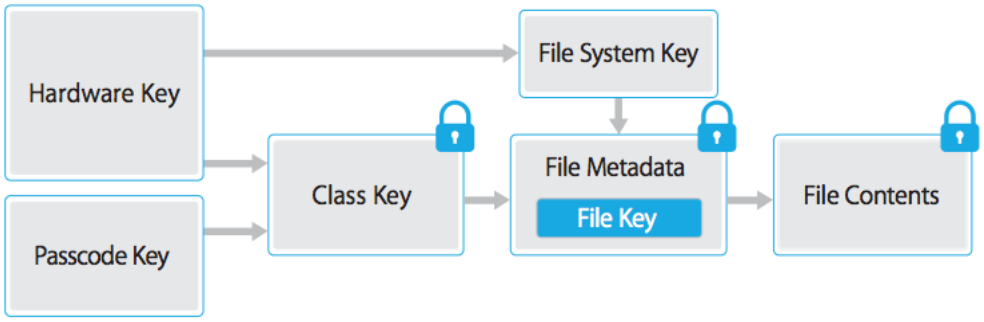
[28] Partitions ή **Διαμέριση δίσκου:** ονομάζουμε τη δημιουργία ξεχωριστών κατατμήσεων (διαμέριση) ενός σκληρού δίσκου χρησιμοποιώντας ένας επεξεργαστή κατατμήσεων. Από την στιγμή που ο δίσκος έχει διαιρεθεί σε κατατμήσεις, κατάλογοι και αρχεία διαφορετικών κατηγοριών μπορούν να αποθηκευθούν σε διαφορετικές κατατμήσεις.

επιχειρήσεις να χρησιμοποιούν τέτοιου είδους μηχανισμούς ασφαλείας ως μέρος της πολιτικής ασφαλείας τους, διαφορετικά δίνει στον χρήστη την δυνατότητα να εισάγει μόνος του τον συγκεκριμένο μηχανισμό ασφαλείας άμα το επιθυμεί. Όσο αφορά τις επιχειρήσεις που αναφέρθηκαν παραπάνω, μπορούν να αναγκάσουν στον χρήστη μιας εταιρικής συσκευής να χρησιμοποιεί έναν κωδικό πρόσβασης με ελάχιστο μήκος, αλφαριθμητική σύνθεση, πολύπλοκους χαρακτήρες, ακόμη και να καθορίζει τα ανώτατα επιτρεπτά χρονικά περιθώρια για την χρήση ενός κωδικού πρόσβασης, επιβάλλοντας πολλές φορές την αλλαγή του μετά από n διάστημα (Dai Zovi,D: 2011).

6.1.2. Ασφάλεια δεδομένων

Η ασφάλεια δεδομένων είναι η κύρια απαίτηση όλων των χρηστών και αποτελεί βασική προϋπόθεση για τη δημιουργία ασφαλών εφαρμογών. Η Apple έχει υιοθετήσει μια σειρά από μηχανισμούς ασφαλείας, με σκοπό την προστασία ευαίσθητων προσωπικών δεδομένων που αποθηκεύονται στην συσκευή ακόμη και όταν η συσκευή έχει κλαπεί. Αυτοί οι μηχανισμοί εμπεριέχουν μηχανισμούς όπως την διαγραφή δεδομένων εξ'απόστασεως και χωρίς φυσική παρουσία, την κρυπτογράφηση των δεδομένων καθώς και την προστασία των δεδομένων. Ο μηχανισμός απομακρυσμένης διαγραφής της Apple επιτρέπει στην συσκευή να διαγράφει όλα τα αρχεία του λειτουργικού συστήματος μόλις η συσκευή θεωρηθεί κλεμμένη, ή όταν έχουν γίνει πάρα πολλές απόπειρες εισαγωγής εσφαλμένου κωδικού. Η συσκευή μπορεί να διαγράψει όλα τα δεδομένα του χρήστη σε μικρό χρονικό διάστημα συνήθως κάτω από τριάντα δευτερόλεπτα (Dai Zovi,D: 2011). Από την άλλη, ο μηχανισμός κρυπτογράφησης εξαναγκάζει όλα τα δεδομένα στο να κρυπτογραφηθούν. Σε συνδυασμό με την κρυπτογράφηση των δεδομένων, τα ίδια τα δεδομένα μπορούν να κρατηθούν ως back up διαμέσου του λογισμικού για υπολογιστές iTunes όπου και εκεί θα κρυπτογραφηθούν για την καλύτερη προστασία των δεδομένων του χρήστη. Κάθε φορά που γίνεται back up στο iTunes, ο κωδικός της συσκευής χρησιμοποιείται για να κρυπτογραφήσει τα δεδομένα. Ανεξάρτητα από τον υπολογιστή που χρησιμοποιούμε για την δημιουργία back up, ό μόνος τρόπος να αποκρυπτογραφηθούν τα δεδομένα που αποθηκεύτηκαν στον υπολογιστή είναι μέσω του κλειδιού που βρίσκεται στην συσκευή και είναι μοναδικός. Ο μηχανισμός προστασία δεδομένων της Apple είναι ο πιο ευρέως γνωστός και αυτός στον οποίο επιτίθενται στις iOS συσκευές. Ο παραπάνω μηχανισμός, χρησιμοποιεί έναν επιταχυντή κρυπτογράφησης σε επίπεδο hardware που χρησιμοποιείται για την κρυπτογράφηση δεδομένων τόσο από την ίδια την Apple όσο και

από του προγραμματιστές 3rd party εφαρμογών για το λειτουργικό iOS. Συνδυάζοντας, το κλειδί κρυπτογράφησης που αποθηκεύεται στην συσκευή μαζί με ένα σετ κωδικών που χρησιμοποιεί ο ίδιος ο χρήστης για να χρησιμοποιήσει την συσκευή διασφαλίζεται η κρυπτογράφηση όλων των αρχείων του συστήματος καθώς και των δεδομένων του χρήστη, αποκρυπτογραφόντας τα αρχεία μόνο και όταν γίνει η αυθεντικοποίηση του χρήστη από την συσκευή καλύπτοντας την προϋπόθεση της εμπιστευτικότητας στα λειτουργικά συστήματα. Η μέθοδος αυτή εξαρτάται βέβαια από την πολυπλοκότητα του κωδικού πρόσβασης που ο ίδιος ο χρήστης χρησιμοποιεί. Αξίζει να σημειωθεί ότι παρόλο που όλα τα αρχεία του συστήματος είναι κρυπτογραφημένα, μόνο κάποια αρχεία χρησιμοποιούν τον μηχανισμό προστασίας δεδομένων της Apple. Για να μπορέσουν οι εφαρμογές 3rd party κατασκευαστών να χρησιμοποιήσουν τον μηχανισμό αυτό θα πρέπει να εισάγουν τον κώδικα που ενεργοποιεί τον μηχανισμό αυτό στις εφαρμογές τους.



Σχήμα 9: Μηχανισμός προστασίας δεδομένων για το iOS
(https://www.apple.com/business/docs/iOS_Security_Guide.pdf)

6.1.3. Ασφάλεια δικτύων

Η ασφάλεια δικτύων έχει υπάρξει ως τεχνική και ως μέθοδος από τότε που υπάρχουν τα δίκτυα και η Apple έχει υιοθετήσει πολλές από τις λύσεις-μεθόδους προστασίας που χρησιμοποιούνται σήμερα με σκοπό την ασφαλή δικτύωση των συσκευών στο λειτουργικό σύστημα iOS. Αυτές περιλαμβάνουν πρωτόκολλα μεταφοράς δεδομένων SSL/TLS, καθώς και μηχανισμούς κρυπτογράφησης και αυθεντικοποίησης WEP/WPA/WPA2 (Miller,C :2012).

6.1.4.Ασφάλεια εφαρμογών

Σε επίπεδο ασφάλειας, εφαρμογές οι οποίες βρίσκονται στο App store της Apple συνήθως εκτελούνται μέσα σε αυτό που αποκαλούμαι Sandbox. Με τον όρο Sandbox όπως και στο λειτουργικό σύστημα της Google που αναφέρθηκε στο παραπάνω κεφάλαιο, ορίζεται ένα περιβάλλον μέσα στο οποίο ο κώδικας μιας εφαρμογής που εκτελείται δε θεωρείται ασφαλής και ως εκ τούτου απομονώνεται από όλες τις υπόλοιπες διεργασίες του συστήματος καθώς και του πόρους του συστήματος που είναι διαθέσιμοι για το λειτουργικό σύστημα. Συγκεκριμένα ο μηχανισμός Sandbox της Apple περιορίζει το μέγεθος μνήμης καθώς και του επεξεργαστικούς κύκλους που μπορεί μια εφαρμογή να χρησιμοποιήσει, ενώ απαγορεύει την προσβασιμότητα σε φακέλους εκτός του φακέλου της ίδιας της εφαρμογής που δημιουργήθηκε κατά αποκλειστικότητα στην εγκατάσταση της. Η Apple παρέχει κλάσεις με σκοπό να χρησιμοποιούν χαμηλού επιπέδου πρόσβαση σε λειτουργίες όπως η κάμερα, το GPS κ.ο.κ. Εφαρμογές οι οποίες ενεργούν μέσα στο Sandbox δε μπορούν να έχουν πρόσβαση σε άλλες εφαρμογές, ούτε και στα δεδομένα του και στους πόρους που αξιοποιούν. Επιπρόσθετα, η Apple χρησιμοποιεί και την μέθοδο του ψηφιακού πιστοποιητικού. Δηλαδή, δεν επιτρέπει σε εφαρμογές που δεν έχουν χορηγηθεί αυτό το πιστοποιητικό από την ίδια την Apple να εκτελεσθούν στο λειτουργικό σύστημα iOS.

6.2. Η ψηφιακή υπογραφή της Apple (Apple's code signing)

Ένας από τους πιο σημαντικούς μηχανισμούς ασφαλείας του λειτουργικού iOS και ίσως ο πιο ισχυρός είναι ο μηχανισμός χρήσης ψηφιακών υπογραφών της Apple. Όλα τα εκτελέσιμα αρχεία καθώς και οι βιβλιοθήκες που υπάρχουν ή πρόκειται να χρησιμοποιηθούν στο λειτουργικό θα πρέπει να έχουν υπογραφεί ψηφιακά από μια αξιόπιστη αρχή όπως η Apple, πριν το kernel επιτρέψει την εκτέλεση τους στο λειτουργικό. Ως εκ τούτου για να μπορέσει οποιαδήποτε εφαρμογή να εκτελεστεί από το kernel στο λειτουργικό iOS θα πρέπει να έχει την υπογραφή από έμπιστες πηγές. Έμπρακτα, αυτό σημαίνει ότι καμία εφαρμογή δε μπορεί να τροποποιηθεί δυναμικά κατά την περίοδο εγκατάστασης και χρήσης ενώ παράλληλα δε μπορεί και να αναβαθμιστεί. Η εφαρμογή της μεθόδου των ψηφιακών υπογραφών καθώς και η απαγόρευση του kernel σε οποιοδήποτε κομμάτι κώδικα που δεν έχει υπογραφεί ψηφιακά λειτουργεί ως κατασταλτικό μέσο, απαγορεύοντας στους χρήστες να κατεβάζουν και να εκτελούν τυχαίες εφαρμογές από το Διαδίκτυο. Όλες οι εφαρμογές πρέπει και ορίζεται από την πολιτική ορθής χρήσης του λειτουργικού της ίδιας της Apple

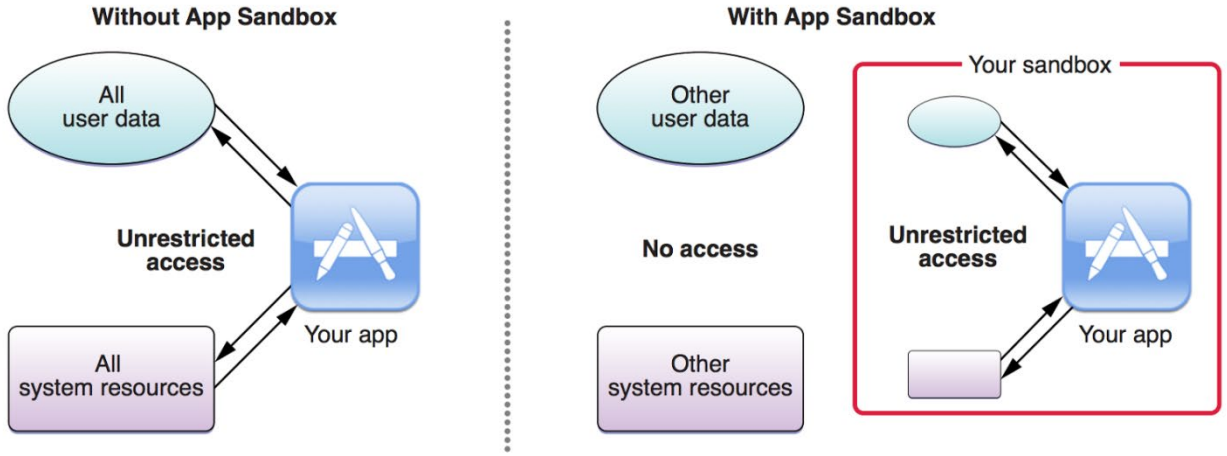
να προέρχονται από το ηλεκτρονικό κατάστημα της Apple app store. Η ίδια η Apple έχει τον απόλυτο έλεγχο της έγκρισης και επιτήρησης οποιασδήποτε εφαρμογής, πριν αυτές ανεβούν στο ηλεκτρονικό της κατάστημα και γίνουν προσβάσιμες στους χρήστες (Dai Zovi,D: 2011). Με τον τρόπο αυτό, η Apple ουσιαστικά διαδραματίζει τον ρόλο του Antivirus για το λειτουργικό της. Αυτού του είδους προστασίας κάνει σχεδόν απίθανη την εμφάνιση οποιουδήποτε Malware. Στην πράξη ελάχιστες περιπτώσεις malware έχουν βρεθεί σε εφαρμογές για το iOS (Egele,M: 2011). Με την χρήση των ψηφιακών υπογραφών η Apple δυσκολεύει το exploitation[29]. Μόλις ένα exploit εκτελέσει κώδικα στην μνήμη της συσκευής, μπορεί να θελήσει να κατεβάσει, εγκαταστήσει και εκτελέσει περαιτέρω κακόβουλες εφαρμογές. Χάρη όμως στον συγκεκριμένο μηχανισμό, το Exploit δε θα μπορέσει να εγκαταστήσει καμία εφαρμογή διότι δε θα είναι ψηφιακά υπογεγραμμένη από την Apple. Ως εκ τούτου, το exploit θα περιοριστεί στη διαδικασία που αρχικά εκμεταλλεύτηκε, εκτός και αν είχε σκοπό την επίθεση άλλων χαρακτηριστικών της συσκευής. Ο μηχανισμός ασφαλείας των ψηφιακών υπογραφών είναι και ο κύριος λόγος που οι χρήστες κάνουν jailbreak. Μόλις επιτευχθεί το jailbreak σε μια συσκευή, αμέσως ο μηχανισμός απενεργοποιείται επιτρέποντας στους χρήστες να εγκαταστήσουν όποια εφαρμογή θέλουν ακόμη και από ανώνυμες πηγές (Miller,C :2012).

6.3. iOS Sandbox

Το τελευταίο τείχος προστασίας του λειτουργικού iOS είναι το Sandbox της. Το Sandbox όπως και στο λειτουργικό Android, επιτρέπει ακόμη και τον πιο λεπτομερές έλεγχο των ενεργειών των διεργασιών που μπορούν να εκτελεστούν από το σύστημα δικαιωμάτων των Unix που θα αναφερθεί παρακάτω. Για παράδειγμα, τόσο η εφαρμογή sms όσο και ο φυλλομετρητής (browser) εκτελούνται κάτω από την άδεια χρήστη mobile, παρόλα αυτά αποδίδουν πολύ διαφορετικές ενέργειες στον τελικό χρήστη. Η εφαρμογή sms προφανώς δε χρειάζεται πρόσβαση στα cookies του φυλλομετρητή, ενώ παράλληλα ο φυλλομετρητής δε χρειάζεται πρόσβαση στα μηνύματα του χρήστη. Συνάμα, οι εφαρμογές 3rd party δεν χρειάζονται πρόσβαση και δε θα έπρεπε να έχουν πρόσβαση σε καμία από τις παραπάνω εφαρμογές και στα αρχεία τους. Το Sandbox λύνει αυτό το

[29] Exploitation: Είναι ένα κομμάτι του λογισμικού, ή ένα κομμάτι των δεδομένων, ή μια ακολουθία εντολών που εκμεταλλεύεται ένα σφάλμα ή ευπάθεια για να προκαλέσει ακούσια ή απρόβλεπτη συμπεριφορά σε λογισμικό ηλεκτρονικών υπολογιστών, υλικού, ή κάτι ηλεκτρονικό. Μια τέτοια συμπεριφορά συχνά περιλαμβάνει την απόκτηση ελέγχου ενός συστήματος, την μη εξουσιοδοτημένη κλιμάκωση προνομιών, ή μια επίθεση DDoS(Denial-of-Service).

πρόβλημα, επιτρέποντας στην Apple να καθορίζει ακριβώς πια δικαιώματα είναι αναγκαία για τις εφαρμογές. Το Sandbox έχει δύο εφαρμογές. Αρχικά, περιορίζει το κακόβουλο λογισμικό που μπορεί να κάνει ζημιά στη συσκευή. Εάν φανταστείτε ότι ένα κομμάτι κακόβουλου λογισμικού καταφέρει να περάσει τους μηχανισμούς ελέγχου της Apple και γίνει διαθέσιμη για κατέβασμα από του χρήστες διαμέσου του App store και τελικά εκτελεστεί από μια συσκευή, τότε και πάλι η εφαρμογή θα περιοριστεί από τους κανόνες του sandbox. Μπορεί το κακόβουλο λογισμικό να καταφέρει να κλέψει όλες τις φωτογραφίες και τις επαφές του χρήστη αλλά δε θα μπορέσει να κάνει τηλεφωνήματα και να στείλει μηνύματα , που άμεσα θα κοστίσει στο χρήστη χρήματα (Miller,C :2012). Το Sandbox επίσης, κάνει την διαδικασία exploitation δυσκολότερη. Εάν ένας εισβολέας βρει μια ευπάθεια στη μειωμένη επιφάνεια επίθεσης που αναφέραμε και παραπάνω, και καταφέρει να πάρει κωδικό εκτέλεσης παρ'όλους τους μηχανισμούς προστασίας του λογισμικού (ASLR-DEP) οι οποίοι θα αναφερθούν παρακάτω, τότε και πάλι θα περιοριστεί στα αρχεία και στις λειτουργίες που το sandbox επιτρέπει (Miller,C :2012). Συμπερασματικά, όλες αυτές οι μέθοδοι ασφαλείας και προστασίας του χρήστη και της ιδιωτικότητας του κάνουν το exploitation καθώς και την εκτέλεση malware σχεδόν απίθανη.



Σχήμα 10: Λειτουργία των εφαρμογών μέσα στα πλαίσια του Sandbox και εκτός[30].

6.4. Σύστημα δικαιωμάτων του λειτουργικού iOS

[30]Source: *https://www.apple.com/business/docs/iOS_Security_Guide.pdf*

Το λειτουργικό σύστημα iOS διαχωρίζει τις διεργασίες χρησιμοποιώντας παραδοσιακούς μηχανισμούς αδειών για τα αρχεία υιοθετημένα από το μοντέλο UNIX που αναφέρθηκαν και στο κεφάλαιο 3. Διαχωρίζει δηλαδή τις άδειες χρήσης για τις διεργασίες ανά χρήστες και ανά ομάδες (UID/GID). Για παράδειγμα, πολλές από τις εφαρμογές που ο χρήστης έχει άμεση πρόσβαση, όπως ο φυλλομετρητής, η εφαρμογή για τα μηνύματα κ.ο.κ., εκτελούνται στο λειτουργικό κάτω από την άδεια χρήστη mobile. Οι σημαντικότερες διεργασίες συστήματος, εκτελούνται με την άδεια προνομιούχου χρήστη root. Ουσιαστικά, αποτελεί άδεια Administrator που δίνει το δικαίωμα αλλαγών σε όλο το φάσμα του συστήματος. Άλλες χαμηλού επιπέδου διεργασίες όπως η διεργασία για την ενεργοποίηση του wifi εκτελούνται με την άδεια προνομίων άλλος χρήστης (other user), μέσω της κλάσης _wireless. Χρησιμοποιώντας αυτό το μοντέλο που υιοθετείται τόσο από το λειτουργικό iOS όσο και από το Android, ο επιτιθέμενος που θα αποκτήσει πλήρη πρόσβαση σε μια εφαρμογή όπως αυτή του safari για παράδειγμα εξαιτίας του ότι έχει η εφαρμογή δικαιώματα mobile δε θα μπορεί να ξεπεράσει τα όρια τον προνομίων του mobile χρήστη και δε θα μπορέσει να ενεργήσει τόσο σε άλλες εφαρμογές όσο και σε διεργασίες χαμηλού επιπέδου που χρειάζονται δικαιώματα root κ.ο.κ (Halbronn,C: 2010).

6.4.1. Κλάσεις προστασίας δεδομένων (Data protection Classes)

Κάθε φορά που δημιουργείται κάποιο αρχείο σε μια iOS συσκευή, αμέσως της αναθέτετε μια κλάση από την εφαρμογή που τη δημιούργησε. Κάθε κλάση χρησιμοποιεί διαφορετική πολιτική η οποία καθορίζει αν τα δεδομένα πρέπει να είναι προσβάσιμα. Οι βασικές κλάσεις και οι πολιτικές τους είναι, η κλάσης πλήρης προστασίας (Complete protection class), η οποία δημιουργεί ένα κλειδί από τον κωδικό του χρήστη της συσκευής και τον μοναδικό κωδικό UID της συσκευής. Αυτή η μέθοδος έχει ως αποτέλεσμα μόλις κλειδώσει η συσκευή (10 δευτερόλεπτα), να κρυπτογραφεί όλα τα αρχεία του φακέλου κάνοντας τα αρχεία μη προσβάσιμα και άχρηστα εκτός και αν ο χρήστης εισάγει τον κωδικό του. Στη συνέχεια υπάρχει η κλάση Protected Unless Open, στην οποία κάποια από τα αρχεία μπορεί να χρειάζονται να εγγράφονται καθώς η συσκευή είναι κλειδωμένη. Στην περίπτωση αυτή εντάσσεται για παράδειγμα η εφαρμογή mail που μπορεί να κατεβάζει κάποιο συνημμένο αρχείο στο προσκήνιο καθώς η συσκευή θα είναι κλειδωμένη. Αυτή η μέθοδος επιτυγχάνετε με την χρήση ελλειπτικής ασύμμετρης

κρυπτογραφίας[31]. Η συνηθισμένη μέθοδος κλειδί για κάθε φάκελο προστατεύεται με την μέθοδο ένα κλειδί για κάθε αρχείο Diffie-Hellman key agreement , που περιγράφεται και στην Nist. Έπειτα υπάρχει η μέθοδο Protected until first user authentication. Η συγκεκριμένη κλάση συμπεριφέρεται το ίδιο με την κλάση πλήρης προστασίας, με την μόνη διαφορά ότι το κλειδί αποκρυπτογράφησης των αρχείων δε διαγράφεται από την μνήμη της συσκευής όταν η συσκευή κλειδωθεί. Σκοπός της συγκεκριμένης κλάσης είναι να προστατέψει την συσκευή από επιθέσεις που εκμεταλλεύονται την επανεκκίνηση της συσκευής. Τελευταία κλάση είναι η No protection, η οποία προστατεύεται μόνο από το UID της συσκευής. Στη συγκεκριμένη κλάση όλα τα κλειδιά αποκρυπτογράφησης υπάρχουν αποθηκευμένα στη συσκευή και η μόνη ουσιαστική ενέργεια που βοηθάει στη χρήση αυτής της κλάσης είναι ότι επωφελείται από την απομακρυσμένη διαγραφή των αρχείων όταν ζητηθεί από τον χρήστη (Dai Zovi,D: 2011).

6.4.2. iOS Cryptography

Η αρχιτεκτονική κρυπτογράφησης που χρησιμοποιεί η Apple ονομάζεται «Apple's common crypto» ουσιαστικά παρέχει κοινά APIs σε όλους του προγραμματιστές που θέλουν να εντάξουν επιπρόσθετα επίπεδα ασφάλειας μέσω της κρυπτογράφησης στις εφαρμογές τους. Η αρχιτεκτονική κρυπτογράφησης της Apple συμπεριλαμβάνει AES, 3DES, και RC4 κρυπτογραφήσεις. Η Apple έχει παντρέψει το framework της με τους μηχανισμούς κρυπτογράφησης σε επίπεδο hardware, παρέχοντας γρήγορα AES κρυπτογράφηση παράλληλα με SHA1 hashing[32], εκ των οποίων και τα δύο χρησιμοποιούνται από την Apple για την υπόγεια προστασία των δεδομένων (Hoog,A: 2011).

6.4.3. Αποτροπή εκτέλεσης δεδομένων-Data Execution Prevention(DEP)

Συνήθως ο μηχανισμός Data Execution Prevention (DEP) λειτουργεί ως μια μέθοδος όπου ο επεξεργαστής μπορεί να ξεχωρίσει ποια κομμάτια μνήμης είναι εκτελέσιμος κώδικας και ποια κομμάτια μνήμης είναι δεδομένα. Ο μηχανισμός DEP δεν επιτρέπει την εκτέλεση δεδομένων παρά μόνο κώδικα. Αυτό είναι σημαντικό κυρίως

[31] Ελλειπτική ασύμμετρη κρυπτογραφία: Είναι μια προσέγγιση κρυπτογραφίας δημοσίου κλειδιού που βασίζεται στην αλγεβρική δομή των ελλειπτικών καμπυλών πάνω σε πεπερασμένα πεδία. Ένα από τα κύρια οφέλη σε σύγκριση με την μη ελλειπτική κρυπτογραφία είναι το ίδιο επίπεδο ασφάλειας που παρέχεται από τα κλειδιά μικρότερου μεγέθους (bit).

[32] SHA-1: είναι μια διεργασία κρυπτογράφησης σχεδιασμένη από τις ΗΠΑ. Το SHA-1 παράγει 160-bit αξίας Hash.

όταν κάποιος προσπαθεί να κάνει exploitation, δηλαδή να ανακαλύψει μια ευπάθεια σε κάποιο συγκεκριμένο σημείο του λειτουργικού συστήματος και το exploit του προσπαθεί να τρέξει payload. Για να γίνει κατανοητή η διαφορά ανάμεσα σε exploit και payload δίνεται το εξής παράδειγμα. Ένας πύραυλος έχει το μεταλλικό περίβλημα, τα καύσιμα για να πετάξει καθώς και οτιδήποτε άλλο είναι χρήσιμο για την εκτόξευση του στον αέρα. Η πυρηνική κεφαλή του βέβαια είναι αυτή που ουσιαστικά κάνει όλη την ζημιά. Χωρίς την κεφαλή ο πύραυλος δε θα κάνει ιδιαίτερη ζημιά στον στόχο του. Δηλαδή, Ο πύραυλος είναι το exploit, το μέσω που κατευθύνει την βόμβα στο σωστό μέρος, ενώ η κεφαλή η ίδια είναι το payload (Dai Zovi,D: 2011). Τα κομμάτια κακόβουλου κώδικα (payloads) δε μπορούν να κάνουν καμία ζημιά χωρίς την πρότερη κατεύθυνση από κάποιο exploit. Το DEP κάνει ένα τέτοιου είδους exploitation σχεδόν απίθανο γιατί ο επεξεργαστής αναγνωρίζει το payload ως δεδομένα και όχι ως εκτελέσιμο κώδικα και έτσι δεν εκτελεί το payload. Ο τρόπος με τον οποίο προσπαθούν οι hackers να προσπελάσουν το DEP είναι συνήθως με ROP (Return-oriented programming) στην οποία ουσιαστικά ο επιτιθέμενος ξαναχρησιμοποιεί κομμάτια κώδικα που ήδη υπάρχουν στη μνήμη και είναι επεξεργάσιμα με ένα τρόπο που δεν είχε οριστεί από τον αρχικό προγραμματιστή της εφαρμογής. Θα μπορούσε να ειπωθεί ότι η ψηφιακή υπογραφή της Apple που αναφέραμε παραπάνω λειτουργεί ως μηχανισμός DEP, όπως αναφέρθηκε και παραπάνω. Στην πραγματικότητα συνηθισμένες επιθέσεις με την χρήση ROP που έχουν σκοπό να δημιουργήσουν ένα μικρό χώρο στη μνήμη που είναι εγγράψιμος και εκτελέσιμος ώστε να εκτελέσουν τα payloads τους δε μπορούν να εκτελεσθούν, καθώς ο μηχανισμός της ψηφιακής υπογραφής επιβλέπει τον κώδικα και αφού δεν έχει ψηφιακή υπογραφή από έμπιστη πηγή ο κώδικας δεν εκτελείται (Dai Zovi,D: 2011).

6.4.4. Address Space Layout Randomization (ASLR)

Η χρήση του ASLR πρώτη φορά γίνεται από το λειτουργικό openBSD και στην συνέχεια εφαρμόζεται σωστά στο λειτουργικό FreeBSD. Η ίδια μέθοδος όσο αφορά την Apple χρησιμοποιείται τόσο στο λειτουργικό MAC OSX όσο και στο iOS. Ο τρόπος με τον οποίο ένας επιτιθέμενος προσπαθεί να ξεπεράσει το DEP είναι χρησιμοποιώντας κομμάτια κώδικα μέσω ROP. Ως εκ τούτου, για να επιτύχει χρειάζεται να γνωρίζει που βρίσκονται τα κομμάτια κώδικα στα πεδία μνήμης. Η μέθοδος ASLR κάνει αυτή την αναζήτηση, δύσκολη μιας ορίζει σε τυχαίες θέσεις τα αντικείμενα στη μνήμη. Όσο αφορά το iOS, όλες οι βιβλιοθήκες, τα stacks και οι διευθύνσεις μνήμης σωρού ορίζονται σε τυχαίες θέσεις στις στοίβες μνήμης. Όταν ένα λειτουργικό σύστημα χρησιμοποιεί DEP

και ASLR μηχανισμούς τότε δεν υπάρχει κάποιος εγγενής τρόπος να γράψει ο επιτιθέμενος κάποιο exploit. Στη πράξη χρειάζεται να βρει και να εκμεταλλευτεί δύο ευπάθειες. Μία για να καταφέρει να εκτελέσει κώδικα και μια για να καταφέρει να κάνει leak κάποια διεύθυνση μνήμης ώστε να εκτελέσει την μέθοδο ROP (Dai Zovi,D: 2011) .

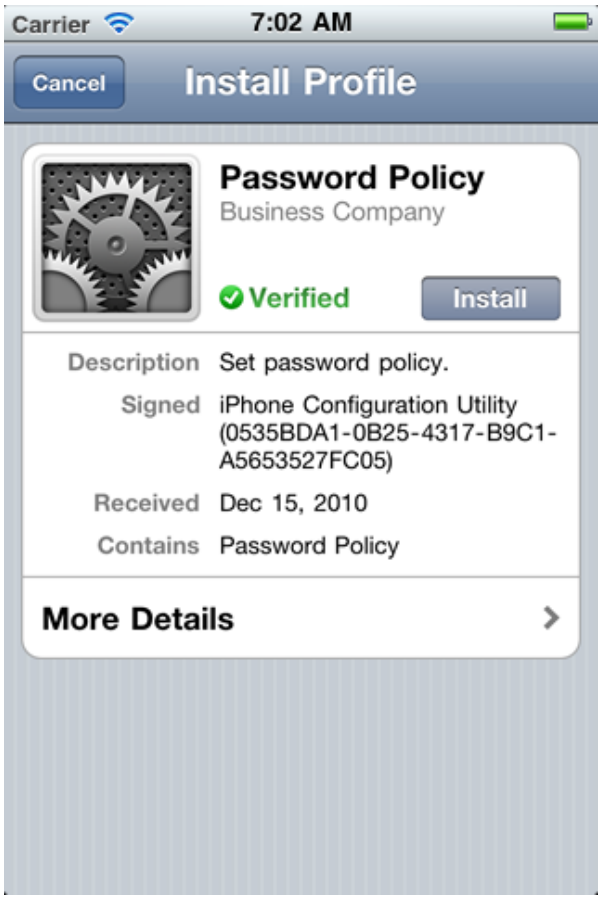
6.5. Εφαρμογές iOS

Οι εφαρμογές για τις έξυπνες συσκευές είναι από τα πιο κρίσιμα στοιχεία της αρχιτεκτονικής ασφαλείας των σύγχρονων κινητών και απασχολεί άμεσα τους ερευνητές ασφαλείας. Καθώς, οι εφαρμογές μπορεί να παρέχουν καταπληκτικά οφέλη παραγωγικότητας για τους χρήστες, μπορεί παράλληλα να δημιουργήσουν προβλήματα στην ασφάλεια, στην σταθερότητα και στην προστασία των δεδομένων του χρήστη αν δε χρησιμοποιηθούν και εξεταστούν ορθά. Εξαιτίας αυτών των λόγων που περιγράφηκαν παραπάνω το λειτουργικό iOS χρησιμοποιεί επίπεδα προστασίας, με σκοπό την προστασία των χρηστών προστατεύοντας και επιβλέποντας τις εφαρμογές με ψηφιακές υπογραφές από την ίδια την εταιρία και καθορίζοντας πια δικαιώματα πρέπει και οφείλει να έχει μια εφαρμογή. Αν οι εφαρμογές που προσπαθούν να ανεβούν στο ηλεκτρονικό κατάστημα της Apple δεν υπογραφούν από την ίδια την Apple ή από κάποια έμπιστη πηγή που χρησιμοποιεί κάποια ψηφιακή υπογραφή εγκεκριμένη από την Apple τότε οι εφαρμογές αυτομάτως απορρίπτονται και δεν ανεβαίνουν στο ηλεκτρονικό κατάστημα της Apple, προστατεύοντας τους χρήστες της από εφαρμογές που δεν είναι έμπιστες και μπορούν να βλάψουν την συσκευή, τα δεδομένα και την ιδιωτικότητα του χρήστη (Dai Zovi,D: 2011).

6.6. Mobile Configuration Profiles

Ένα configuration profile ουσιαστικά είναι ένα XML αρχείο το οποίο επιτρέπει την διανομή προφίλ ρυθμίσεων. Αν μια εταιρία χρειάζεται να διανέμει προφίλ ρυθμίσεων ειδικά διαμορφωμένες από την ίδια σε ένα εύρος συσκευών iOS ή για να παρέχει ειδικά διαμορφωμένες ρυθμίσεις email, network σε ένα εύρος συσκευών, τα προφίλ διαμόρφωσης είναι ο ευκολότερος δρόμος. Η Apple έδωσε την δυνατότητα στις εταιρίες να διαμορφώνουν την δική τους πολιτική ελέγχου εκμεταλλεύοντας τις μεθόδους προστασίας της συσκευής που παρέχει η Apple. Ένα προφίλ διαμόρφωσης παρέχει ρυθμίσεις όπως, τον περιορισμό λειτουργιών της συσκευής, ρυθμίσεις wifi, ρυθμίσεις VPN, ρυθμίσεις Email, ρυθμίσεις συναλλαγών, LDAP ρυθμίσεις, ρυθμίσεις για το

CalDAV ημερολογίο, κλειδιά και διαπιστευτήρια ειδικά διαμορφωμένα για τον χρήστη της συσκευής ή των συσκευών (Halbronn,C: 2010, Miller,C :2012).



Εικόνα 8: *Εγκατάσταση Configuration Profiles σε iOS*

6.7. Kernel

Το kernel που χρησιμοποιεί το λειτουργικό σύστημα iOS στον πυρήνα του δεν είναι βασισμένο στο Linux kernel που χρησιμοποιεί το Android αλλά είναι δημιουργημένο από την ίδια την Apple και ονομάζεται XNU. Το XNU kernel δημιουργήθηκε αρχικά από την NeXT και ουσιαστικά είναι ένα υβριδικό kernel που συνδυάζει το Mach micro-kernel και του 4.3BSD kernel. Ουσιαστικά είναι ένας πυρήνας που διαχειρίζεται APIs σε Objective-C και θεωρείται open-source project. Το kernel διαχειρίζεται τους οδηγούς που απαιτούνται για τη λειτουργία του hardware των συσκευών iOS, τις άδειες προστασίας των αρχείων και τα Frameworks που ονομάζονται TrustedBSD frameworks. Τέλος, το kernel ελέγχει τις εφαρμογές για την ψηφιακή υπογραφή της apple πριν τις εκκινήσει, την λειτουργία του sandbox καθώς και την ενδοεπικοινωνία των διεργασιών, το ASLR που αναφέραμε παραπάνω και τα δικαιώματα κάθε εφαρμογής καθώς και τους κανόνες που πρέπει να ακολουθούν τα συγκεκριμένα δικαιώματα.

7.Συμπεράσματα

Σκοπός του εβδόμου κεφαλαίου, είναι η ένωση όλων των συστατικών στοιχείων των παραπάνω κεφαλαίων με στόχο την ισότιμη σύγκριση των δύο λογισμικών με ίδια κριτήρια και κοινούς κανόνες. Με τα δεδομένα αυτά, δημιουργήθηκε ένας πίνακας σε μορφή γραφήματος δεδομένων (infographic) ώστε ο απλός χρήστης να έχει την δυνατότητα να γνωρίσει όσο πιο απλά γίνεται, τα προβλήματα, τις ανάγκες, τους στόχους, τις απόψεις άλλων χρηστών (σε στατιστικό επίπεδο), τους μηχανισμούς ασφαλείας και τις παροχές των δύο λειτουργικών συστημάτων. Για να γίνει όμως η παραπάνω σύγκριση, δυνατή, πρέπει όσα αναφέρθηκαν να κατηγοριοποιηθούν σε οχτώ διακριτές κατηγορίες με την κάθε κατηγορία να έχει τα δικά της κριτήρια. Οι κατηγορίες οι οποίες ορίστηκαν ώστε να διευκολυνθεί η σύγκριση είναι οι εξής:

- Γενικά προβλήματα
- Ψηφιακά καταστήματα
- Αναβαθμίσεις
- Τεχνικοί μηχανισμοί ασφαλείας
- Ερωτηματολόγιο χρηστών
- Μηχανισμοί προστασίας χρήστη
- Συνδυασμός τεχνικών ασφαλείας υλικού-λογισμικού
- Πεδίο επίθεσης

Οι παραπάνω κατηγορίες καθώς και τα κριτήρια τους αναφέρονται εκτενέστερα στις παρακάτω υποενότητες. Τέλος, τα κριτήρια είναι βασισμένα στις ενότητες των παραπάνω κεφαλαίων όπου αναλύονταν εκτενέστερα οι μηχανισμοί ασφαλείας, η ιστορία και η φιλοσοφία των δύο λογισμικών, τα προβλήματα που αντιμετωπίζουν κ.ο.κ.

7.1. Κατηγορία γενικών προβλημάτων

Πρώτη κατηγορία είναι η κατηγορία των γενικών προβλημάτων με κριτήρια τα εξής:

- Εταιρικό πλάνο
- Γλώσσα προγραμματισμού
- Εύρος συσκευών
- Εύρος επιθέσεων
- Συσκευές ανά λογισμικό
- Custom rom
- Bloatware συσκευών

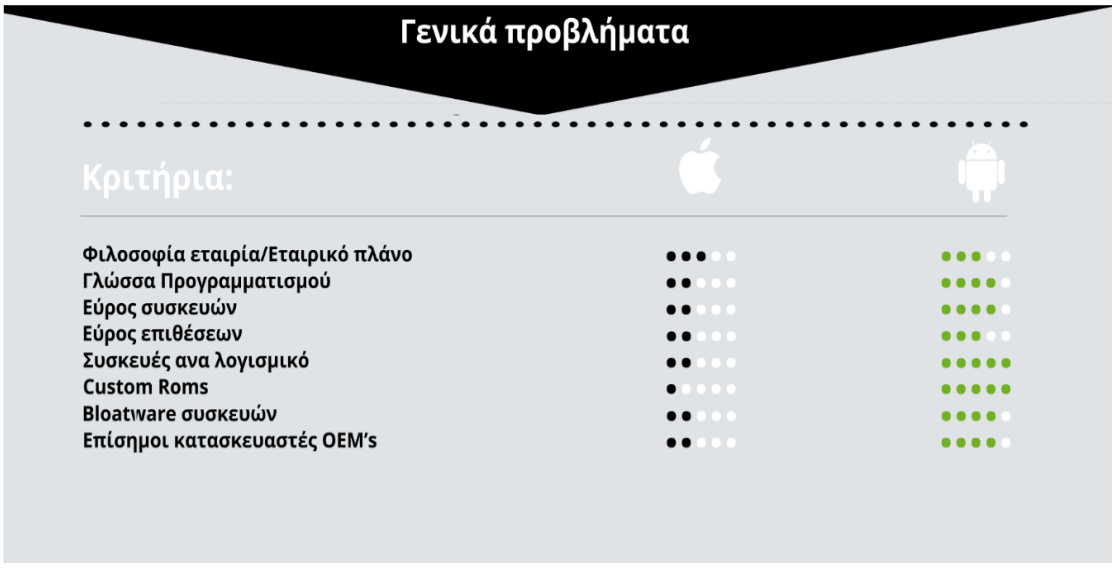
- Επίσημοι Κατασκευαστές OEM

Στόχος της πρώτης κατηγορίας είναι η σύγκριση των εταιρικών πλάνων και της φιλοσοφίας πίσω από τα λειτουργικά συστήματα Android και iOS καθώς και των γενικών προβλημάτων που αντιμετωπίζουν οι δύο εταιρίες.

Το λογισμικό Android σύμφωνα και με όσα αναφέρονται και στα παραπάνω κεφάλαια αποτελεί την επιτομή της καινοτομίας. Η δήλωση αυτή αν και βαρύγδουπη μπορεί να αιτιολογηθεί καθώς η Google ως εταιρία λόγο του επιχειρηματικού της μοντέλου δε φοβάται να δοκιμάσει καινούργιες μεθόδους και να στηριχτεί στην κοινότητα των χρηστών της. Από την οπτική των χρηστών βέβαια που ασχολούνται με την ασφάλεια, οι κινήσεις της Google εξετάζονται με ιδιαίτερο φόβο καθώς η μεγάλη ελευθερία και η γνώση του πηγαίου κώδικα από όλους δημιουργεί καινούργια ερωτήματα και καινούργιες ανησυχίες. Από την άλλη η χρήση της Java ως την κύρια γλώσσα προγραμματισμού του λογισμικού, μιας γλώσσας την οποία κατά πολλούς δε θεωρείται ασφαλές και από την οποία έχουν παρατηρηθεί διάφορες ευπάθειες σε πολλά λειτουργικά και λογισμικά έξω από τα όρια των κινητών συσκευών, δημιουργεί απορίες και κρίνεται ως επίφοβη. Εν συνεχεία το μεγάλο εύρος των συσκευών, καθώς και η αναλογία παραλλαγών του λογισμικού ανά συσκευή (Tablet, Smartphones, Wearables, Cars, κ.ο.κ) δημιουργεί ανησυχίες ως προς την ακεραιότητα του λογισμικού, αφού το εύρος επιθέσεων είναι μεγάλο, τόσο λόγο του ανοιχτού πηγαίου κώδικα της όσο και του μεγάλου βεληνεκούς του ονόματος της ίδιας της εταιρίας. Τέλος, οι εμπειρίες των χρηστών έχουν δείξει ότι η πλατφόρμα ακόμη δεν είναι ασφαλές ώστε ο χρήστης να επαναπαυτεί πλήρως στις μεθόδους-πρακτικές ασφαλείας της Google, κυρίως λόγω των πολλών OEM's με τους οποίους η ίδια η Google άμεσα ή έμμεσα συνεργάζεται και οι οποίοι προεγκαθιστώντας εφαρμογές οι οποίες εκτελούνται και δε διαγράφονται (Bloatware) δημιουργούν την επιτακτική ανάγκη των χρηστών να χρησιμοποιούν μη ασφαλές εκδόσεις, (είτε αυτές είναι Custom roms 3rd party, είτε επίσημων OEM's) του λειτουργικού Android, με στόχο την ταχύτητα και την ομαλότητα του λειτουργικού συστήματος, αφήνοντας σε δεύτερη μοίρα την ασφάλεια της συσκευής τους.

Από την άλλη η Apple, χρησιμοποιεί ένα κλειστό επιχειρησιακό μοντέλο, όπου το iOS είναι εσώκλειστο μέσα στις i-συσκευές της ομώνυμης εταιρίας. Βέβαια το μοντέλο αυτό δε μπορεί να χαρακτηριστεί ως ασφαλές καθώς αν αποτύχει ένας μηχανισμός ασφαλείας του λειτουργικού συστήματος ή άμα βρεθεί μια ευπάθεια στο λειτουργικό σύστημα, αυτό άμεσα ρίχνει το βάρος στην Apple αφού απευθείας όλο το λειτουργικό καθώς και οι εφαρμογές που βρίσκονται στο οικοσύστημα της βρίσκονται σε άμεσο

κίνδυνο. Η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η Objective-C, η οποία έχει χαρακτηριστεί ως αρκετά ασφαλής γλώσσας όταν βέβαια χρησιμοποιείται σε συνδυασμό με άλλους μηχανισμούς ασφαλείας. Βέβαια και η Objective-C δεν πρέπει να παρουσιάζεται ως η πιο ασφαλής γλώσσα προγραμματισμού, κυρίως λόγω των προβλημάτων ελέγχου πρόσβασης με τη χρήση των αντικειμένων που συμπυκνώνουν τα δεδομένα (Turner,S : 2014). Παράλληλα η κίνηση της Apple να χρησιμοποιήσει το Cocoa Touch αφαιρεί πολλές ιδιότητες της ίδιας της γλώσσας καθώς και της δυνατότητας της να λειτουργεί με άλλες γλώσσες προγραμματισμού, με αποτέλεσμα να μειώνει ακόμη περισσότερο το εύρος των επιθέσεων που μπορούν να επιτευχθούν στις συσκευές της. Ακόμη οι i-συσκευές είναι ελάχιστες και πλήρη ελεγχόμενες από την ίδια την εταιρία λόγω του εταιρικού της πλάνου, οι συνεργάτες (OEM's) είναι ελάχιστοι και πλήρως ελεγχόμενοι από την ίδια την Apple, ενώ Custom roms δεν υπάρχουν διαθέσιμες και οι προεγκατεστημένες εφαρμογές των i-συσκευών καθορίζονται και περιορίζονται από την ίδια την εταιρία και τις ανάγκες των χρηστών της. Τέλος αν και η Apple αποτελεί δεύτερη δύναμη στην τεχνολογική κοινότητα των έξυπνων κινητών συσκευών, δεν έχουν δημοσιοποιηθεί πολλές επιθέσεις απέναντι στην ίδια την εταιρία κυρίως λόγω του διαφορετικού της επιχειρηματικού της μοντέλου και του τρόπου με τον οποίο αντιμετωπίζει και προστατεύει το αγαθό της, που είναι οι χρήστες της.



Πίνακας 13: Κατηγορία γενικών προβλημάτων

7.2. Κατηγορία ψηφιακών καταστημάτων

Δεύτερη κατηγορία είναι η κατηγορία των ψηφιακών καταστημάτων, με κριτήρια τα εξής:

- Άδειες χρήσης
- Άδεια εξουσιοδότησης αγορών
- Ψηφιακές υπογραφές
- Ψηφιακά καταστήματα ανά λογισμικό
- Εύρος εφαρμογών
- Έλεγχος εφαρμογών
- Απομακρυσμένος έλεγχος-διαγραφή
- Malware ανά εφαρμογή

Στόχος της δεύτερης κατηγορίας είναι να συγκριθούν και να εξεταστούν οι μέθοδοι ασφαλείας που χρησιμοποιούν οι δύο εταιρίες ως προς τον έλεγχος και την ασφάλεια των εφαρμογών 3rd party κατασκευαστών που βρίσκονται στα ψηφιακά τους καταστήματα καθώς και τις μεθόδους προστασίας που χρησιμοποιούν σε περίπτωση εύρεσης κακόβουλης εφαρμογής.

Το λειτουργικό Android στην κατηγορία των ψηφιακών καταστημάτων έναντι του iOS δείχνει τις πραγματικές του αδυναμίες. Η Google λόγω του επιχειρησιακού της μοντέλου δίνει άδεια χρήσης του λειτουργικού της, ελεύθερα στο κοινό χωρίς καθόλου ελέγχους, στην προσπάθεια της να μεγαλώσει την κοινότητα των χρηστών της, πάντα βέβαια με καλές προθέσεις (Αναφέρονται στο κεφάλαιο 2). Όσο αφορά το ψηφιακό της κατάστημα δε χρησιμοποιεί καθόλου την μέθοδο των ψηφιακών υπογραφών καθιστώντας τον έλεγχο των εφαρμογών που «ανεβαίνουν» στο ηλεκτρονικό της κατάστημα σχεδόν άχρηστο, ενώ παράλληλα επιτρέπει την εγκατάσταση εφαρμογών εκτός του καταστήματος της στο λειτουργικό της σύστημα, στα οποία βέβαια δεν έχει καμία απολύτως δικαιοδοσία με αποτέλεσμα τον μηδενικό έλεγχο των εφαρμογών, την παραβίαση της εμπιστοσύνης των προγραμματιστών για προστασία των εφαρμογών τους από την ίδια την Google, την μεγάλη έξαψη κακόβουλων εφαρμογών, τόσο εντός όσο και εκτός τους ψηφιακού της καταστήματος. Λόγω των παραπάνω, η Google δε μπορεί και μάλλον δεν δύναται να διαγράφει και εφαρμογές από τις συσκευές των χρηστών στις οποίες έχουν παρατηρηθεί κακόβουλες ενέργειες χωρίς την έγκριση του χρήστη.

Από την άλλη η Apple έχει τον απόλυτο έλεγχο των αδειών χρήσης κάθε εφαρμογής, προστατεύει το ψηφιακό της κατάστημα με άδειες εξουσιοδότησης αγορών, παρέχει ψηφιακές υπογραφές τόσο στους προγραμματιστές του οικοσυστήματος της, τις οποίες ελέγχει σε τακτά διαστήματα, δεν επιτρέπει την δημιουργία, το «ανέβασμα» εφαρμογών και την χρήση τους σε συσκευές χρηστών, από ψηφιακά καταστήματα τρίτων, με αποτέλεσμα την ουσιαστική αχρηστία κάθε απόπειρας δημιουργίας και χρήσης ψηφιακών καταστημάτων πέρα από της ιδίας. Παράλληλα, επειδή ο αριθμός και των δύο καταστημάτων είναι μεγάλος, ο κίνδυνος μη ορθού ελέγχου ενέχει και στα δύο λογισμικά γέρνοντας βέβαια την ζυγαριά προς την μεριά της Apple λόγω κυρίως, των τεχνικών μέσων που χρησιμοποιεί και τα οποία αναφέρθηκαν εκτενέστερα στα κεφάλαια 3 και 5. Τέλος, δεν έχουν παρατηρηθεί περιπτώσεις κακόβουλων εφαρμογών στο ηλεκτρονικό κατάστημα της Apple, παρόλο που η ίδια η Apple έχει μεριμνήσει σε περίπτωση εύρεσης κακόβουλης εφαρμογής, διαγράφοντας απευθείας την εφαρμογή από το ηλεκτρονικό της κατάστημα ώστε να μη διαδοθεί σε άλλες συσκευές και διαγράφοντας την εφαρμογή από όσες συσκευές την έχουν κατεβάσει ενώ παράλληλα ελέγχει τις ζημιές που μπορεί να έχει δημιουργήσει.

Πίνακας 14: Κατηγορία Ψηφιακών καταστημάτων

7.3. Κατηγορία αναβαθμίσεις

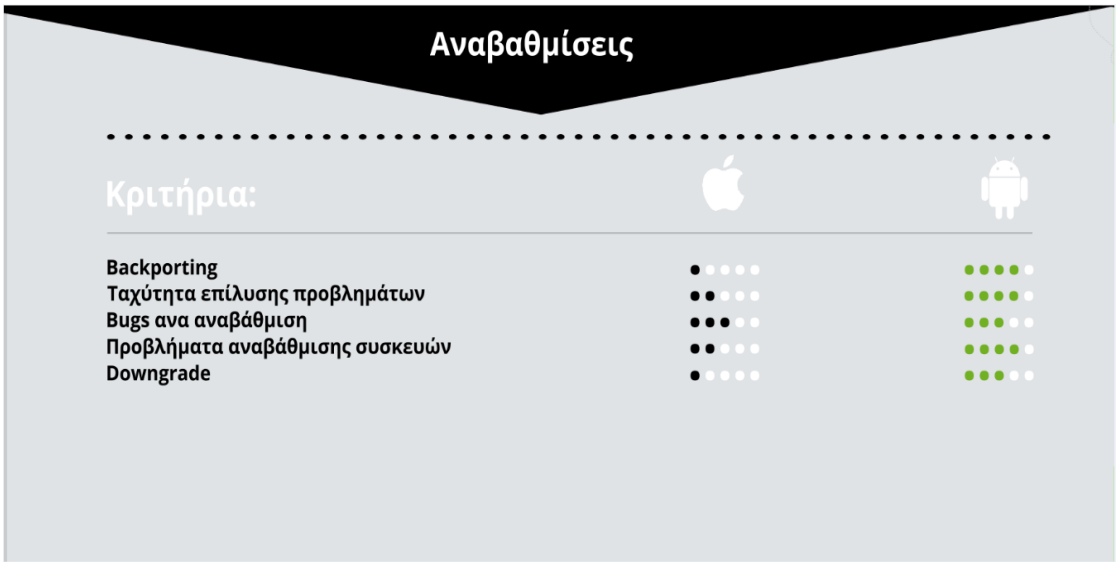
Τρίτη κατηγορία είναι η κατηγορία της διαδικασίας αναβάθμισης του λειτουργικού συστήματος με τα εξής κριτήρια:

- Backporting
- Ταχύτητα επίλυσης προβλημάτων
- Bugs ανά αναβάθμιση
- Προβλήματα αναβάθμισης συσκευών
- Downgrade

Σκοπός της κατηγορίας αυτής είναι να διερευνήσει τα προβλήματα ενημέρωσης του λογισμικού καθώς και της ταχύτητας επίλυσης προβλημάτων του λογισμικού από τις δύο εταιρίες.

Στο οικοσύστημα των Android το πρόβλημα της ενημέρωσης του λειτουργικού συστήματος είναι γνωστό σε όλους τους χρήστες της αλλά και στην ίδια την Google. Βέβαια μπορεί να δικαιολογηθεί η Google λόγω του επιχειρησιακού της μοντέλου και του δωρεάν χαρακτήρα του λειτουργικού συστήματος. Παρόλα αυτά τα ποσοστά backporting των συσκευών που χρησιμοποιούν το παραπάνω λειτουργικό είναι μεγάλα όπως φαίνεται και στο γράφημα στο κεφάλαιο 1 όπου οι περισσότερες συσκευές βρίσκονται ακόμη στην έκδοση 2.3.7. Gingerbread, δηλαδή σε έκδοση που σταμάτησε την υποστήριξη της η Google στις 21 Σεπτεμβρίου του 2011. Παράλληλα, η αργοπορημένη ταχύτητα επίλυσης των προβλημάτων που προκύπτουν στο λειτουργικό, από την Google και η ακόμη πιο αργή ενημέρωση του λειτουργικού συστήματος από εταιρίες όπως την Samsung, δημιουργεί μεγάλα Backports και μεγάλους κινδύνους στους χρήστες των συσκευών του οικοσυστήματος του Android. Ακόμη, μεγαλύτερο πρόβλημα αποτελεί το γεγονός ότι η μόνη εταιρία που απευθείας ενημερώνει το λειτουργικό σύστημα στην τελευταία έκδοση είναι η Google στην σειρά Nexus. Το βάρος βέβαια εδώ πέφτει στους ώμους των OEM's οι οποίοι δεν αναβαθμίζουν απευθείας τις συσκευές τους εξαιτίας πάλι του μοντέλου ενημέρωσης που ακολουθούν(90 μέρες στις ναυαρχίδες συσκευές και 7+ μήνες για τις υπόλοιπες). Τέλος, το downgrade στις Android συσκευές είναι δυνατό γιατί η Google όπως αναφέρθηκε στο πρώτο κεφάλαιο έχει ξεκλείδωτο Bootloader και δίνει άμεσα την δυνατότητα στους χρήστες της να αναβαθμίζουν ή να υποβαθμίσουν τις εκδόσεις του λειτουργικού συστήματος των συσκευών τους όποτε επιθυμούν καθιστώντας την ασφάλεια των συσκευών των χρηστών σε πολλές περιπτώσεις ανασφαλείς.

Παράλληλα στο οικοσύστημα του iOS δεν υπάρχουν τα ίδια προβλήματα με τα Android καθώς η αναβάθμιση των συσκευών γίνεται σχεδόν άμεσα στις i συσκευές και πάντα κατ'επιλογήν του χρήστη. Στην διαδικασία αναβάθμισης η Apple έχει λάβει υπόψη την προστασία των χρηστών σε περίπτωση που βρεθούν bugs έχοντας κρατήσει backup των αρχείων και του προ τελευταίου ενημερωμένου λογισμικού στο itunes. Τα bugs, βέβαια είναι ελάχιστα στις αναβαθμίσεις του λογισμικού αφού ακόμη και αν βρεθούν η Apple μέσα σε περίοδο δύο εβδομάδων κλείνει με patches όλες τις ευπάθειες που γνωστοποιούνται. Τέλος, όσο αφορά το Backporting, οι συσκευές της Apple είναι κατά 90% ενημερωμένες στην τελευταία έκδοση καθιστώντας το λογισμικό αρκετά ασφαλές όσο αφορά την διαδικασία ενημέρωσης του λογισμικού και επίλυσης των προβλημάτων στις συσκευές της.



Πίνακας 15: Κατηγορία αναβαθμίσεων

7.4. Κατηγορία τεχνικοί μηχανισμοί ασφαλείας

Τέταρτη κατηγορία είναι η κατηγορία των τεχνικών μηχανισμών ασφαλείας των δύο λειτουργικών συστημάτων με τις εξής κατηγορίες:

- Sandbox
- ASLR
- Δικαιώματα αρχείων
- Kernel
- Σύστημα κρυπτογραφίας
- Data execution prevention
- Δικαιώματα API
- Κλάσεις προστασίας

- Υιοθέτηση στοιχείων από το μοντέλο UNIX
- Μηχανισμός απομόνωσης διεργασιών
- Εικονικά μηχανήματα VMs

Σκοπός της κατηγορίας αυτής είναι να ερευνήσει τις τεχνικές μεθόδους που αναλύθηκαν στα κεφάλαια 5 και 6, των δύο λειτουργικών συστημάτων.

Το λειτουργικό σύστημα Android βασισμένο στο μοντέλο που κληρονομεί από τα Linux συστήματα φέρνει μαζί του μια άκρως μελετημένη Unix-like διαδικασία απομόνωσης διεργασιών καθώς και την αρχή των διαφόρων προνομιών, που διαχωρίζεται στους λιγότερο προνομιούχους χρήστες και στους περισσότερο προνομιούχους χρήστες. Ακόμη, χάρη στο μοντέλο UNIX το Android υιοθετεί δικαιώματα για εφαρμογές που αξιοποιούν οι προγραμματιστές για να φτάσουν στα κατώτερα στρώματα του λειτουργικού συστήματος, τα λεγόμενα API, ενώ επίσης υπάρχουν δικαιώματα για τα αρχεία συστήματος, καθώς και δικαιώματα IPC δηλαδή δικαιώματα για επικοινωνία μεταξύ των διαφόρων εφαρμογών του συστήματος. Εν συνεχεία, για να μπορεί το ίδιο το λειτουργικό σύστημα να καταλαβαίνει ποια εφαρμογή έχει υψηλού επιπέδου δικαιώματα και ποια είναι αυτά τα δικαιώματα, η Google δίνει την δυνατότητα στον προγραμματιστή να αναγράφει ποια δικαιώματα ζητάει από τον χρήστη να δώσει έγκριση για να γνωρίζει και το λειτουργικό σύστημα ότι επιτρέπεται να χρησιμοποιεί τους συγκεκριμένους πόρους. Το λειτουργικό σύστημα Android χρησιμοποιεί την λεγόμενη κρυπτογράφηση δημοσίου κλειδιού (Public key Cryptography) ή αλλιώς την κρυπτογράφηση ασύμμετρου κλειδιού (Asymmetric Cryptography) για πολλούς σκοπούς που σχετίζονται με τις εφαρμογές. Ως εκ τούτου, ένα μεγάλο μέρος του Sandbox του λειτουργικού συστήματος Android στηρίζεται σε μερικές βασικές έννοιες, όπως: α) την πρότυπη διαδικασία απομόνωσης του Linux, β) μοναδικές ταυτότητες χρήστη που προσδίδονται για τις περισσότερες εφαρμογές (UIDs), γ) περιορισμένα δικαιώματα του αρχείου συστήματος (file system permissions). Επιπρόσθετα το ASLR λειτουργεί στα Android από την έκδοση 4.0 και ακόμη βρίσκεται σε πειραματικό στάδιο σε σχέση με τον ανταγωνιστή (iOS) που χρησιμοποιεί μια αρκετά αναβαθμισμένη έκδοση ASLR στο λειτουργικό του σύστημα. Όσο αφορά το kernel που χρησιμοποιεί η Google στη καρδιά του Android λέγεται ότι περιέχει πάνω από 250 διαφορετικές τροποποιήσεις έχοντας την διαχείριση χαμηλού και υψηλού επιπέδου διεργασιών και ελέγχων όλου του συστήματος. Ακόμη, η Google ορίζει η ίδια τις κλάσεις προστασίας των αρχείων ως αναγνώσιμες, δίνοντας στο χρήστη την άμεση τροποποίηση και ανάγνωση τους μέσω του file manager του λειτουργικού. Τέλος, ο μηχανισμός

αντιγραφής zygote, επιτρέπει στο σύστημα να εξυπηρετεί όλες τις διεργασίες ταυτόχρονα καθώς και τις εφαρμογές γρηγορότερα, αφού δε θα χρειάζεται να επαναλαμβάνετε η ακριβές διαδικασία φόρτωσης. Η επιτυχία του Zygote βασίζεται στο γεγονός ότι τρέχει σε εικονικό μηχάνημα με το όνομα Dalvik το οποίο βέβαια δεν επηρεάζει την ασφάλεια της συσκευής αλλά λειτουργεί ως επιταχυντής διεργασιών και ως επιπρόσθετος έλεγχος των βασικών διεργασιών εκκίνησης.

Το λειτουργικό σύστημα iOS βασισμένο τόσο στο μοντέλο που κληρονομεί από τα Linux συστήματα όσο και από το καλά ανεπτυγμένο και καλά μελετημένο λειτουργικό σύστημα της Apple για τους προσωπικούς της υπολογιστές το OSX φέρνει μερικά χαρακτηριστικά που ασφαλίζουν τις φορητές της συσκευές. Το kernel, για παράδειγμα που βρίσκεται στην καρδιά του συστήματος των i συσκευών ουσιαστικά είναι ένας πυρήνας που διαχειρίζεται APIs σε Objective-C και θεωρείται open-source project. Το kernel του iOS δεν είναι βασισμένο στο linux αλλά στο λογισμικό openBSD, με μερικές βασικές ιδιομορφίες. Χάρη, στο πιο σημαντικό μηχανισμό ασφαλείας, αυτό των ψηφιακών υπογραφών της Apple το kernel ουσιαστικά ελέγχει την εγκυρότητα των υπογραφών αυτών πριν επιτρέψει την εκκίνηση οποιασδήποτε διεργασίας στο σύστημα. Το λειτουργικό σύστημα iOS, βέβαια όπως και το Android, διαχωρίζει τις διεργασίες χρησιμοποιώντας παραδοσιακούς μηχανισμούς αδειών για τα αρχεία υιοθετημένα από το μοντέλο UNIX. Διαχωρίζει δηλαδή τις άδειες χρήσης για τις διεργασίες ανά χρήστες και ανά ομάδες (UID/GID). Ακόμη, χρησιμοποιεί κλάσεις προστασίας δεδομένων (DPC) που επιτρέπει την ανάθεση μιας κλάσης προστασίας με διαφορετική πολιτική να καθορίζει πως προστατεύονται τα δεδομένα και αν πρέπει να είναι προσβάσιμα. Ακόμη, Η αρχιτεκτονική κρυπτογράφησης που χρησιμοποιεί η Apple που ονομάζεται «Apple's common crypto» ουσιαστικά παρέχει κοινά APIs σε όλους του προγραμματιστές που θέλουν να εντάξουν επιπρόσθετα επίπεδα ασφάλειας μέσω της κρυπτογράφησης στις εφαρμογές τους. Ενώ, το Sandbox, που υιοθετεί από το μοντέλο UNIX λειτουργεί ως τοίχος ασφαλείας σε περίπτωση που κάποιος μηχανισμός ασφαλείας αποτύχει. Επίσης, άλλη μια τεχνική ασφαλείας που χρησιμοποιεί το λειτουργικό iOS είναι ο μηχανισμός Data Execution Prevention (DEP) που λειτουργεί ως μια μέθοδος όπου ο επεξεργαστής μπορεί να ξεχωρίσει ποια κομμάτια μνήμης είναι εκτελέσιμος κώδικας και ποια κομμάτια μνήμης είναι δεδομένα. Τέλος, το λειτουργικό σύστημα iOS δε χρησιμοποιεί εικονικά μηχανήματα, ενώ ο μηχανισμός ASLR σε συνδυασμό με το DEP που υιοθετείται από το λειτουργικό της ομώνυμης εταιρίας MAC OS είναι γνωστό για την ανεπτυγμένη χρήση και τον περιορισμό των σφαλμάτων.

Πίνακας 16: Κατηγορία τεχνικών μηχανισμών ασφαλείας

7.5. Κατηγορία ερωτηματολόγιο χρηστών

Πέμπτη κατηγορία είναι η κατηγορία των ερωτηματολογίων χρηστών με τα εξής κριτήρια:

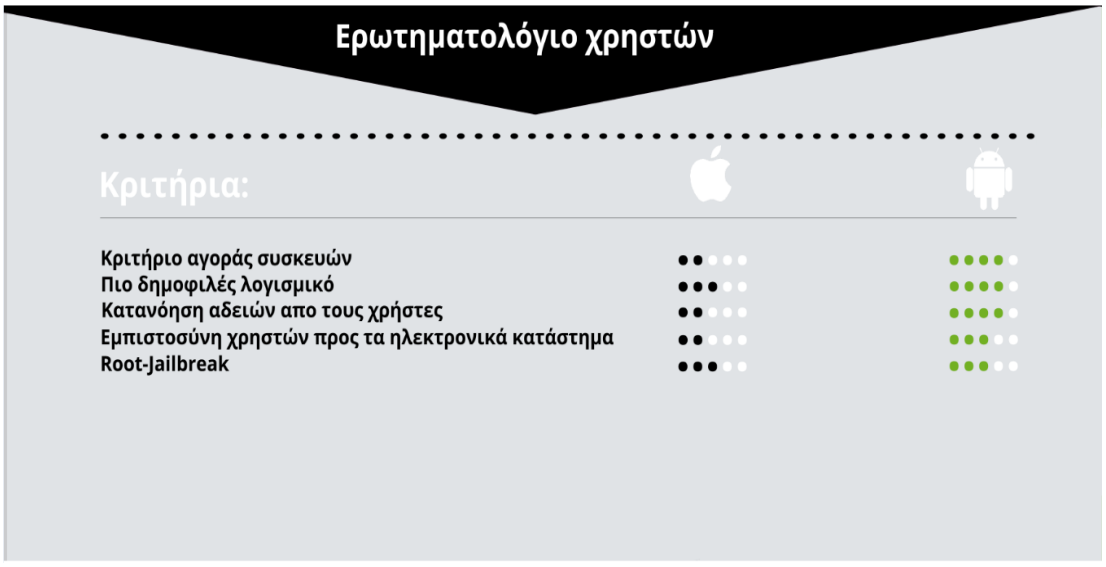
- Κριτήριο αγοράς συσκευής
- Πιο δημοφιλές λογισμικό
- Κατανόηση αδειών από τους χρήστες
- Εμπιστοσύνη χρηστών προς τα ηλεκτρονικά κατάστημα
- Root-Jailbreak

Σκοπός της κατηγορίας αυτής είναι η εξέταση και η αξιολόγηση των απαντήσεων των χρηστών όσο αφορά τα δύο μοντέλα ασφαλείας μέσα από την εμπειρία χρήσης των δύο λειτουργικών συστημάτων.

Το Android ως το πιο δημοφιλές λογισμικό με ποσοστό 74.2% , είναι και το λογισμικό που δημιουργεί βέβαια και τις μεγαλύτερες απορίες στο τομέα της ασφάλειας και της προστασίας των χρηστών της. Όσο αφορά το κριτήριο αγοράς συσκευών που χρησιμοποιούν το λειτουργικό σύστημα Android, οι χρήστες απάντησαν σε ποσοστό 59.2% ότι αγόρασαν την συσκευή τους με βασικό κριτήριο το κόστος. Εν συνεχεία, με ποσοστό 26,6% οι ίδιοι χρήστες έχουν προσπαθήσει να κάνουν Root την συσκευή τους, ενώ παράλληλα το 61,3% δεν γνωρίζει τους κινδύνους που δημιουργούνται όταν κάνουν Root την συσκευή τους. Τέλος, με ποσοστό 41,2% οι χρήστες δείχνουν την εμπιστοσύνη τους προς το ηλεκτρονικό κατάστημα, κυρίως λόγω του βαρύγδουπου ονόματος της εταιρίας. Τέλος, παρατηρείται με ποσοστό 74,6% ότι, οι χρήστες κατανοούν τις άδειες που

ζητάνε οι εφαρμογές από τους ιδίους, παρόλο που το ποσοστό, δεν αντικατοπτρίζει την πραγματικότητα λόγω κυρίως των ερωτημάτων που οι ίδιοι οι χρήστες είχαν κατά την διάρκεια διεξαγωγής του ερωτηματολογίου. Με τα αποτελέσματα αυτά μπορεί να ειπωθεί ότι οι χρήστες περιμένουν καλύτερης ποιότητας ασφάλεια από την Google, ενώ παράλληλα απολαμβάνουν την ελευθερία επιλογής των στοιχείων ασφαλείας καθώς και τον συνδυασμό απόδοσης-ασφάλειας-κόστους. Βέβαια, το γεγονός ότι το Android είναι το πιο δημοφιλές λογισμικό, δημιουργεί και την επιτακτική ανάγκη προστασίας των χρηστών της από κακόβουλες επιθέσης.

Το iOS ως δεύτερο πιο δημοφιλές λογισμικό της αγοράς με ποσοστό 16,36% χαρακτηρίζεται από την ποιότητα του λογισμικού του με ποσοστό 49,7%. Τα ποσοστά που αφορούν τους κινδύνους από το Jailbreak καθώς και της ερώτησης περί κατανόησης των αδειών είναι ίδια με τα ποσοστά που αναφέρθηκαν παραπάνω για τα Android, ενώ το ηλεκτρονικό κατάστημα της Apple φαίνεται να βαθμολογούν ως ασφαλές και πολύ ασφαλές με ποσοστό 57%. Τέλος, θεωρείται σημαντικό να ειπωθεί ότι σε μέτρηση 500 χρηστών το iOS θεωρείται ως το πιο ασφαλές λειτουργικό σύστημα. Όσο αφορά το iOS, οι χρήστες της δε χρειάζεται να γνωρίζουν για τις άδειες που ζητάνε οι εφαρμογές αφού η Apple της επιβλέπει γι'αυτούς, ενώ παράλληλα μπορούν να κλείνουν όποια άδεια θεωρούν ότι μια εφαρμογή δεν πρέπει να έχει διαμέσου των ρυθμίσεων του κινητού τους.



Πίνακας 17: Κατηγορία ερωτηματολογίων χρηστών

7.6. Κατηγορία μηχανισμοί προστασίας χρήστη

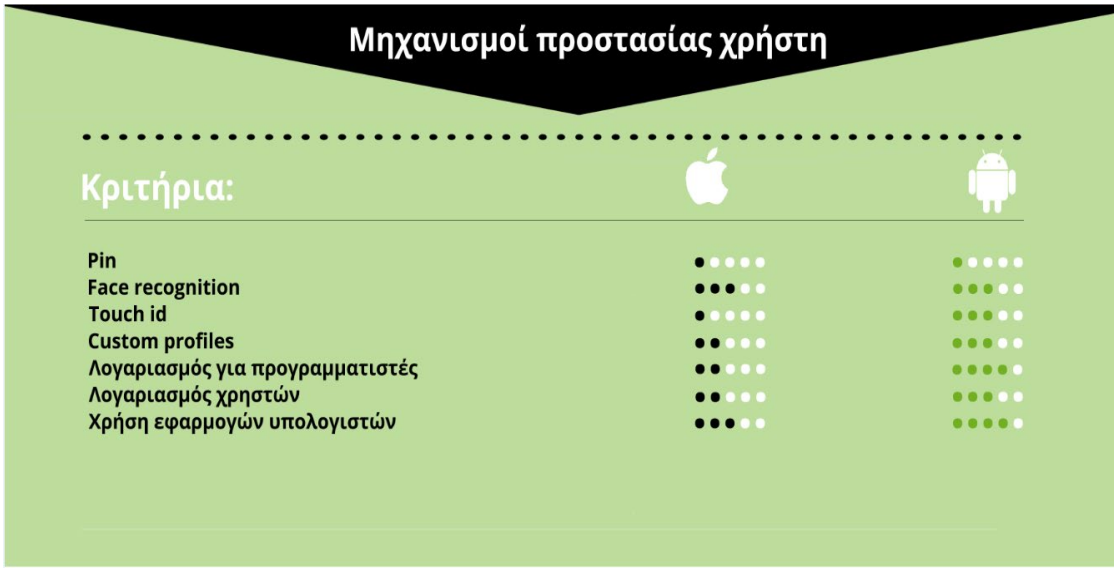
Έκτη κατηγορία είναι η κατηγορία των μηχανισμών προστασίας χρήστη με τα εξής κριτήρια:

- Pin
- Facerecognition
- Touch id
- Custom profiles
- Λογαριασμός για προγραμματιστές
- Λογαριασμός χρηστών
- Χρήση εφαρμογών υπολογιστών

Σκοπός της κατηγορίας αυτής είναι η ανάδειξη των μηχανισμών προστασίας που μπορούν οι ίδιοι οι χρήστες να αξιοποιήσουν με σκοπό την εξασφάλιση της ιδιωτικότητας των δεδομένων τους.

Στην κατηγορία αυτή οι μηχανισμοί που παρέχουν στους χρήστες στους και τα δύο λογισμικά είναι σχεδόν πανομοιότυπα. Για παράδειγμα η προστασία από μη εξουσιοδοτημένη χρήση με την χρήση του PIN ή Pattern είναι κοινή και για τα δύο λογισμικά. Η προστασία μέσω αναγνώρισης του προσώπου είναι επίσης κοινή και για τα δύο λογισμικά αλλά ακόμη βρίσκεται σε πρώιμο στάδιο. Από την άλλη το Android υστερεί στην προστασία των δεδομένων των χρηστών σε εταιρικό επίπεδο. Δεν έχει καθόλου Custom profiles που ελέγχονται από τον εργοδότη, αλλά παρέχει διαφορετικούς λογαριασμούς χρήστη στους οποίους για παράδειγμα ο κύριος χρήστης της συσκευής μπορεί να ελέγξει τις εφαρμογές, τα δεδομένα και τις κινήσεις κάποιου τρίτου που διαχειρίζεται την συσκευή του. Οι λογαριασμοί που δημιουργούν οι προγραμματιστές όπως αναφέρθηκαν και στο κεφάλαιο 2 ενεργοποιούνται ακαριαία μετά την καταβολή των 30 ευρώ αλλά δεν ελέγχονται επαρκώς από την Google. Το ίδιο ισχύει και για τους λογαριασμούς των χρηστών. Βέβαια η Google με την κίνηση αυτή προσπαθεί να προστατέψει την ανωνυμία των χρηστών της (μέρος του επιχειρησιακού της μοντέλου). Τέλος, όσο αφορά τα Android, η χρήση του υπολογιστή καθίσταται άχρηστη αφού δεν υπάρχει λογισμικό επίσημο από την εταιρία που να κρατάει αντίγραφα ασφαλείας του λογισμικού κ.ο.κ. Παρόλα αυτά εταιρίες όπως η Samsung έχουν προβεί στην δημιουργία τέτοιων λογισμικών που έπειτα από δοκιμή, δε μπορούν να θεωρηθούν ασφαλές, αφού ακόμη δεν λειτουργούν ορθά και παρουσιάζουν προβλήματα.

Από την άλλη η Apple με μια τελείως διαφορετική κατεύθυνση χρησιμοποιεί το Touch id από το iPhone 5s και έπειτα με μεγάλη επιτυχία. Διατηρώντας την ακεραιότητα, διαθεσιμότητα και εμπιστευτικότητα των πληροφοριών του χρήστη σε αρκετά υψηλό ποσοστό. Ακόμη, μέσω των custom profiles η Apple έδωσε την δυνατότητα στις εταιρίες να διαμορφώνουν την δική τους πολιτική ελέγχου εκμεταλλεύοντας τις μεθόδους προστασίας της συσκευής που παρέχει η Apple εξασφαλίζοντας στο άρτιο τον έλεγχο των δεδομένων, ρυθμίσεων και χρηστών που ενεργούν σε μια εταιρική συσκευή. Επίσης, η Apple κρίνει αναγκαίο τον έλεγχο όλων των προγραμματιστών που δημιουργούν εφαρμογές για το οικοσύστημα τους και ελέγχει σε τακτά διαστήματα τόσο τις εφαρμογές τους, όσο και την ακεραιότητα των στοιχείων που παρέχουν οι προγραμματιστές στους ίδιους, ώστε αν αναφερθεί παρατυπία ή εξαπάτηση του χρήστη, να μπορεί η Apple να αποδώσει ορθά ευθύνες. Τέλος, η Apple με την χρήση του itunes προστατεύει τόσο την συσκευή ώστε να αντιστοιχεί σε ένα και μοναδικό χρήστη, όσο και τα δεδομένα του με εικονικά αντίγραφα τα οποία αποθηκεύονται στον υπολογιστή του χρήστη τοπικά και τα οποία μπορούν να χρησιμοποιηθούν από τον ίδιο όταν και όποτε ζητηθεί.



Πίνακας 18: Κατηγορία μηχανισμών προστασίας χρήστη

7.7. Κατηγορία συνδυασμός τεχνικών ασφαλείας υλικού–λογισμικού

Έβδομη κατηγορία είναι η κατηγορία του συνδυασμού τεχνικών ασφαλείας υλικού-λογισμικού με τα εξής κριτήρια:

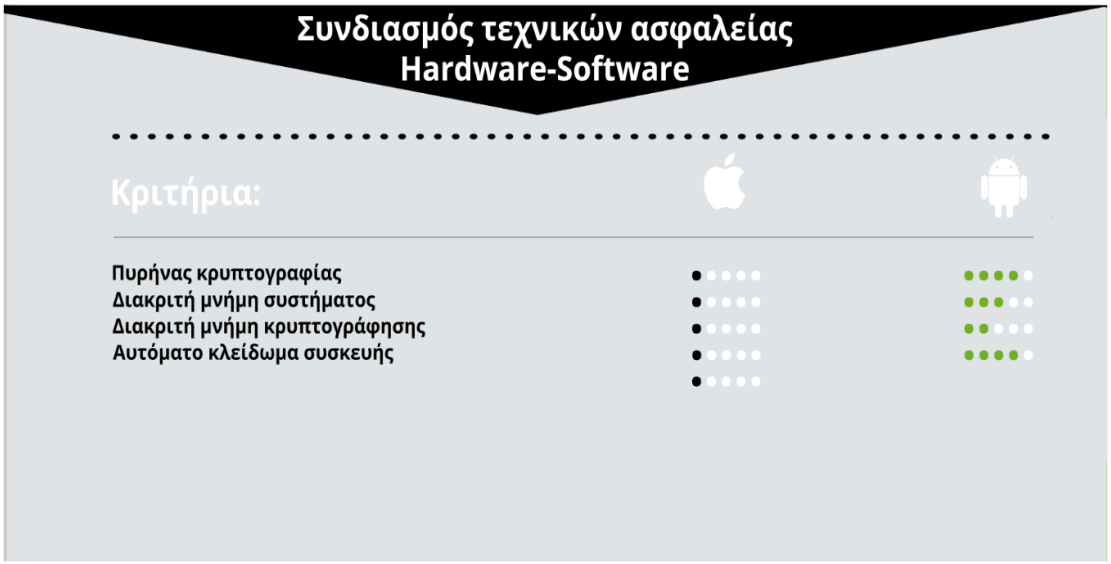
- Πυρήνας κρυπτογραφίας
- Διακριτή μνήμη συστήματος
- Διακριτή μνήμη κρυπτογράφησης

- Αυτόματο κλείδωμα συσκευής

Σκοπός της κατηγορίας αυτής είναι να εστιάσει στο πως συνδυάζονται σημαντικές τεχνικές ασφαλείας σε επίπεδο λογισμικού, για την ασφάλεια των δεδομένων του χρήστη σε επίπεδο υλικού.

Το Android, αρχικά δε χρησιμοποιεί ξεχωριστό πυρήνα κρυπτογράφησης καθώς η Google δεν είναι η εταιρία η οποία φτιάχνει το δικό της υλικό. Η παροχή διακριτού πυρήνα κρυπτογράφησης για την επιτάχυνση και προστασία της διαδικασίας κρυπτογράφησης βασικών αρχείων του συστήματος ή δεδομένων του χρήστη, εξαρτάται από τους επίσημους κατασκευαστές της και την ευχέρεια παροχής τέτοιων μεθόδων. Το Android, βέβαια χρησιμοποιεί διακριτή μνήμη συστήματος και κρυπτογράφησης, αλλά λόγω του μεγάλου βάρους του συστήματος στην ελεύθερη μνήμη και στις μεθόδους διαχωρισμού της ίδιας ποσότητας μνήμης σε διακριτή και προσβάσιμη από τους χρήστες, αχρηστεύει εμμέσως πλην σαφώς την διαδικασία της κρυπτογράφησης. Βέβαια το σύστημα σε χαμηλού επιπέδου διεργασίες κρυπτογραφεί συνέχεια, απλά η διαδικασία συνεχούς κρυπτογράφησης και αποκρυπτογράφησης δεδομένων στο λειτουργικό Android δε μπορεί να θεωρηθεί και το «ατού» του λογισμικού. Τέλος, σε περίπτωση που το κινητού χαθεί, η κλειδωθεί είναι ιδιαίτερα εύκολη η διαδικασία ξεκλειδώματος της συσκευής, κάνοντας ένα «hard restart» την συσκευή, διαγράφοντας βέβαια όλα τα αρχεία που είχε ο προηγούμενος ιδιοκτήτης.

Το iOS από την άλλη, επειδή είναι «προσκολλημένο» στις συσκευές της εταιρίας Apple η οποία δημιουργεί το λογισμικό στις προδιαγραφές του υλικού της, συνηθίζει να χρησιμοποιεί δύο πυρήνες για την επεξεργασία δεδομένων του χρήστη και παράλληλα να εκτελεί διεργασίες που απαιτούνται από το σύστημα για την εκκίνηση και την διατήρηση του και ένα ξεχωριστό από όλο το λογισμικό πυρήνα κρυπτογράφησης. Επίσης, χρησιμοποιεί διακριτή μνήμη συστήματος για χαμηλού επιπέδου διεργασίες και ξεχωριστή μνήμη που βρίσκεται ενσωματωμένη στην καρδιά του πυρήνα κρυπτογράφησης για την κρυπτογράφηση και αποκρυπτογράφηση δεδομένων και πληροφοριών. Όλοι οι κωδικοί του χρήστη φυλάγονται σε αυτή την ξεχωριστή μνήμη του συστήματος και ακόμη και αν κλαπεί η συσκευή δεν υπάρχει τρόπος ούτε από την ίδια την Apple να επαναφέρει τις πληροφορίες που βρίσκονται εσώκλειστες στην μνήμη. Τέλος, σε περίπτωση απώλειας ή κλοπής της συσκευής και μετά από προτροπή του χρήστη, η συσκευή κλειδώνει αυτόματα χωρίς να υπάρχει τρόπος, πέρα από την επανενεργοποίηση της από τον χρήστη, λειτουργίας της συσκευής.



Πίνακας 19: Κατηγορία συνδυασμού τεχνικών ασφαλείας υλικού-λογισμικού.

7.8. Κατηγορία πεδίο επίθεσης

Όγδοη και τελευταία κατηγορία είναι η κατηγορία του πεδίου επίθεσης με τα εξής κριτήρια:

- Υπηρεσίες
- Μέγεθος επιχείρησης
- Εμπιστοσύνη χρηστών
- Rooting-Jailbreak
- Προστασία εφαρμογών

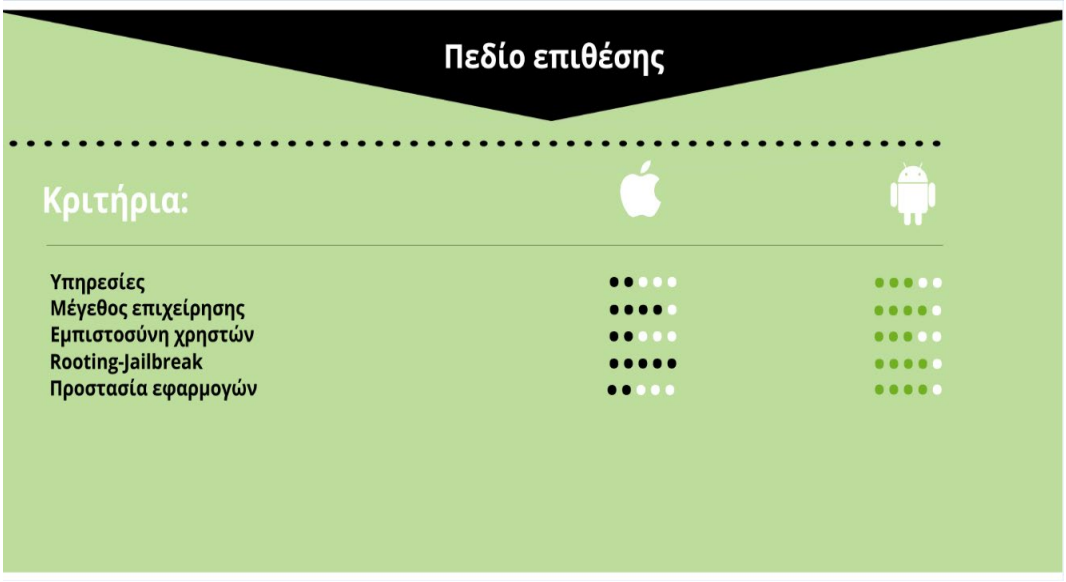
Σκοπός της τελευταίας κατηγορίας είναι να εξηγήσει και να ερευνήσει τις ευκαιρίες επίθεσης που βρίσκουν οι Hackers, την εμπιστοσύνη των χρηστών απέναντι στους δύο κολοσσούς, το rooting και το Jailbreak και την προστασία των εφαρμογών, καθώς και τις υπηρεσίες που οι δύο υπηρεσίες προσφέρουν.

Όσο αφορά την Google είναι μια εταιρία πολύ γνωστή στο τομέα του ηλεκτρονικού Διαδικτύου με αρκετές δωρεάν υπηρεσίες που καλύπτουν ένα μεγάλο φάσμα των εργασιών της καθημερινότητας του χρήστη. Σήμερα, η Google κατέχει το μεγαλύτερο Cloud Server με πάνω από 500.000 CPUs. Θεωρείτο, ως καινοτόμος εταιρία και το μοντέλο ανάπτυξης της σχετίζεται άμεσα με την εμπιστοσύνη που της επιδεικνύουν οι χρήστες της. Το αίσθημα ελευθερίας που προσδίδει όμως στο λειτουργικό Android, επιτρέποντας στους χρήστες της είτε είναι «γνώστες», είτε όχι να «πειράξουν» τις συσκευές τους, ώστε να τους ικανοποιεί το αποτέλεσμα, κάνοντας τις συσκευές τους όσο πιο προσωπικές γίνεται, δημιουργεί κινδύνους, όσο αφορά την γενικότερη ασφάλεια του λογισμικού. Το

ίδιο το κοινό της Google γνωρίζει ότι το Android δεν είναι το πιο ασφαλές λειτουργικό. Σε μεγάλο ποσοστό 41.2% , όμως πιστεύουν ότι το Playstore είναι ασφαλές και ότι η Google τους προστατεύει, λόγω του ονόματος της. Βέβαια, η Google δε φαίνεται να πτοείται αφού με δηλώσεις στο Google I/O του 2015 ο CEO της εταιρίας αναφέρει ότι το Android είναι πιο ασφαλές πλατφόρμας από ότι το iOS. Ξεχνώντας, βέβαια να αναφερθεί στα προβλήματα Backporting και στην ανάπτυξη Malware στο Playstore καθώς και σε όλα τα προβλήματα που επιδυκνύει η πλατφόρμα από το open source μοντέλο που ακολουθεί. Βέβαια, η Google φαίνεται πως τα τελευταία χρόνια για να σταματήσει το Backporting στις συσκευές της, ωθεί δύο καινούργια μοντέλα. Το λεγόμενο Android ONE το οποίο είναι φθηνές κινητές συσκευές που αναβαθμίζονται απευθείας από την Google και οι οποίες χρησιμοποιούν το λογισμικό της Google απευθείας από το AOSP, χωρίς τροποποιήσεις. Το δεύτερο μοντέλο είναι η μεταστροφή της στο κλειστό επιχειρησιακό μοντέλο, αφού πλέον ακόμη και η εφαρμογή «ημερολόγιο» που βρισκόταν μέσα στις εφαρμογές της σουίτας που πρόσφερε το Android στο AOSP απευθείας, μπορεί να την κατεβάσει ο χρήστης από το ψηφιακό της κατάστημα και να αναβαθμίζεται ακαριαία από την ίδια την Google. Το ίδιο ισχύει και για την εφαρμογή SMS και Τηλέφωνο που θεωρούνται εφαρμογές που εμπεριέχονται a priori ως προ εγκατεστημένες σε **κάθε** λογισμικό και οι οποίες δεν αναβαθμίζονται.

Από την άλλη η Apple, έχει δημιουργήσει μια κοινότητα χρηστών που χρησιμοποιούν το μοτίβο του ότι πληρώνεις, παίρνεις. Η Apple δημιουργεί και πουλάει προϊόντα με το μοτίβο, ο χρήστης πάνω από όλα και ότι κάθε συσκευή και κάθε χρήστης είναι μοναδικός. Το υψηλό κόστος των συσκευών της σε συνδυασμό με την άρτια ομάδα επίλυσης προβλημάτων ασφαλείας της εταιρίας και τη φιλική προς το χρήστη, χρήση της διεπιφάνειας χρήσης του λειτουργικού της, φαντάζει ως το ιδανικό πακέτο. Επειδή, όμως το κόστος είναι αρκετά υψηλό για τις συσκευές της ομώνυμης εταιρίας και επειδή η ποιότητα της ασφάλειας των συσκευών τους είναι αρκετά υψηλή λόγω του ότι θέλουν πρωτίστως οι χρήστες τους να έχουν την καλύτερη εμπειρία χρήσης (ιός = μη ομαλότητα χρήσης του λειτουργικού) οι Hackers μεν δυσκολεύονται να βρουν κάποια ευπάθεια αφού θα έπρεπε να έχουν και MAC υπολογιστή, που επίσης είναι ακριβή συσκευή για να γράψουν τον ιό τους και αφετέρου δεν υπάρχει επαρκής λόγος να ασχοληθούν με την Apple, αφού το πεδίο επίθεσης τους απέναντι στο Android (περισσότεροι χρήστες) είναι πολύ μεγαλύτερο και οι ιοί που μπορούν να δοκιμάσουν και οι οποίοι είναι γραμμένοι σε Java και μπορούν να εγκατασταθούν χωρίς το Playstore, τις καθιστούν ιδανικές περιπτώσεις για κακόβουλες επιθέσεις. Βέβαια, όταν ο χρήστης κάνει Jailbreak την

συσκευή του για να αποκτήσει απόλυτη ελευθερία, όλα όσα έχουν προαναφερθεί στα παραπάνω κεφάλαια και αφορούν την ασφάλεια του λογισμικού πλέον καταργούνται και η συσκευή θεωρείται πλέον επικίνδυνη (βλέπε κεφάλαιο 4.3).



Πίνακας 20: Κατηγορία πεδίου επίθεσης

8.Επίλογος

Η ασφάλεια των λογισμικών είναι σαν ένα παιχνίδι σκάκι. Πτυχιακές τέτοιου είδους μπορεί να δώσουν το στίγμα στρατηγικής των εταιριών της εποχής και ειδικότερα όταν στο αποτέλεσμα του παιχνιδιού αυτού εμπεριέχεται, η ιδιωτικότητα και η προστασία των προσωπικών δεδομένων των χρηστών και των εταιριών. Είναι γνωστό, ότι κακόβουλοι χάκερ πάντα θα ψάχνουν για νέους τρόπους να επιτεθούν τις εφαρμογές, ειδικά αυτές που εμπεριέχουν πληροφορίες υψηλής αξίας. Καθώς, βέβαια, στην αγορά θα παρουσιάζονται εφαρμογές κρίσιμης αξίας για τις εταιρίες και τους χρήστες της, τόσο θα υπάρχει η επιτακτική ανάγκη εξέλιξης των μηχανισμών και μεθόδων ασφαλείας από τις δύο εταιρίες (Apple-Google). Σε μια εποχή που η τεχνολογία ακμάζει με ραγδαίους ρυθμους, δημιουργείται η ανάγκη μελέτης τόσο των μεθόδων όσο και των πρακτικών που ακολουθούν οι εταιρίες για την διασφάλιση της ιδιωτικότητας των χρηστών τους. Συμπερασματικά, το μοντέλο του iOS παρουσιάζεται ως το πιο ασφαλές μοντέλο ασφάλειας, αφού η ίδια η Apple έχει τον πλήρη έλεγχο της συσκευή και του περιβάλλοντος ανάπτυξης των εφαρμογών του οικοσυστήματος της. Παρόλα αυτά, το μοντέλο του Android της Google έχει την μεγαλύτερη προοπτική για ασφάλεια σε επίπεδο εφαρμογών αφού βασίζεται στον ανοιχτό χαρακτήρα του NDK της. Δηλαδή, στην ανοιχτή κοινότητα και πλατφόρμα κατασκευής εφαρμογών. Όμως, κανένα από τα δύο λογισμικά δεν μπορεί επουδενί να θεωρηθεί απολύτως ασφαλές και για αυτό δε μπορεί να αντικαταστήσει πλήρως τους προσωπικούς υπολογιστές. Βέβαια, περαιτέρω μελέτη και ανάπτυξη των παραπάνω μεθόδων, θα χρειαστεί ώστε να δημιουργηθεί ένα ολοκληρωμένο μοντέλο ασφάλειας εφαρμογών για περιβάλλοντα κινητών συσκευών.

8.1. Επόμενα Βήματα

Με βάση τα όσα αναφέρθηκαν στα προηγούμενα κεφάλαια προκύπτουν μια σειρά από ερευνητικά ζητήματα που αποτελούν και τα επόμενα βήματα της παρούσας έρευνας.

Στη πορεία της έρευνας θα γίνει μια προσπάθεια δημιουργίας μιας κοινότητας χρηστών που μέσα από τις προτάσεις τους, να παραχθεί ένα μοντέλο ασφάλειας που θα ελέγχεται από τους ίδιους τους χρήστες και θα διασφαλίζει την ιδιωτικότητα των δεδομένων τους. Επίσης, δεύτερος βασικός άξονας είναι η μελέτη των υφιστάμενων μοντέλων και η αναζήτηση των ελαττωμάτων τους καθώς και η καλυτέρευση των

τεχνικών μηχανισμών ασφαλείας των συστημάτων σε ένα λειτουργικό σύστημα βασισμένο στον ανοιχτό πηγαίο κώδικα του Android.

Βιβλιογραφία

Aiken, M., Fähndrich, M., Hawblitzel, C., Hunt, G., & Larus, J. (2006). Deconstructing process isolation. In Proceedings of the 2006 workshop on Memory system performance and correctness (pp. 1-10). ACM.

Benenson, Z., Gassmann, F., & Reinfelder, L. (2013). Android and iOS users' differences concerning security and privacy. In CHI'13 Extended Abstracts on Human Factors in Computing Systems (pp. 817-822). ACM.

Burnette, E. (2009). Hello, Android: introducing Google's mobile development platform. Pragmatic Bookshelf.

Curry, D. A. (1992). UNIX system security

Dai Zovi, D. A. (2011). Apple iOS 4 security evaluation. Black Hat USA.

Developers, A. (2011). What is Android.

Drake, J. J., Lanier, Z., Mulliner, C., Fora, P. O., Ridley, S. A., & Wicherski, G. (2014). Android Hacker's Handbook. John Wiley & Sons.

Egele, M., Kruegel, C., Kirda, E., & Vigna, G. (2011). PiOS: Detecting Privacy Leaks in iOS Applications. In NDSS.

Enck, W., Octeau, D., McDaniel, P., & Chaudhuri, S. (2011). A Study of Android Application Security. In USENIX security symposium (Vol. 2, p. 2).

Enck, W., Ongtang, M., & McDaniel, P. D. (2009). Understanding Android Security. IEEE security & privacy, 7(1), 50-57.

Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (pp. 627-638). ACM.

Goadrich, M. H., & Rogers, M. P. (2011). Smart smartphone development: iOS versus android. In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 607-612). ACM.

Gritzalis, D., Gritzalis, S. & Katsikas, S. (2004). Information systems security. New Technology, Greece

Halbronn, C., & Sigwald, J. (2010). iPhone security model & vulnerabilities. InProceedings of Hack in the box sec-conference. Kuala Lumpur, Malaysia.

Hoog, A., & Strzempka, K. (2011). iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices. Elsevier.

Khanna, S. O., & Patel, P. N. (2011). ANDROID MOBILE SECURITY-AN ISSUE OF FUTURE. International Journal of Advanced Research in Computer Science, 2(5).

Miller, C., Blazakis, D., DaiZovi, D., Esser, S., Iozzo, V., & Weinmann, R. P. (2012). IOS Hacker's Handbook. John Wiley & Sons.

Miller, C. (2011). Mobile attacks and defense. Security & Privacy, IEEE, 9(4), 68-70.

Seriot, N. (2010). iPhone Privacy. In Black Hat USA

Turner, S. (2014). Security vulnerabilities of the top ten programming languages: C, Java, C++, Objective-C, C#, PHP, Visual Basic, Python, Perl, and Ruby. *Journal of Technology Research*, *5*, 1.