Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Κατεύθυνση

«Τεχνολογίες Διαχείρισης Ασφάλειας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ψηφιακή εγκληματολογία μνήμης σε συσκευές Android** **Live memory forensics on Android devices**
Ονοματεπώνυμο Φοιτητή	**Σπυρίδων Γενετζάκης**
Πατρώνυμο	**Ιωάννης**
Αριθμός Μητρώου	**ΜΠΣΠ15016**
Επιβλέπων	**Παναγιώτης Κοτζανικολάου** **Επίκουρος Καθηγητής**

Ημερομηνία Παράδοσης Οκτώβριος 2018

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Παναγιώτης Κοτζανικολάου Επίκουρος Καθηγητής	Μιχαήλ Ψαράκης Επίκουρος Καθηγητής	Δημήτριος Αποστόλου Αναπληρωτής Καθηγητής

This page intentionally left blank

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα επίκουρο καθηγητή κ. Παναγιώτη Κοτζανικολάου καθώς και τον κ. Αλέξανδρο Βασιλαρά για την βοήθεια και τις πολύτιμες συμβουλές τους αναφορικά με το θέμα της ψηφιακής εγκληματολογίας σε συσκευές Android.

Επίσης, θα ήθελα να ευχαριστήσω τους συμφοιτητές και συνοδοιπόρους μου για την άψογη συνεργασία που είχαμε καθ' όλη την διάρκεια του μεταπτυχιακού προγράμματος.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου, οι οποίοι με βοήθησαν και με στήριξαν στη διάρκεια του μεταπτυχιακού.

Περίληψη

Στην παρούσα μεταπτυχιακή διατριβή παρουσιάζονται και αναλύονται τόσο τα θεωρητικά όσο και τα τεχνικά στοιχεία που περιλαμβάνονται στην επιστήμη της «Ψηφιακής εγκληματολογίας μνήμης σε συσκευές Android». Στο θεωρητικό μέρος, αρχικά γίνεται ανάλυση του περιβάλλοντος Android (λειτουργικό σύστημα, αρχιτεκτονική, σύστημα αρχείων κτλ.) και στη συνέχεια αναλύονται οι τρόποι εξαγωγής των δεδομένων από συσκευές τύπου Android. Συνεχίζοντας, περιγράφονται αναλυτικά τα βήματα που ακολουθούνται για την πραγματοποίηση μίας ψηφιακής εξέτασης φορητής συσκευής. Έπειτα, περιγράφονται και αναλύονται διάφορα εμπορικά και ανοιχτού κώδικα εργαλεία που χρησιμοποιούνται στην ψηφιακή εγκληματολογία και αναφέρεται επιγραμματικά τι είδους πληροφορίες μπορούν να ανακτηθούν με την χρήση τους. Στην συνέχεια, γίνεται εκτενέστερη αναφορά στην ψηφιακή εγκληματολογία μνήμης και αναφέρονται οι διαφορές μεταξύ «ζωντανής(live)» και «νεκρής(dead)» ψηφιακής εγκληματολογίας, καθώς και τα προβλήματα/προκλήσεις που συνεπάγεται η ψηφιακή εγκληματολογία μνήμης σε περιβάλλον Android. Στο πρακτικό μέρος της διατριβής, αναλύεται ένα δείγμα μνήμης από συσκευή Android το οποίο αποκτήθηκε με την χρήση του LiME.

Λέξεις Κλειδιά: ψηφιακή εγκληματολογία μνήμης, Android συσκευές, έξυπνες κινητές συσκευές, «live» και «dead» ψηφιακή εγκληματολογία μνήμης.

Abstract

The present dissertation presents and analyzes both the theoretical and the technical information included in the science of "Live Memory Forensics on Android Devices". In the theoretical part, the Android environment (operating system, architecture, file system, etc.) is being presented and we analyze the methods used to export data from Android devices. Continuing, we mention and detail the steps that an examiner should follow to perform a digital forensic analysis of a mobile device. Then, various commercial and open source tools are mentioned with references to the information that can be recovered with the usage of each one. Subsequently, there is a more extensive reference to android memory forensics and the differences between "live" and "dead" digital forensics approach, as well as the problems / challenges posed by memory forensics in an Android environment. In the practical part of the dissertation, we analyze a memory dump from an Android device that was acquired using the tool LiME.

Keywords: live memory forensics, android devices, smartphones, "live" and "dead" memory forensics.

Πίνακας Περιεχομένων

1. Εισαγωγή 11

1.1 Εξέλιξη των έξυπνων κινητών συσκευών 11

1.2 Ορισμοί και έννοιες της ψηφιακής εγκληματολογίας 12

1.3 Διάρθρωση της μελέτης..... 13

2. Το λειτουργικό σύστημα Android 15

2.1 Η αρχιτεκτονική του λειτουργικού συστήματος Android 15

2.1.1 Ο πυρήνας του Linux (The Linux Kernel)..... 17

2.1.2 Το επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer) 17

2.1.3 Το επίπεδο εκτέλεσης Android (Android Runtime) 17

2.1.4 Το επίπεδο βιβλιοθηκών C/C++ (Native C/C++ Libraries)..... 18

2.1.5 Το επίπεδο διεπαφής προγραμματισμού εφαρμογών Java (Java API Framework) 18

2.1.6 Το επίπεδο εφαρμογών συστήματος (System Apps) 19

2.2 Οι εκδόσεις του λειτουργικού συστήματος Android 19

2.3 Τα αρχεία συστήματος στο λειτουργικό σύστημα Android 22

2.4 Η καταχώρηση δεδομένων στο λειτουργικό σύστημα Android 24

3. Ψηφιακά πειστήρια κινητών συσκευών 25

3.1 Εισαγωγή 25

3.2 Αποθήκευση δεδομένων στο λειτουργικό σύστημα Android..... 25

3.3 Τρόποι εξαγωγής δεδομένων της εσωτερικής μνήμης 29

3.4 Τρόποι εξαγωγής δεδομένων της κάρτας μνήμης(sd card) 32

3.5 Τρόποι εξαγωγής δεδομένων της κάρτας SIM..... 32

3.6 Τρόποι εξαγωγής δεδομένων της μνήμης RAM 33

3.7 Πληροφορίες που μπορούν να ανακτηθούν 33

4. Βήματα εξέτασης ψηφιακών πειστηρίων κινητών συσκευών 35

4.1 Εισαγωγή 35

4.2 1η φάση: Intake(προσκόμιση) 36

4.3 2η φάση: Identification(αναγνώριση) 36

4.4 3η φάση: Preparation(προετοιμασία) 36

4.5 4η φάση: Isolation(απομόνωση) 36

4.6 5η φάση: Processing(επεξεργασία) 37

4.7 6η φάση: Verification(επαλήθευση) 37

4.8 7η φάση: Documentation / Reporting(τεκμηρίωση / αναφορά) 37

4.9 8η φάση: Presentation(παρουσίαση) 38

4.10 9η φάση: Archiving(αρχειοθέτηση) 38

5. Εργαλεία εξέτασης ψηφιακών πειστηρίων σε περιβάλλον Android . 39

5.1 Εισαγωγή 39

5.2 Εργαλεία λογικής απόκτησης (logical acquisition) 39

5.2.1 Το εργαλείο SAFT 39

5.2.2 Το εργαλείο LiME 40

5.2.3 Το εργαλείο Nandroid Backup 40

5.2.4 Η σουίτα εργαλείων OSAF-TK (Open Source Android Forensics – Tool Kit) 40

5.2.5 Το Santoku Linux 40

5.2.6 Το εργαλείο Andriller 41

5.2.7 Το εργαλείο LiMEaide 41

5.3 Εργαλεία φυσικής απόκτησης (physical acquisition) 42

5.3.1 Το εργαλείο UFED Ultimate 42

5.3.2 Το εργαλείο Belkasoft Evidence Center 42

5.3.3 Το εργαλείο MOBILedit Forensic Express 43

5.3.4 Το εργαλείο Oxygen Forensic Detective 44

5.3.5 Το εργαλείο XRY – Extract 44
5.3.6 Το εργαλείο Paraben E3: DS 44
5.3.7 Τα εργαλεία AccessData MPE+ and nFIELD 45
5.4 Εργαλεία ανάλυσης της μνήμης (memory analysis) 45
5.5 Εργαλεία ανάλυσης της ζωντανής μνήμης (live memory analysis) .. 45
5.5.1 Το Volatility Framework 45
5.5.2 Το εργαλείο Autopsy 46
5.5.3 Το Rekall Framework 46
6. Ψηφιακή εγκληματολογία «ζωντανής» μνήμης (Live Memory Forensics) 47
6.1 Εισαγωγή 47
6.2 Προδιαγραφές εργαλείων για την απόκτηση της μνήμης RAM 47
6.3 Διαφορές μεταξύ «dead» και «live» ψηφιακής εγκληματολογίας 48
6.3.1 Στατική ανάλυση («dead» analysis) 48
6.3.2 Δυναμική ανάλυση («live» analysis) 48
6.3.3 Διαφορές μεταξύ των δύο προσεγγίσεων 49
6.3.4 Προβλήματα/προκλήσεις ψηφιακής εγκληματολογίας μνήμης σε συσκευές Android 49
7. Εξέταση «ζωντανής» μνήμης κινητής συσκευής Android 51
7.1 Εισαγωγή 51
7.2 Διαδικασία εξέτασης «ζωντανής» μνήμης 51
8. Επίλογος 67
8.1 Σύνοψη 67
8.2 Συμπεράσματα 67
9. Βιβλιογραφία – Πηγές 69

This page intentionally left blank

1. Εισαγωγή

1.1 Εξέλιξη των έξυπνων κινητών συσκευών

Στη σημερινή εποχή, οι φορητές συσκευές έχουν εξελιχθεί και ουσιαστικά αποτελούν μικρούς φορητούς ηλεκτρονικούς υπολογιστές με δυνατότητες εφάμιλλες και μερικές φορές καλύτερες από τους απλούς καθημερινούς ηλεκτρονικούς υπολογιστές. Τα έξυπνα τηλέφωνα (smart phones) είναι συσκευές που μπορούν να εκτελέσουν όλες τις κλασσικές λειτουργίες ενός παλιού τηλεφώνου (π.χ. να πραγματοποιούν κλήσεις, να στέλνουν μηνύματα κτλ.) αλλά ταυτόχρονα δίνουν στον χρήστη την δυνατότητα να πραγματοποιήσει πολλές και διαφορετικές ενέργειες που μέχρι πρότινος θα έκανε μόνο μέσω των υπολογιστών. Τα τελευταία χρόνια, τα κινητά τηλέφωνα έχουν αλλάξει ριζικά τον τρόπο που επικοινωνεί ο σύγχρονος άνθρωπος και έχουν ουσιαστικά καταφέρει να αντικαταστήσουν συσκευές και μέσα που χρησιμοποιούν οι άνθρωποι σε καθημερινή βάση.

Οι καινούριες τεχνολογίες και οι συνεχείς καινοτομίες στον κόσμο της τεχνολογίας έχουν οδηγήσει στη δημιουργία νέων έξυπνων κινητών συσκευών με διαφορετικά λειτουργικά συστήματα, ικανά να εκτελούν ταυτόχρονα πολλαπλές διεργασίες, ελαχιστοποιώντας τον χρόνο απόκρισης και αναμονής για τους χρήστες. Η πιο διάσημη κινητή πλατφόρμα είναι το λειτουργικό σύστημα Android. Το Android είναι μία πλατφόρμα για κινητά τηλέφωνα ανοιχτού κώδικα, που διαχειρίζεται η Open Handset Alliance (OHA). Η Open Handset Alliance (OHA) είναι μια κοινοπραξία 84 εταιριών που συστάθηκε για την ανάπτυξη ανοικτών προτύπων για τις κινητές συσκευές. Μερικές από τις πιο γνωστές εταιρίες που περιλαμβάνονται είναι οι ακόλουθες: η Google, η HTC, η Sony, η Dell, η Intel, η Motorola, η Qualcomm, η Texas Instruments, η Samsung Electronics, η LG Electronics, η T-Mobile,η Nvidia κτλ. Η OHA ιδρύθηκε στις 5 Νοεμβρίου του 2007 έχοντας 34 ιδρυτικά μέλη και επικεφαλής την Google, συμπεριλαμβανομένων διαφόρων κατασκευαστών κινητών συσκευών, προγραμματιστών εφαρμογών, φορέων κινητής τηλεφωνίας και κατασκευαστών τσιπ. Το Android, το λογισμικό ορόσημο της συμμαχίας (που αναπτύχθηκε για πρώτη φορά από την Google το 2007), βασίζεται σε άδεια ανοιχτού κώδικα και έχει εξελιχθεί ραγδαία τα τελευταία 10 χρόνια.[1]

Τα κινητά τηλέφωνα αποτελούν αναπόσπαστο κομμάτι της καθημερινότητάς μας και το γεγονός αυτό τα καθιστά ευάλωτα και χρήσιμα σαν εργαλεία για πραγματοποίηση κακόβουλων και εγκληματικών ενεργειών. Τα νέα χαρακτηριστικά των κινητών, όπως η πρόσβαση στο Διαδίκτυο, η χρήση της τοποθεσίας GPS ή της τεχνολογίας Bluetooth αποτελούν μερικούς μόνο από τους τρόπους που οι κυβερνοεγκληματίες εκμεταλλεύονται για να πραγματοποιήσουν κακόβουλες ή εγκληματικές ενέργειες. Οι κυβερνοεγκληματίες μπορούν εύκολα να εγκαταστήσουν ένα πρόγραμμα απομακρυσμένης διαχείρισης RAT (Remote Administration Tool) μεταμφιεσμένο σαν μία απλή εφαρμογή στο λειτουργικό σύστημα της κινητής συσκευής και να αποκτήσουν πρόσβαση σε ένα φάσμα λειτουργιών του κινητού αυτού, όπως για

[1] https://en.wikipedia.org/wiki/Open_Handset_Alliance

παράδειγμα της κάμερας, της οθόνης, του μικροφώνου της συσκευής κτλ. Αυτό πρακτικά σημαίνει ότι οι εγκληματίες μπορούν να έχουν απομακρυσμένη πρόσβαση στα κινητά τηλέφωνα των χρηστών - θυμάτων και μπορούν να προσπελάσουν τα δεδομένα των χρηστών για όσο χρονικό διάστημα επιθυμούν.

Τα ψηφιακά πειστήρια σε συσκευές Android αποτελούν μέρος της ψηφιακής εγκληματολογίας και παρουσιάζουν πολλές δυνατότητες αλλά ταυτόχρονα και πολλές προκλήσεις. Για να είναι επιτυχημένη μία εξέταση ψηφιακών πειστηρίων, απαιτούνται γνώσεις τόσο της πλατφόρμας του Android όσο και των εργαλείων ψηφιακών πειστηρίων που υπάρχουν διαθέσιμα για να μπορεί κάποιος να εξάγει και να αναλύσει τα δεδομένα ενός κινητού.

1.2 Ορισμοί και έννοιες της ψηφιακής εγκληματολογίας

Η ψηφιακή εγκληματολογία είναι η διαδικασία αποκάλυψης και ερμηνείας ηλεκτρονικών δεδομένων. Σκοπός της διαδικασίας αυτής, είναι η διαφύλαξη κάθε αποδεικτικού στοιχείου στην πιο πρωτότυπη μορφή του, ενώ διεξάγεται μια δομημένη έρευνα συλλέγοντας, προσδιορίζοντας και επικυρώνοντας τις ψηφιακές πληροφορίες με σκοπό την ανακατασκευή παρελθοντικών γεγονότων. Το πλαίσιο αυτό είναι συνήθως για τη χρήση των δεδομένων/αποδείξεων σε δικαστήριο, αν και η ψηφιακή εγκληματολογία μπορεί να χρησιμοποιηθεί σε πληθώρα άλλων περιπτώσεων.[2]

Η φύση των ψηφιακών πειστηρίων είναι τέτοια που θέτει ειδικές προκλήσεις για την αποδοχή τους κυρίως από τα δικαστήρια. Για να αντιμετωπιστούν αυτές οι προκλήσεις ακολουθούνται κατάλληλες και διεθνώς αναγνωρισμένες διαδικασίες που σχετίζονται με την ψηφιακή εγκληματολογία. Οι διαδικασίες αυτές αποτελούνται από τέσσερις βασικές φάσεις:

- την προετοιμασία (preparation),
- την απόκτηση (acquisition),
- την ανάλυση (analysis),
- την τελική ανάλυση των αποτελεσμάτων και την αναφορά (final output analysis and reporting).

Ας δούμε λίγο αναλυτικότερα όμως την κάθε φάση. Η φάση της προετοιμασίας περιλαμβάνει την αναζήτηση, την αναγνώριση και την συγκέντρωση των ψηφιακών πειστηρίων. Θεωρώ ότι η προετοιμασία αποτελεί το σημαντικότερο βήμα της ψηφιακής έρευνας γιατί καθορίζει το κατά πόσο τα στοιχεία θα μπορέσουν να γίνουν αποδεκτά στο δικαστήριο.

Η διαδικασία της απόκτησης βοηθά τον ερευνητή να καταστήσει τα ψηφιακά πειστήρια ορατά και να εξηγήσει την προέλευση και τη σημασία τους. Αυτή η διαδικασία πρέπει να τεκμηριώσει το περιεχόμενο και την κατάσταση των στοιχείων στο σύνολό τους.

Στη φάση της ανάλυσης, απομονώνονται οι ουσιαστικές και χρησιμοποιήσιμες πληροφορίες από τον ερευνητή και διαχωρίζονται από τις αχρείαστες. Έχοντας λάβει υπόψιν τον τεράστιο όγκο των πληροφοριών που μπορεί να είναι αποθηκευμένος στα μέσα αποθήκευσης που χρησιμοποιούν οι ψηφιακές συσκευές, μπορεί εύκολα να γίνει αντιληπτό ότι η συγκεκριμένη φάση της εξέτασης είναι το κρισιμότερο και δυσκολότερο σημείο μίας ψηφιακής έρευνας.

[2] https://www.techopedia.com/definition/27805/digital-forensics

Η τελευταία φάση της εγκληματολογικής εξέτασης ολοκληρώνεται με την σύνταξη της αναφοράς στην οποία περιγράφεται με λεπτομέρεια όλη η διαδικασία της εξέτασης. Είναι σημαντικό τα στοιχεία να παρουσιάζονται με σαφήνεια και η αναφορά να μπορεί να γίνει αντιληπτή και κατανοητή από άτομα που δεν έχουν ουδεμία σχέση με το αντικείμενο. Καλό θα είναι στην αναφορά να περιλαμβάνεται και οπτικό υλικό για ευκολότερη και ορθότερη απεικόνιση και παρουσίαση των αποτελεσμάτων.

1.3 Διάρθρωση της μελέτης

Στο δεύτερο κεφάλαιο που ακολουθεί, γίνεται αναφορά στο λειτουργικό σύστημα του Android. Περιγράφεται αναλυτικά η αρχιτεκτονική των έξι βασικών επιπέδων του Android, γίνεται μία σύντομη ιστορική αναδρομή των διαφορετικών εκδόσεων από την ημερομηνία κυκλοφορίας του μέχρι και σήμερα, και παρουσιάζεται η δομή των αρχείων του συστήματος, καθώς επίσης και ο τρόπος με τον οποίο τα δεδομένα καταχωρούνται στις συσκευές με λειτουργικό σύστημα Android.

Στο τρίτο κεφάλαιο περιγράφεται η έννοια των ψηφιακών πειστηρίων. Περιγράφονται τα μέρη στα οποία μπορούμε να αποθηκεύσουμε τα δεδομένα σε συσκευές Android, αναφέρονται και αναλύονται οι τρόποι απόκτησης των δεδομένων από τα διάφορα μέρη της συσκευής και τέλος γίνεται αναφορά στις πληροφορίες που μπορούν να ανακτηθούν μέσω των ψηφιακών πειστηρίων.

Στο τέταρτο κεφάλαιο γίνεται αναλυτική περιγραφή των βημάτων που ακολουθούνται για μία ολοκληρωμένη ψηφιακή εξέταση.

Στο πέμπτο κεφάλαιο, γίνεται καταγραφή και αναλυτική παρουσίαση διάφορων εμπορικών εργαλείων και εργαλείων ανοιχτού κώδικα και αντιστοίχισής τους με τις μεθόδους απόκτησης σε συσκευές Android. Επίσης, για κάθε εργαλείο περιγράφεται το είδος των πληροφοριών που μπορούν να ανακτηθούν με την χρήση του.

Στο έκτο κεφάλαιο γίνεται αναλυτική περιγραφή της ψηφιακής εγκληματολογίας μνήμης σε συσκευές Android και περιγράφονται οι διαφορές μεταξύ «ζωντανής» και «νεκρής» ψηφιακής εγκληματολογίας.

Στο έβδομο κεφάλαιο πραγματοποιείται ανάλυση της μνήμης από μία συσκευή Android η οποία αποκτήθηκε με την χρήση του εργαλείου LiME.

Στο όγδοο κεφάλαιο περιλαμβάνεται η σύνοψη καθώς και τα συμπεράσματα της εργασίας.

Στο ένατο και τελευταίο κεφαλαίο γίνεται εκτενής αναφορά στην βιβλιογραφία και τις πηγές που χρησιμοποιηθήκαν.

This page intentionally left blank

2. Το λειτουργικό σύστημα Android

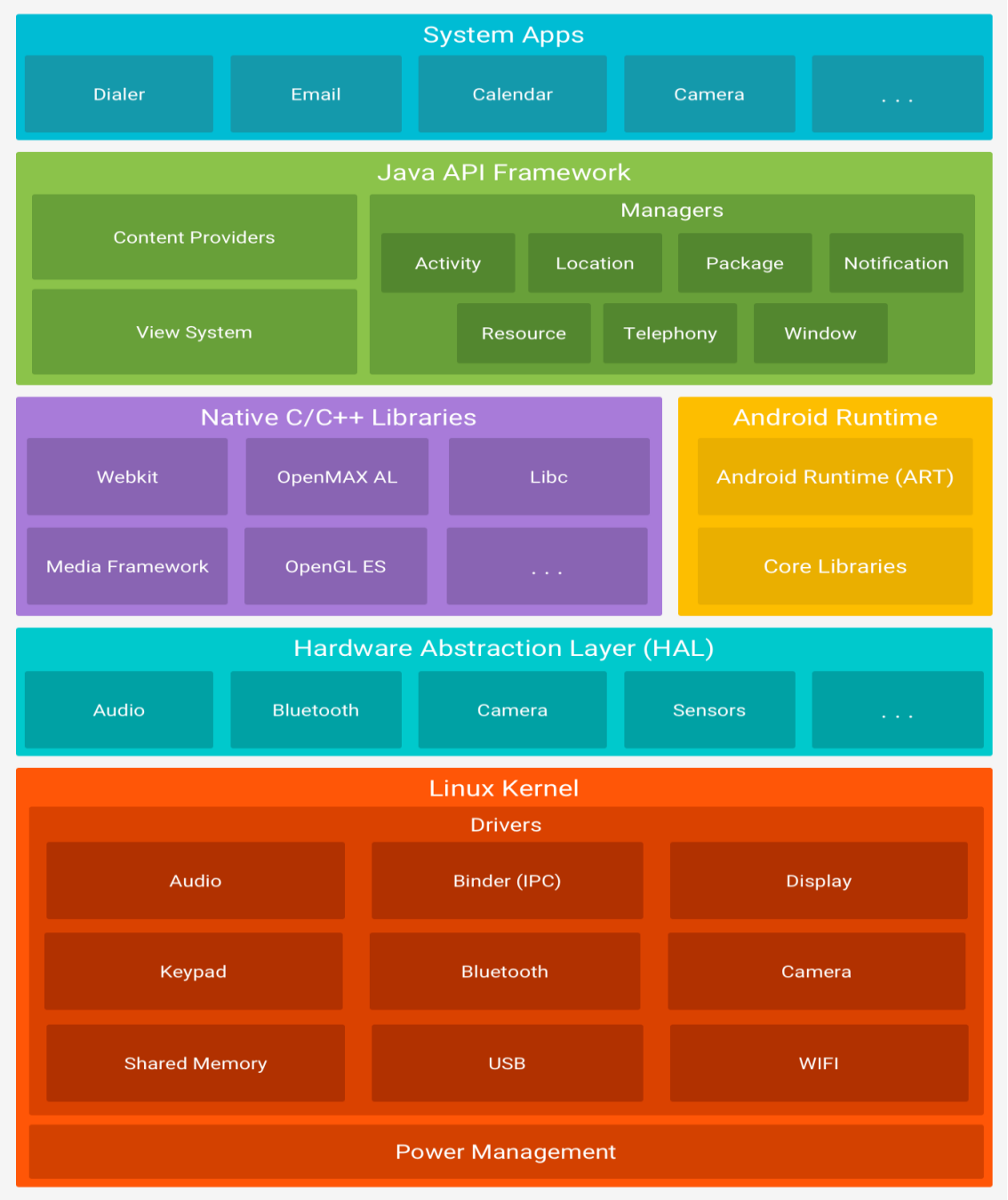
2.1 Η αρχιτεκτονική του λειτουργικού συστήματος Android

Το Android αποτελεί ένα λειτουργικό σύστημα το οποίο χρησιμοποιείται για διάφορες φορητές συσκευές. Ο πυρήνας του Android είναι βασισμένος στον πυρήνα του Linux, ο οποίος δημιουργήθηκε αρχικά από τον Φινλανδό φοιτητή πληροφορικής Λίνους Τόρβαλντς το 1991 και αποτελεί μία πρωτότυπη υλοποίηση πυρήνα λειτουργικού συστήματος. Αν και δεν χρησιμοποιεί κώδικα του UNIX, στην πραγματικότητα μπορεί να θεωρηθεί ένα παρεμφερές σύστημα (ο ακριβής όρος είναι Unix-like) ή μία πιο ελεύθερη υλοποίησή του, αφού διαθέτει τις περισσότερες εντολές του και την ίδια σχεδόν δομή αρχείων, ενώ η φιλοσοφία της σχεδίασής του πλησιάζει περισσότερο το UNIX από οποιοδήποτε άλλο λειτουργικό σύστημα.[3] Κληρονομώντας λοιπόν όλα τα χαρακτηριστικά ασφαλείας του Linux και τις τεχνικές διαχείρισης μνήμης αλλά και επεξεργαστή που αυτό διαθέτει, το Android αποτελεί ένα αρκετά αξιόπιστο και ταυτόχρονα ασφαλές λειτουργικό σύστημα. Η αρχιτεκτονική του Android αποτελείται από έξι βασικά επίπεδα, που είναι τα ακόλουθα:

- Ο πυρήνας του Linux (The Linux Kernel)
- Το επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer, γνωστό και ως «HAL»)
- Το επίπεδο εκτέλεσης Android (Android Runtime)
- Το επίπεδο βιβλιοθηκών C/C++ (Native C/C++ Libraries)
- Το επίπεδο διεπαφής προγραμματισμού εφαρμογών Java (Java API Framework)
- Το επίπεδο εφαρμογών συστήματος (System Apps)

Για να καταλάβει κανείς καλύτερα την αρχιτεκτονική του Android, ακολουθεί μία γραφική απεικόνιση των επιπέδων της αρχιτεκτονικής καθώς και αναλυτική περιγραφή των επιπέδων που αναφέρθηκαν, ξεκινώντας από το χαμηλότερο προς το υψηλότερο επίπεδο(βλ. Εικόνα 1).

[3] https://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%AE%CE%BD%CE%B1%CF%82_Linux



Εικόνα 1. Η αρχιτεκτονική του λειτουργικού συστήματος Android.
(Πηγή: https://developer.android.com/guide/platform/images/android-stack_2x.png)

2.1.1 Ο πυρήνας του Linux (The Linux Kernel)

Ο κορμός της πλατφόρμας του Android είναι βασισμένος στον πυρήνα του Linux. Για παράδειγμα, το επίπεδο εκτέλεσης του Android βασίζεται στον πυρήνα του Linux για υποκείμενες λειτουργίες του, όπως η διαχείριση της επεξεργαστικής ισχύος και η διαχείριση των χαμηλών επιπέδων μνήμης του συστήματος. Η χρήση του πυρήνα του Linux επιτρέπει στο Android να αξιοποιεί τα βασικά χαρακτηριστικά ασφαλείας του Linux και παράλληλα επιτρέπει στους κατασκευαστές των συσκευών να αναπτύσσουν προγράμματα οδήγησης υλικού για έναν οικείο πυρήνα λειτουργικού συστήματος.

2.1.2 Το επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer)

Το επίπεδο αφαίρεσης υλικού (HAL) παρέχει τυποποιημένες διεπαφές που εκθέτουν τις δυνατότητες υλικού των συσκευών στο υψηλότερο επίπεδο διεπαφής προγραμματισμού μέσω εφαρμογών Java. Το επίπεδο αφαίρεσης υλικού αποτελείται από πολλαπλές ενότητες βιβλιοθήκης, κάθε μία από τις οποίες ενσωματώνει μία διεπαφή για έναν συγκεκριμένο τύπο υλικού, όπως η κάμερα ή η μονάδα bluetooth της συσκευής. Όταν ένα API(επίπεδο διεπαφής προγραμματισμού) πλαισίου κάνει μία κλήση για να αποκτήσει πρόσβαση στο υλικό της συσκευής, το σύστημα του Android φορτώνει τη λειτουργική μονάδα της βιβλιοθήκης για το στοιχείο υλικού που ζητήθηκε.

2.1.3 Το επίπεδο εκτέλεσης Android (Android Runtime)

Οι συσκευές που χρησιμοποιούν έκδοση Android 5.0 (επίπεδο 21 API) και πάνω, επιτρέπουν σε κάθε εφαρμογή που εκτελείται να τρέχει με δική της διεργασία και έχοντας ξεχωριστή παρουσία στο επίπεδο εκτέλεσης του Android. Το επίπεδο εκτέλεσης του Android είναι γραμμένο με τέτοιο τρόπο ώστε να επιτρέπει την εκτέλεση πολλαπλών εικονικών μηχανών σε συσκευές που διαθέτουν χαμηλή μνήμη εκτελώντας αρχεία τύπου DEX (Dalvik Executable), μια μορφή bytecode ειδικά σχεδιασμένη για το Android η οποία έχει βελτιστοποιηθεί έχοντας σαν σκοπό να αφήνει το μικρότερο δυνατό αποτύπωμα στη μνήμη. Εργαλεία που έχουν δημιουργηθεί για να εξυπηρετηθεί η ανάγκη αυτή, όπως το Jack, έχουν την δυνατότητα να μεταγλωττίσουν πηγές του κώδικα Java σε αρχεία τύπου DEX, τα οποία μπορούν να τρέξουν στην πλατφόρμα του Android.

Μερικά από τα κύρια χαρακτηριστικά του επιπέδου εκτέλεσης Android περιλαμβάνουν τα εξής:

- Τους τρόπους μεταγλώττισης ahead-of-time (AOT) και just-in-time (JIT)
- Βελτιστοποιημένη συλλογή άχρηστων αρχείων (Optimized garbage collection, συντομογραφία GC)
- Καλύτερη υποστήριξη εντοπισμού σφαλμάτων, συμπεριλαμβανομένου ενός ειδικού προφίλ δειγματοληψίας, λεπτομερείς διαγνωστικές εξαιρέσεις και αναφορές σφαλμάτων καθώς και την δυνατότητα καθορισμού σημείων παρακολούθησης για την παρακολούθηση συγκεκριμένων πεδίων.

Το Android περιλαμβάνει επίσης ένα σύνολο βασικών βιβλιοθηκών χρόνου εκτέλεσης που παρέχουν τις περισσότερες λειτουργίες της γλώσσας προγραμματισμού Java,

συμπεριλαμβανομένων ορισμένων λειτουργιών της γλώσσας Java 8, οι οποίες χρησιμοποιούνται από το πλαίσιο επιπέδου διεπαφής προγραμματισμού Java.

2.1.4 Το επίπεδο βιβλιοθηκών C/C++ (Native C/C++ Libraries)

Πολλά βασικά στοιχεία και υπηρεσίες του συστήματος Android, όπως το επίπεδο εκτέλεσης Android και το επίπεδο εκτέλεσης υλικού, είναι γραμμένα σε κώδικα ο οποίος για να τρέξει και να εκτελεστεί επιτυχώς χρειάζεται βιβλιοθήκες οι οποίες είναι γραμμένες στις γλώσσες προγραμματισμού C και C++. Η πλατφόρμα του Android παρέχει το επίπεδο διεπαφής προγραμματισμού πλαισίου Java, για να αποκαλύψει την λειτουργικότητα ορισμένων βιβλιοθηκών στις εφαρμογές. Για να γίνει καλύτερα κατανοητό αυτό, ας δούμε το παρακάτω παράδειγμα: υπάρχει η δυνατότητα κάποιος προγραμματιστής να έχει πρόσβαση στο OpenGL ES μέσω του Java OpenGL API του Android για να προσθέσει υποστήριξη για την σχεδίαση και τον χειρισμό γραφικών απεικόνισης 2D και 3D στην εφαρμογή του. Αν κάποιος αναπτύσσει μια εφαρμογή που απαιτεί κώδικα C ή C++, του δίνετε η δυνατότητα να χρησιμοποιήσει το Android NDK (Native Development Kit) για να αποκτήσει πρόσβαση σε ορισμένες από αυτές τις βιβλιοθήκες απευθείας πάνω από τον κώδικα.

2.1.5 Το επίπεδο διεπαφής προγραμματισμού εφαρμογών Java (Java API Framework)

Το σύνολο των χαρακτηριστικών του λειτουργικού συστήματος Android είναι διαθέσιμο στους χρήστες μέσω των API (επίπεδα διεπαφής προγραμματισμού) τα οποία είναι γραμμένα σε γλώσσα Java. Αυτά τα API αποτελούν τις δομικές μονάδες που απαιτούνται για την δημιουργία των Android εφαρμογών απλοποιώντας την επαναχρησιμοποίηση των κύριων μερών και υπηρεσιών του συστήματος, τα οποία περιλαμβάνουν τα εξής:

- Σύστημα προβολής (View System): επεκτάσιμο και παραμετροποιήσιμο που μπορεί να χρησιμοποιηθεί για να δημιουργηθεί ένα περιβάλλον διεπαφής μίας εφαρμογής, συμπεριλαμβανομένων των λιστών, των πλεγμάτων, των πλαισίων κειμένου, των κουμπιών, ακόμη και ενός ενσωματωμένου προγράμματος περιήγησης ιστού.
- Διαχειριστής πόρων (Resource Manager): παρέχει πρόσβαση σε πόρους που δεν χρησιμοποιούν κώδικα, όπως γραφικά και αρχεία διαμόρφωσης.
- Διαχειριστής ειδοποιήσεων (Notification Manager): επιτρέπει σε όλες τις εφαρμογές να εμφανίζουν προσαρμοσμένες ειδοποιήσεις στη γραμμή κατάστασης.
- Διαχειριστής δραστηριοτήτων (Activity Manager): διαχειρίζεται τον κύκλο ζωής των εφαρμογών και παρέχει μία κοινή στοίβα πλοήγησης.
- Διαχειριστής τοποθεσίας (Location Manager): παρέχει πρόσβαση στις υπηρεσίες τοποθεσίας του συστήματος. Ο διαχειριστής τοποθεσίας επιτρέπει στις εφαρμογές να λαμβάνουν περιοδικές ενημερώσεις της γεωγραφικής θέσης της συσκευής ή ακόμα και να πραγματοποιούν συγκεκριμένη ενέργεια όταν βρίσκονται σε γνωστή γεωγραφική θέση (π.χ. αυτόματη ενεργοποίηση Wi-Fi).

- Διαχειριστής πακέτων (Package Manager): ελέγχει το χρόνο ζωής των εφαρμογών και φροντίζει για την διατήρηση μίας στοίβας, η οποία δίνει την δυνατότητα στον χρήστη να πλοηγείτε σε προηγούμενες οθόνες.
- Παροχείς περιεχομένου (Content Providers): επιτρέπουν σε εφαρμογές να έχουν πρόσβαση σε δεδομένα από άλλες εφαρμογές ή να μπορούν να μοιράζονται τα δικά τους δεδομένα.

Οι προγραμματιστές έχουν πλήρη πρόσβαση στα ίδια πλαίσια API που χρησιμοποιούν οι εφαρμογές του συστήματος Android.

2.1.6 Το επίπεδο εφαρμογών συστήματος (System Apps)

Το λειτουργικό σύστημα Android διαθέτει ένα σύνολο βασικών εφαρμογών για λειτουργίες όπως η ανταλλαγή μηνυμάτων ηλεκτρονικού ταχυδρομείου, η προβολή και ανταλλαγή μηνυμάτων SMS, η διαχείριση ημερολογίου, η περιήγηση στο Διαδίκτυο, η προβολή και οργάνωση των επαφών και πολλά άλλα. Οι προεγκατεστημένες εφαρμογές που περιλαμβάνονται στην πλατφόρμα και εξυπηρετούν τις συγκεκριμένες δυνατότητες μπορούν να αλλάξουν, αν ο χρήστης εξυπηρετείται καλύτερα από εφαρμογές τρίτων και δεν είναι υποχρεωτική η χρήση τους. Αυτό όμως που δεν είναι εφικτό στις περισσότερες περιπτώσεις, είναι η οριστική απεγκατάσταση των προεγκατεστημένων εφαρμογών, παρά μόνο η δυνητική απενεργοποίηση τους. Οι εφαρμογές του συστήματος λειτουργούν ως εφαρμογές που εξυπηρετούν τις ανάγκες των χρηστών αλλά ταυτόχρονα παρέχουν και βασικές δυνατότητες στις οποίες μπορούν να έχουν πρόσβαση οι προγραμματιστές μέσω των δικών τους εφαρμογών.[4]

2.2 Οι εκδόσεις του λειτουργικού συστήματος Android

Η πρώτη παρουσίαση της πλατφόρμας του Android έγινε επίσημα στις 5 Νοεμβρίου του 2007. Σχεδόν ένα χρόνο αργότερα, στις 23 Σεπτεμβρίου του 2008 κυκλοφόρησε στην αγορά η πρώτη συσκευή με λειτουργικό σύστημα Android (HTC Dream). Ακολουθεί ένας αναλυτικός πίνακας που περιλαμβάνει το ιστορικό όλων των εκδόσεων Android από το 2008 μέχρι και σήμερα.[5]

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
Κανένα	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουαρίου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3

[4] https://developer.android.com/guide/platform/

[5] https://en.wikipedia.org/wiki/Android_version_history

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
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 Φεβρουάριου 2011	11–13
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
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.1	22 Αυγούστου 2016	24-25
Oreo	8.0-8.1	21 Αυγούστου 2017 – 5 Δεκεμβρίου 2017	26-27
Pie	9.0	6 Αυγούστου 2018	28

Παρά το νεαρό της ηλικίας του, το Android έχει καταφέρει σύμφωνα και με επίσημα στοιχεία τα οποία ανακοινώθηκαν από την Google τον Μάιο του 2017, να αριθμεί 2 δισεκατομμύρια ενεργές συσκευές παγκοσμίως. Αυτό αποτυπώνεται και στον παρακάτω πίνακα (Εικόνα 2) που μας δείχνει το παγκόσμιο μερίδιο αγοράς των λειτουργικών συστημάτων για φορητές συσκευές. Παρατηρείται λοιπόν η ξεκάθαρη κυριαρχία του Android απέναντι στο αντίπαλο δέος (iOS) με ποσοστό 85% έναντι του μόλις 14,7% που λαμβάνει το iOS.

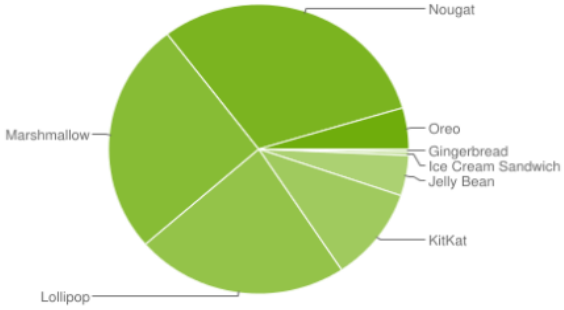
Period	Android	iOS	Windows Phone	Others
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%
2016Q4	81.4%	18.2%	0.2%	0.2%
2017Q1	85.0%	14.7%	0.1%	0.1%

Source: IDC, May 2017

Εικόνα 2. Android market share, May 2017.
(Πηγή: https://www.idc.com/promo/smartphone-market-share/os)

Η κατανομή των εκδόσεων Android σύμφωνα με επίσημα στοιχεία που ανακοίνωσε η Google τον Απρίλιο του 2018 αποτυπώνεται στην εικόνα που ακολουθεί (Εικόνα 3).

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.5%
5.0	Lollipop	21	4.9%
5.1		22	18.0%
6.0	Marshmallow	23	26.0%
7.0	Nougat	24	23.0%
7.1		25	7.8%
8.0	Oreo	26	4.1%
8.1		27	0.5%



Data collected during a 7-day period ending on April 16, 2018.
Any versions with less than 0.1% distribution are not shown.

Εικόνα 3. Η κατανομή των εκδόσεων του Android μέχρι τις 16 Απριλίου 2018.
(Πηγή: https://developer.android.com/about/dashboards/index.html)

Αυτό που προκαλεί τεράστια εντύπωση κοιτώντας τα παραπάνω στατιστικά, είναι το γεγονός ότι υπάρχουν ενεργές συσκευές οι οποίες τρέχουν ακόμα το λειτουργικό σύστημα «Gingerbread» το οποίο πρωτοκυκλοφόρησε στα τέλη του 2010.

Η αυξημένη δημοτικότητα του Android τα τελευταία χρόνια, έχει επιφέρει και σημαντική αύξηση στον αριθμό των ευπαθειών που επηρεάζουν το εν λόγω λειτουργικό σύστημα, έχοντας συνολικά 1926 καταγεγραμμένες ευπάθειες, γεγονός που το κατατάσσει τρίτο στην σχετική λίστα του cvedetails, μετά τον πυρήνα του Linux (Linux Kernel) ο οποίος έχει 2145 ευπάθειες και το λειτουργικό σύστημα Mac OS με 2084 ευπάθειες.[6]

Από τις συνολικά 1926 ευπάθειες που παρουσιάζει το Android, οι 1365 έχουν αναγνωριστεί τα τελευταία δύο χρόνια (523 για το 2016 και 842 για το 2017) καθιστώντας το Android το πιο ευάλωτο λειτουργικό σύστημα των τελευταίων δύο χρόνων και τοποθετώντας το πρώτο στην σχετική λίστα με τα περισσότερα CVE's για τις τελευταίες δύο χρονιές.[7][8]

2.3 Τα αρχεία συστήματος στο λειτουργικό σύστημα Android

Το λειτουργικό σύστημα Android χρησιμοποιεί πολλά διαφορετικά διαμερίσματα (π.χ. εκκίνηση, σύστημα, ανάκτηση, δεδομένα κ.λπ.) για την οργάνωση των αρχείων και των φακέλων στη συσκευή, ακριβώς όπως και το λειτουργικό σύστημα των Windows. Κάθε ένα από τα διαμερίσματα έχει την δική του λειτουργικότητα.

Υπάρχουν έξι κύρια διαμερίσματα τα οποία συναντάμε σε συσκευές Android. Βέβαια, ενδέχεται να υπάρχουν και κάποια άλλα διαμερίσματα διαθέσιμα καθώς παρουσιάζονται διαφορές από μοντέλο σε μοντέλο. Τα βασικότερα διαμερίσματα λοιπόν είναι τα εξής:

- /boot: αποτελεί το διαμέρισμα εκκίνησης των συσκευών Android, όπως καταλαβαίνει κάποιος και από την ονομασία του. Περιλαμβάνει τον πυρήνα του Android και το ramdisk. Αποτελεί απαραίτητο διαμέρισμα για την εκκίνηση της συσκευής. Η διαγραφή αυτού του διαμερίσματος από την λειτουργία ανάκτησης (recovery mode) πρέπει να γίνει εφόσον είναι απολύτως απαραίτητο και μόλις γίνει, η συσκευή δεν πρέπει να επανεκκινηθεί πριν την εγκατάσταση ενός νέου διαμερίσματος, το οποίο μπορεί να γίνει με την εγκατάσταση μιας ROM που περιλαμβάνει ένα διαμέρισμα / boot.

- /system: το διαμέρισμα αυτό περιέχει στην ουσία ολόκληρο το λειτουργικό σύστημα, εκτός από τον πυρήνα του λειτουργικού και το ramdisk. Περιλαμβάνει λοιπόν την διεπαφή χρήστη του Android καθώς και όλες τις εφαρμογές του συστήματος που υπάρχουν προεγκατεστημένες στη συσκευή. Η διαγραφή του συγκεκριμένου διαμερίσματος θα αφαιρέσει το Android από τη συσκευή χωρίς να το καταστήσει μη εκκινήσιμο και θα υπάρχει η δυνατότητα να τοποθετηθεί το τηλέφωνο σε λειτουργία ανάκτησης ή εκκίνησης για να εγκατασταθεί μία καινούρια ROM.

[6] https://www.cvedetails.com/top-50-products.php?year=0, πρόσβαση την 01/10/2018

[7] https://www.cvedetails.com/top-50-products.php?year=2016, πρόσβαση την 07/03/2018

[8] https://www.cvedetails.com/top-50-products.php?year=2017, πρόσβαση την 07/03/2018

- /recovery: το διαμέρισμα ανάκτησης μπορεί να θεωρηθεί ως ένα εναλλακτικό διαμέρισμα εκκίνησης που επιτρέπει στον χρήστη να εκκινήσει την συσκευή σε μια κονσόλα ανάκτησης, με απώτερο σκοπό την εκτέλεση προηγμένων λειτουργιών ανάκτησης και συντήρησης στη συσκευή.

- /data: αποκαλείται και userdata και είναι το διαμέρισμα δεδομένων που περιέχει τα δεδομένα του χρήστη. Σε αυτόν το χώρο αποθηκεύονται οι επαφές, τα μηνύματα, οι ρυθμίσεις και οι εφαρμογές που έχουν εγκατασταθεί. Η διαγραφή του συγκεκριμένου διαμερίσματος ουσιαστικά πραγματοποιεί μία επαναφορά εργοστασιακών ρυθμίσεων στη συσκευή, επαναφέροντάς το στην αρχική του κατάσταση ή στην κατάσταση που ήταν μετά από την τελευταία επίσημη ή προσαρμοσμένη εγκατάσταση ROM. Όταν πραγματοποιείται ολική διαγραφή των δεδομένων ή εργοστασιακή επαναφορά της συσκευής από την λειτουργία ανάκτησης, αυτό είναι το διαμέρισμα που επηρεάζεται.

- /cache: αποτελεί το διαμέρισμα στο οποίο αποθηκεύονται τα δεδομένα που γίνονται προσβάσιμα πιο τακτικά καθώς και μέρη των εφαρμογών. Η διαγραφή της προσωρινής μνήμης δεν επηρεάζει τα προσωπικά δεδομένα του χρήστη αλλά απλώς απαλείφει τα υπάρχοντα δεδομένα που σχετίζονται με τις εφαρμογές, τα οποία ξαναχτίζονται αυτόματα καθώς συνεχίζεται η χρήση της συσκευής.

- /misc: αποτελεί το διαμέρισμα που περιέχει διάφορες ρυθμίσεις του συστήματος με τη μορφή on/off διακοπτών. Αυτές οι ρυθμίσεις ενδέχεται να περιλαμβάνουν CID πληροφορίες (Carrier ή Region ID), ρυθμίσεις για το USB καθώς και ορισμένες ρυθμίσεις υλικού κτλ. Είναι ένα κρίσιμο διαμέρισμα και αν τυχόν είναι κατεστραμμένο ή απουσιάζει, πολλές από τις λειτουργίες της συσκευής δεν θα μπορούν να λειτουργήσουν κανονικά.

Εκτός από τα έξι διαμερίσματα που αναφέρθηκαν και αφορούν το σύστημα, υπάρχουν και δύο βασικά διαμερίσματα που σχετίζονται με την κάρτα μνήμης, τα οποία είναι:

- /sdcard: το συγκεκριμένο διαμέρισμα δεν περιλαμβάνεται στην εσωτερική μνήμη της συσκευής αλλά στην κάρτα SD (εφόσον υπάρχει). Από άποψη χρήσης, αυτός είναι ο χώρος αποθήκευσης που χρησιμοποιείται από τον χρήστη ανάλογα με τις ανάγκες του, είτε για να αποθηκεύσει αρχεία μουσικής, έγγραφα, εφαρμογές κτλ. Η διαγραφή του διαμερίσματος είναι απόλυτα ασφαλής, αρκεί να έχουν δημιουργηθεί από τον χρήστη τα κατάλληλα αντίγραφα ασφαλείας των δεδομένων που χρειάζεται.

- /sd-ext: το συγκεκριμένο δεν αποτελεί ένα τυπικό διαμέρισμα Android, αλλά έχει γίνει δημοφιλές σε όσους έχουν εγκαταστήσει μία προσαρμοσμένη έκδοση ROM. Στην ουσία αποτελεί ένα επιπλέον διαμέρισμα στην κάρτα SD που λειτουργεί ως διαμέρισμα δεδομένων όταν χρησιμοποιείται με συγκεκριμένες μονάδες ROM που έχουν ενεργοποιημένες ειδικές λειτουργίες, οι οποίες ονομάζονται APP2SD + ή data2ext. Είναι ιδιαίτερα χρήσιμο σε συσκευές που διαθέτουν μικρή εσωτερική μνήμη η οποία κατανέμεται στο διαμέρισμα δεδομένων. Συνεπώς, οι χρήστες που

επιθυμούν να εγκαταστήσουν περισσότερα προγράμματα από αυτά που μπορεί να «σηκώσει» η εσωτερική μνήμη της συσκευής τους, μπορούν να δημιουργήσουν αυτό το διαμέρισμα και να το χρησιμοποιήσουν με μία προσαρμοσμένη έκδοση ROM που υποστηρίζει τη συγκεκριμένη δυνατότητα, με σκοπό να αποκτήσουν πρόσθετο αποθηκευτικό χώρο για την εγκατάσταση εφαρμογών. Η διαγραφή του συγκεκριμένου διαμερίσματος θα είχε σαν αποτέλεσμα για τον χρήστη αντίστοιχο της διαγραφής του διαμερίσματος δεδομένων. Θα έχανε δηλαδή τις επαφές, τα μηνύματα SMS, τις εφαρμογές και τις ρυθμίσεις.[9][10]

2.4 Η καταχώρηση δεδομένων στο λειτουργικό σύστημα Android

Η καταχώρηση των δεδομένων σε συσκευές Android γίνεται σε έξι διαφορετικά μέρη. Όταν πραγματοποιείται μία έρευνα ψηφιακών πειστηρίων, ο έλεγχος που χρειάζεται να γίνει είναι σε πέντε από τα έξι μέρη που πραγματοποιείται η καταχώρηση δεδομένων. Αυτά τα μέρη είναι τα ακόλουθα:

- η εσωτερική μνήμη,
- η εξωτερική μνήμη,
- η βάση δεδομένων SQL lite,
- το δίκτυο,
- η βιβλιοθήκη μέσων sharedpreferences.

Αρκετά από τα δεδομένα των συσκευών Android αποθηκεύονται στην εσωτερική μνήμη της συσκευής και συγκεκριμένα στον φάκελο /data/data. Αυτός ο φάκελος είναι προσβάσιμος μόνο αν ο συνδεδεμένος χρήστης έχει δικαιώματα διαχειριστή.

[9] https://www.addictivetips.com/mobile/android-partitions-explained-boot-system-recovery-data-cache-misc/

[10] http://www.uobabylon.edu.iq/eprints/publication_4_13681_1356.pdf, σελίδες 2-16

3. Ψηφιακά πειστήρια κινητών συσκευών

3.1 Εισαγωγή

Χρησιμοποιώντας τον όρο της ψηφιακής εγκληματολογίας κινητών συσκευών (mobile forensics), εννοούμε την επιστήμη ανάκτησης στοιχείων και δεδομένων από τα κινητά τηλέφωνα, χρησιμοποιώντας συγκεκριμένες τεχνικές και τρόπους ανάλυσης των ανακτηθέντων στοιχείων εφαρμόζοντας κοινώς αποδεκτές μεθοδολογίες.[11]

Η συνεχής εξέλιξη των κινητών συσκευών παρουσιάζει αρκετές δυσκολίες στον κλάδο της ψηφιακής εγκληματολογίας καθώς τα εργαλεία και οι μέθοδοι διαφοροποιούνται, ανάλογα με την έκδοση και την μάρκα της εκάστοτε υπό εξέταση συσκευής.

3.2 Αποθήκευση δεδομένων στο λειτουργικό σύστημα Android

Τα δεδομένα σε μία κινητή συσκευή μπορούν να αποθηκευτούν σε διαφορετικά μέρη ανάλογα με το σκοπό που εξυπηρετούν. Η αποθήκευση χωρίζεται σε δύο βασικές κατηγορίες, την εσωτερική και την εξωτερική. Μία μεγάλη διαφορά που παρουσιάζει η μία κατηγορία από την άλλη είναι ότι στην εσωτερική αποθήκευση, η θέση αποθήκευσης των δεδομένων είναι προκαθορισμένη. Αντίθετα, στην εξωτερική αποθήκευση ο χρήστης αποθηκεύει τα δεδομένα σε οποιαδήποτε θέση εκείνος επιθυμεί. Πιο συγκεκριμένα, τα εσωτερικά δεδομένα όλων των εφαρμογών που υπάρχουν εγκατεστημένες στην συσκευή (εφαρμογές του συστήματος ή εφαρμογές που έχουν εγκατασταθεί από τον ίδιο τον χρήστη) αποθηκεύονται αυτόματα στον υποκατάλογο ../data/data.

Το Android δίνει την ευελιξία στους προγραμματιστές να εφαρμόσουν ορισμένες επιλογές σχετικά με την αποθήκευση των δεδομένων στη συσκευή. Η επιλογή που μπορεί να εφαρμοστεί εξαρτάται από τον χρήστη αλλά και από τον τύπο της εκάστοτε συσκευής. Γενικά, τα δεδομένα μπορούν να αποθηκεύονται στις παρακάτω θέσεις:

- Κάρτα SIM: Η κάρτα SIM (Subscriber Identify Module), είναι μία έξυπνη κάρτα η οποία πιστοποιεί την ταυτότητα του χρήστη και παρέχει ορισμένες βασικές λειτουργίες του κινητού. Για να αποκτήσει πρόσβαση ο χρήστης στην κάρτα SIM, απαιτείται ο αριθμός PIN (Personal Identification Number). Ο αριθμός PIN μπορεί να αποτελείται από τέσσερα έως οκτώ συνολικά ψηφία. Όταν η συσκευή ενεργοποιηθεί, ζητείται από τον χρήστη η εισαγωγή του αριθμού αυτού. Αν ο χρήστης αποτύχει να εισάγει το σωστό PIN μετά από τρεις προσπάθειες, η κάρτα SIM κλειδώνει προσωρινά και ο μοναδικός τρόπος για να καταστεί λειτουργική ξανά είναι να εισάγουμε τον οκταψήφιο κωδικό PUK (Personal Unblocking Key). Σε περίπτωση που ο χρήστης εισάγει λανθασμένα τον κωδικό PUK δέκα φορές, η κάρτα αχρηστεύεται και ο κάτοχος της πρέπει να ζητήσει από τον πάροχο του μία νέα κάρτα. Τέλος, υπάρχει και ο κωδικός ADM, τον οποίο γνωρίζει μόνο ο εκάστοτε πάροχος για την κάθε κάρτα και ο οποίος δίνει την

[11] https://www.techopedia.com/definition/27805/digital-forensics

δυνατότητα πλήρους πρόσβασης στα περιεχόμενα της κάρτας, καθώς επίσης και την δυνατότητα προσθήκης, τροποποίησης ή διαγραφής τους, ακόμα και απομακρυσμένα. Η δομή της κάρτας SIM οργανώνεται σε καταλόγους και αρχεία, όπου μέσα στους καταλόγους βρίσκονται αρχεία και φυλάσσονται διάφορες πληροφορίες. Σε κάθε ένα από αυτά τα αρχεία, υπάρχουν διαφορετικά δικαιώματα πρόσβασης (ανάγνωση, εγγραφή, τροποποίηση, διαγραφή κτλ.). Στην κάρτα SIM υπάρχουν περίπου εκατό αρχεία ανάλογα με το πρότυπο και κάποια επιπρόσθετα που διατηρεί ο εκάστοτε πάροχος. Ενδεικτικά, αξίζει να αναφερθεί ότι κάποια αρχεία έχοντας σαν βάση το πρότυπο, περιέχουν τις δυνατότητες του κινητού, τον σειριακό αριθμό της κάρτας, τον κατάλογο παρόχων και ονομάτων τους, το κατά προτίμηση δίκτυο και την προεπιλεγμένη γλώσσα, τον κατάλογο των επαφών, τα μηνύματα και τις κλήσεις *(Fabio Casadei, 2005).* Πολλά από τα αρχεία αυτά μπορούν να αξιοποιηθούν ως ψηφιακά πειστήρια, καθώς ο χρήστης δεν έχει άμεση πρόσβαση σε αυτά και συνεπώς δεν γνωρίζει την ύπαρξη τους προκειμένου να τα τροποποιήσει. Σε περίπτωση όμως που η εγκυρότητά τους αμφισβητείται, τότε υπάρχει η δυνατότητα να επαληθευτεί το γνήσιο τους με βάση τα αρχεία που διατηρεί ο εκάστοτε πάροχος. Συνοπτικά, τα αρχεία που συναντάμε σε μία κάρτα SIM είναι τα παρακάτω:

- **Αρχείο ICCID** (Integrated Circuit Card Identifier): το ICCID είναι ένα μοναδικός σειριακός αριθμός ο οποίος είναι τυπωμένος στο πλαστικό περίβλημα της κάρτας αλλά και προγραμματισμένος στο αντίστοιχο αρχείο της κάρτας SIM και υποδηλώνει διεθνώς σε ποια χώρα ανήκει η συγκεκριμένη κάρτα.

- **Αρχείο IMSI** (International Mobile Subscriber Identify): το IMSI είναι ένας μοναδικός αριθμός που αποτελείται από 15 ψηφία και χρησιμοποιείται για την αναγνώριση του συνδρομητή από το σύστημα. Αποτελεί στην ουσία το μυστικό κλειδί για την πιστοποίηση του εκάστοτε χρήστη. Μέσω του συγκεκριμένου αριθμού, μπορεί να ταυτοποιηθεί ο αριθμός τηλεφώνου της κάρτας SIM, ακόμα και αν η ισχύ της κάρτας έχει λήξει και δεν χρησιμοποιείται πλέον.[12]

- **Αρχείο Location Information και αρχείο Broadcast Control Channel**: στο αρχείο πληροφοριών συγκεκριμένης περιοχής (Location Information) βρίσκεται η προσωρινή ταυτότητα του κινητού, ένας αριθμός που παρουσιάζει ομοιότητες με τον αριθμό IMSI και χρησιμοποιείται για λόγους ασφαλείας. Αυτό αποσκοπεί ούτως ώστε η μόνιμη ταυτότητα του εκάστοτε χρήστη να μην εκπέμπεται συνεχώς στο δίκτυο όπως επίσης και ο αριθμός περιοχής που αντιστοιχεί στη χώρα, στο δίκτυο ή σε μία ευρύτερη περιοχή, η οποία μπορεί να περιλαμβάνει δεκάδες ή ακόμα και εκατοντάδες κυψέλες. Στο αρχείο πληροφοριών των καναλιών που ελέγχουν την εκπομπή, αποθηκεύεται η ταυτότητα του τρέχοντος καναλιού ελέγχου επικοινωνίας καθώς και των έξι γειτονικών καναλιών. Από το συνδυασμό των στοιχείων LAI (Location Area Identity) και BCCH (Broadcast

[12] https://www.scribd.com/document/322856651/IMSI-ICCID-IMEI

Control Channel) μπορεί να βρεθεί η τελευταία περιοχή στην οποία ήταν ενεργό το κινητό. Τα δεδομένα αυτά παραμένουν στην κάρτα SIM και μετά την απενεργοποίηση της συσκευής και ανανεώνονται κάθε φορά που ο χρήστης αλλάζει περιοχές. Συνεπώς, εκτός από τη χώρα προέλευσης της κάρτας, είναι δυνατή και η αναγνώριση της τοποθεσίας στην οποία χρησιμοποιήθηκε τελευταία φορά η κάρτα SIM (και κατ' επέκταση η συσκευή).

- **Αρχείο αποθήκευσης SMS**: με τις σύγχρονες κάρτες SIM, ο χρήστης μπορεί να αποθηκεύσει συνολικά 35 μηνύματα (όσες και οι θέσεις αποθήκευσης μηνυμάτων δηλαδή). Το γεγονός αυτό έχει σαν αποτέλεσμα, σε περίπτωση που ο κατασκευαστής του κινητού δεν έχει σαν προεπιλεγμένη μνήμη αποθήκευσης την εσωτερική μνήμη του κινητού, να μπορούν να βρεθούν όλα τα γραπτά μηνύματα. Όταν όμως έχει συμπληρωθεί ο διαθέσιμος χώρος, υπάρχει η δυνατότητα τα μηνύματα να αποθηκεύονται και στη μνήμη του κινητού. Η διαγραφή ενός μηνύματος SMS δεν συνεπάγεται και την ολοκληρωτική διαγραφή του. Επί της ουσίας, αυτό που κάνουμε διαγράφοντας ένα μήνυμα, είναι ότι χαρακτηρίζουμε μία συγκεκριμένη περιοχή μνήμης ως ελεύθερη για εκ νέου αποθήκευση.
- **Αρχείο AND** (Abbreviated Dialing Numbers): οι σύγχρονες κάρτες SIM διαθέτουν συνολικά 250 θέσεις που προορίζονται για το αρχείο όπου φυλάσσεται ο κατάλογος με τις επαφές. Στην περίπτωση του καταλόγου επαφών, όταν διαγράφουμε μία επαφή, στην ουσία γεμίζουμε την συγκεκριμένη περιοχή με δυαδικά «1», συνεπώς η ανάκτηση και σε αυτή την περίπτωση είναι ανέφικτη. Το ουσιαστικό συμπέρασμα που μπορεί ένας ψηφιακός αναλυτής να εξάγει, είναι η επιβεβαίωση της διαδικασίας της διαγραφής, εφόσον οι θέσεις μνήμης στο αρχείο αυτό καταλαμβάνονται με τη σειρά και σε περίπτωση διαγραφής τους, η θέση ουσιαστικά παραμένει κενή.

- Εσωτερική μνήμη συσκευής: ορισμένα από τα δεδομένα που αποθηκεύονται στην εσωτερική μνήμη του κινητού είναι τα ακόλουθα: ο αριθμός IMEI (International Mobile Equipment Identity), οι ρυθμίσεις της ώρας, οι ρυθμίσεις του ήχου, οι ρυθμίσεις της έντασης, τα μηνύματα SMS, το ημερολόγιο, το ιστορικό των κλήσεων, τα εκτελέσιμα αρχεία, οι εφαρμογές ή τα παιχνίδια, τα δεδομένα πολυμέσων όπως οι εικόνες, τα βίντεο, οι ηχογραφήσεις κτλ., εφόσον βέβαια κάτι τέτοιο έχει οριστεί από τον εκάστοτε κατασκευαστή. Επιπρόσθετα, σε περίπτωση που αναφερόμαστε σε έξυπνο κινητό που έχει την δυνατότητα σύνδεσης στο διαδίκτυο, αποθηκεύονται στοιχεία όπως το ιστορικό περιήγησης του χρήστη, οι αγαπημένες του ιστοσελίδες, τα ονόματα των Wi-Fi hotspots στα οποία έχει συνδεθεί κτλ. Στην συσκευή, είναι επίσης εφικτό να πραγματοποιηθεί η ανάκτηση των δεδομένων από προηγούμενες κάρτες SIM που είχαν χρησιμοποιηθεί με την συσκευή ή ακόμα και η ανάκτηση τυχόν διεγραμμένων αρχείων. Η εξέταση των δεδομένων μπορεί να γίνει από έναν ψηφιακό ερευνητή με τη χρήση ειδικών εργαλείων λογισμικού που αποτυπώνουν την εσωτερική μνήμη του τηλεφώνου σε μορφή κλώνου. Αναφορικά με το τεχνικό κομμάτι, η συγκεκριμένη διαδικασία είναι αρκετά πολύπλοκη. Αυτό οφείλεται στο γεγονός ότι τα δεδομένα είναι αδόμητα και πρέπει να μεταφρασθούν στο συγκεκριμένο σύστημα αρχείων. Υπάρχουν αρκετά τέτοια εργαλεία, τα οποία

μπορούν να εξάγουν δεδομένα ακόμη και αν η συσκευή είναι κλειστή, κλειδωμένη, κτλ. Παρά τα πλεονεκτήματα που παρουσιάζει η συγκεκριμένη μέθοδος «κλωνοποίησης» της εσωτερικής μνήμης, υπάρχουν και ορισμένα μειονεκτήματα. Το βασικότερο από αυτά, είναι ότι δεν υπάρχει τρόπος να ανιχνευθεί εάν η εσωτερική μνήμη έχει υποστεί οποιαδήποτε τροποποίηση, ενώ αξίζει να αναφερθεί ότι υπάρχουν και ορισμένες κινητές συσκευές που έχουν προγραμματισθεί με σκοπό την διαγραφή στοιχείων από τη μνήμη τους, εάν για οποιονδήποτε λόγο εισαχθεί μια διαφορετική κάρτα SIM από την τελευταία που χρησιμοποιήθηκε με την συσκευή.

- Εξωτερική κάρτα μνήμης SD(Secure Digital): Οι εξωτερικές κάρτες μνήμης παρέχουν στην συσκευή ένα επιπλέον χώρο αποθήκευσης, κάτι το οποίο είναι ιδιαιτέρως χρήσιμο αν αναλογιστούμε τις απαιτήσεις που έχουν οι χρήστες από τις τελευταίας τεχνολογίας κινητές συσκευές τους (πολλά GB δεδομένων με βίντεο, φωτογραφίες, αρχεία μουσικής κτλ.). Επίσης, οι κάρτες αυτές μπορούν να μεταφέρονται και στον υπολογιστή, ή σε περίπτωση που αυτό δεν είναι εφικτό, να συνδέεται ο υπολογιστής απευθείας με το κινητό για αποθήκευση των δεδομένων απευθείας από την συσκευή. Συνεπώς, η πιθανότητα σύνδεσης του υπολογιστή με την κινητή συσκευή, οδηγεί τον ψηφιακό ερευνητή στην αναγκαστική επέκταση της έρευνάς του στον σκληρό δίσκο αλλά και στην μνήμη του υπολογιστή, όπου μπορεί να εμπεριέχονται σημαντικά επιπλέον στοιχεία (όταν αυτό είναι εφικτό).
- Αποθηκευτικός χώρος στο σύννεφο (cloud): Όπως συμβαίνει και στους ηλεκτρονικούς υπολογιστές, έτσι και στις κινητές συσκευές ο χρήστης έχει την δυνατότητα να χρησιμοποιήσει αποθηκευτικό χώρο στο σύννεφο (cloud) ως επιπρόσθετο χώρο για να αποθηκεύσει τα δεδομένα του (αντίστοιχα δεδομένα όπως θα αποθήκευε και στην εξωτερική κάρτα μνήμης που αναφέραμε νωρίτερα). Όλες οι σύγχρονες κινητές συσκευές Android διαθέτουν την εφαρμογή Google Drive, η οποία δίνει την δυνατότητα στον χρήστη ανάλογα με το πλάνο που έχει επιλέξει, να διαθέτει από 15GB αποθηκευτικό χώρο στο σύννεφο μέχρι και 30TB με μηνιαία ή ετήσια χρέωση (με το ανάλογο αντίτιμο φυσικά). Ωστόσο, υπάρχει πληθώρα τέτοιων εφαρμογών που ο χρήστης μπορεί να εγκαταστήσει και να του προσφέρουν αντίστοιχα δυνατότητες cloud αποθήκευσης.
- Μνήμη RAM (Random Access Memory): σε αντίθεση με τα μέσα αποθήκευσης στα οποία έχει γίνει αναφορά μέχρι τώρα, η μνήμη RAM είναι η μόνη που αποθηκεύει <u>προσωρινά</u> στοιχεία και δεδομένα που αφορούν την συσκευή. Πιο συγκεκριμένα, μέσω της μνήμης RAM μπορεί ο ψηφιακός ερευνητής να ανακτήσει δεδομένα που σχετίζονται με την χρήση ορισμένων εφαρμογών, το ιστορικό ιδιωτικών συζητήσεων που συχνά δεν αποθηκεύονται σε παραδοσιακά μέρη αποθήκευσης όπως η εσωτερική μνήμη, κωδικούς χρήστη ή κλειδιά κρυπτογράφησης, τυχόν κακόβουλες διεργασίες που εκτελούνται στο παρασκήνιο, δικτυακές επικοινωνίες κτλ. Η ανάλυση της μνήμης RAM, είναι ένα σημαντικό εργαλείο που βοηθά τον ψηφιακό ερευνητή να εντοπίσει και να καταλάβει σε βάθος ένα κακόβουλο λογισμικό το οποίο τρέχει στην συσκευή σε πραγματικό χρόνο. Περισσότερα στοιχεία για την μνήμη RAM, θα αναλυθούν και στα επόμενα κεφάλαια.

3.3 Τρόποι εξαγωγής δεδομένων της εσωτερικής μνήμης

Η εξαγωγή δεδομένων που πραγματοποιείται σε κινητές συσκευές μπορεί να ταξινομηθεί ανάλογα με την μέθοδο που θα χρησιμοποιηθεί για την απόκτηση των δεδομένων. Ακολουθούν τα διαφορετικά είδη απόκτησης δεδομένων με μία σύντομη περιγραφή για το καθένα.

Χειροκίνητη απόκτηση (manual acquisition): ο εξεταστής χρησιμοποιεί το περιβάλλον χρήστη για να διερευνήσει το περιεχόμενο της μνήμης του τηλεφώνου. Επομένως, η συσκευή χρησιμοποιείται κανονικά, ενώ ο εξεταστής λαμβάνει φωτογραφίες από τα περιεχόμενα κάθε οθόνης. Αυτή η μέθοδος έχει το πλεονέκτημα ότι το λειτουργικό σύστημα δεν απαιτεί από τον ερευνητή την χρήση εξειδικευμένων εργαλείων ή εξοπλισμού με σκοπό την μετατροπή των ακατέργαστων δεδομένων σε πληροφορίες που να μπορούν να ερμηνευθούν από οποιονδήποτε. Σε πραγματικές συνθήκες, η συγκεκριμένη μέθοδος μπορεί να εφαρμοστεί σε όλες τι φορητές συσκευές (κινητά τηλέφωνα, tablets, συστήματα πλοήγησης GPS κτλ.). Μερικά από τα μειονεκτήματα που παρουσιάζει η συγκεκριμένη μέθοδος είναι τα εξής:

- η ανάκτηση των δεδομένων περιορίζεται στα δεδομένα που μπορεί ο ερευνητής να προσπελάσει από το λειτουργικό σύστημα,
- όλα τα δεδομένα είναι διαθέσιμα μόνο σε μορφή εικόνων,
- η όλη διαδικασία είναι εξαιρετικά χρονοβόρα, αν αναλογιστεί κανείς το ολοένα αυξανόμενο μέγεθος της εσωτερικής μνήμης των έξυπνων συσκευών.

Λογική απόκτηση (logical acquisition): η λογική απόκτηση συνεπάγεται ένα bit by bit αντίγραφο των αντικειμένων λογικής αποθήκευσης (π.χ. τους καταλόγους και τα αρχεία) που βρίσκονται σε ένα λογικό αποθηκευτικό χώρο (π.χ. ένα διαμέρισμα αρχείων του συστήματος). Η λογική απόκτηση έχει το πλεονέκτημα ότι οι δομές δεδομένων του συστήματος είναι ευκολότερες για την εξαγωγή και την οργάνωση μέσω ενός εργαλείου. Η λογική εξαγωγή ανακτά πληροφορίες από τη συσκευή, χρησιμοποιώντας την διεπαφή προγραμματισμού εφαρμογών του αρχικού εξοπλισμού για τον συγχρονισμό των περιεχομένων του τηλεφώνου με έναν προσωπικό υπολογιστή. Μία λογική εξαγωγή είναι γενικά ευκολότερη για να δουλέψει, καθώς δεν παράγει ένα μεγάλο δυαδικό BLOB (Binary Large Object). Ωστόσο, ένας εξειδικευμένος ερευνητής ψηφιακών πειστηρίων θα μπορέσει να εξάγει πολύ περισσότερες πληροφορίες με την μέθοδο της φυσικής απόκτησης.

Απόκτηση συστήματος αρχείων (file system acquisition): η λογική απόκτηση συνήθως δεν μας επιστρέφει πληροφορίες οι οποίες είχαν διαγραφεί από το σύστημα. Αυτό οφείλεται στο γεγονός ότι συνήθως οι πληροφορίες απομακρύνονται από το σύστημα αρχείων του τηλεφώνου. Ωστόσο, σε ορισμένες περιπτώσεις (ειδικά με πλατφόρμες που βασίζονται σε SQLite, όπως το iOS και το Android) το τηλέφωνο μπορεί να διατηρεί ένα αρχείο βάσης δεδομένων με πληροφορίες που δεν αντικαθιστούν τις διαγραμμένες πληροφορίες αλλά απλώς επισημαίνει το αρχείο της βάσης ως διαγραμμένο και διαθέσιμο για μεταγενέστερη αντικατάσταση. Σε τέτοιες περιπτώσεις, αν η συσκευή επιτρέπει πρόσβαση στο σύστημα αρχείων μέσω της διεπαφής συγχρονισμού, είναι δυνατή η ανάκτηση των πληροφοριών που

έχουν διαγραφεί. Η εξαγωγή του συστήματος αρχείων είναι χρήσιμη για την κατανόηση της δομής του αρχείου, του ιστορικού περιήγησης στο Web ή της χρήσης των εφαρμογών, καθώς και για την παροχή στον εξεταστή της δυνατότητας ανάλυσης με παραδοσιακά εργαλεία ψηφιακής εγκληματολογίας.

Φυσική απόκτηση (physical acquisition): η φυσική απόκτηση προϋποθέτει ένα αντίγραφο bit by bit ενός ολόκληρου φυσικού μέρους (π.χ. μνήμη flash). Ως εκ τούτου, είναι η μέθοδος που μοιάζει περισσότερο με την εξέταση ψηφιακών πειστηρίων ενός προσωπικού υπολογιστή. Η φυσική απόκτηση έχει το πλεονέκτημα ότι επιτρέπει την εξέταση τόσο των διαγραμμένων αρχείων, όσο και τυχόν υπολειμμάτων δεδομένων. Η φυσική εξαγωγή αποκτά πληροφορίες από τη συσκευή αποκτώντας άμεση πρόσβαση στις μνήμες φλας. Γενικά, αυτό είναι πιο δύσκολο να επιτευχθεί επειδή ο κατασκευαστής του εξοπλισμού της συσκευής χρειάζεται να θωρακίσει την συσκευή από την αυθαίρετη ανάγνωση της μνήμης. Επομένως, μια συσκευή μπορεί να είναι κλειδωμένη σε έναν συγκεκριμένο πάροχο. Για να ξεπεραστεί αυτή η ασφάλεια, οι πωλητές εργαλείων ψηφιακής εγκληματολογίας για φορητές συσκευές αναπτύσσουν συχνά τους δικούς τους φορτωτές εκκίνησης (boot loaders), επιτρέποντας στο εργαλείο ψηφιακών πειστηρίων να έχει πρόσβαση στη μνήμη (και συχνά, επίσης, να παρακάμπτει τους κωδικούς πρόσβασης των χρηστών ή τα κλειδώματα μοτίβων). Γενικά, η φυσική εξαγωγή χωρίζεται σε δύο στάδια, την φάση της απόρριψης και τη φάση της αποκωδικοποίησης. Συνολικά υπάρχουν τέσσερις διαφορετικοί τρόποι μέσω των οποίων μπορεί να υλοποιηθεί η μέθοδος της φυσικής απόκτησης, οι οποίοι είναι:

- η μέθοδος JTAG: η συγκεκριμένη μέθοδος είναι μία μη επεμβατική μορφή φυσικής απόκτησης που θα μπορούσε να πετύχει την εξαγωγή των δεδομένων από μία κινητή συσκευή, ακόμη και όταν τα δεδομένα θα ήταν δύσκολο να προσπελαστούν μέσω προγραμμάτων λογισμικού επειδή η συσκευή είναι κατεστραμμένη, κλειδωμένη ή ακόμα και κρυπτογραφημένη. Η συσκευή ωστόσο για να επιτευχθεί αυτό, πρέπει να είναι τουλάχιστον εν μέρει λειτουργική (μικρές βλάβες δεν θα εμπόδιζαν τη μέθοδο αυτή).

  Η διαδικασία περιλαμβάνει τη σύνδεση με τις λεγόμενες Test Access Ports (TAPs) σε μία συσκευή και την καθοδήγηση του επεξεργαστή για να μεταφέρει ακατέργαστα δεδομένα που είναι αποθηκευμένα σε συνδεδεμένα τσιπ μνήμης. Πρόκειται για ένα τυπικό χαρακτηριστικό που θα μπορούσε να συναντήσει κανείς σε πολλά διαφορετικά μοντέλα κινητών τηλεφώνων, το οποίο παρέχει στο κινητό τηλέφωνο μια διεπαφή χαμηλού επιπέδου εκτός του λειτουργικού συστήματος. Οι ψηφιακοί εγκληματολογικοί ερευνητές δείχνουν μεγάλο ενδιαφέρον για την συγκεκριμένη μέθοδο, καθώς θεωρητικά επιτρέπει την άμεση πρόσβαση στη μνήμη της κινητής συσκευής χωρίς να την θέτει σε κίνδυνο. Παρά το γεγονός αυτό, η μέθοδος JTAG αποτελεί μία χρονοβόρα διαδικασία και απαιτεί εκ των προτέρων γνώση συγκεκριμένων πραγμάτων (όχι μόνο του JTAG για το μοντέλο του τηλεφώνου που θα διερευνηθεί, αλλά και του τρόπου οργάνωσης εκ νέου του δυαδικού αποτελέσματος που αποτελείται από τις δομές μνήμης του τηλεφώνου).

- η μέθοδος Hex Dump: παρόμοια με την μέθοδο JTAG, η συγκεκριμένη μέθοδος είναι μία ακόμα μέθοδος για τη φυσική εξαγωγή ακατέργαστων πληροφοριών που είναι

αποθηκευμένες στη μνήμη flash. Εκτελείται συνδέοντας τον εγκληματολογικό σταθμό εργασίας στη συσκευή και στη συνέχεια, διοχετεύει στη συσκευή έναν κώδικα που δεν έχει υπογραφεί ή ένα bootloader, κάθε ένα από αυτά θα έχει σκοπό την μεταφόρτωση της μνήμης από το τηλέφωνο στον υπολογιστή. Το αποτέλεσμα που προκύπτει είναι αρκετά τεχνικό (απεικονίζεται σε δυαδική μορφή) και απαιτεί ένα άτομο με την κατάλληλη τεχνική κατάρτιση για να το αναλύσει. Επιπλέον, ο εξεταστής έρχεται σε επαφή με τεράστια ποσότητα δεδομένων, αν αναλογιστεί κανείς ότι μέσω της συγκεκριμένης μεθόδου ανακτώνται και τα διαγραμμένα δεδομένα και το καλύτερο κομμάτι είναι, ότι η όλη διαδικασία είναι ανέξοδη.

- η μέθοδος chip-off: αποτελεί μία διαδικασία που αναφέρεται στη λήψη δεδομένων απευθείας από το τσιπ μνήμης της κινητής συσκευής. Σύμφωνα με τις προετοιμασίες που σχετίζονται με αυτό το επίπεδο, το τσιπ απομακρύνεται από τη συσκευή και χρησιμοποιείται ένας αναγνώστης τσιπ ή ένα δεύτερο τηλέφωνο για την εξαγωγή δεδομένων που είναι αποθηκευμένα στην εξεταζόμενη συσκευή. Θα πρέπει να σημειωθεί ότι αυτή η μέθοδος είναι τεχνικά απαιτητική λόγω της μεγάλης ποικιλίας τύπων chip που υπάρχουν στην αγορά κινητής τηλεφωνίας. Επίσης, η διαδικασία chip-off είναι δαπανηρή, απαιτείται εξιδεικευμένη εκπαίδευση και ο εξεταστής θα πρέπει να διαθέτει συγκεκριμένο υλικό για τη διεξαγωγή απομάκρυνσης και θέρμανσης του τσιπ μνήμης. Τα bits και τα bytes των πληροφοριών που ανακτώνται από τη μνήμη δεν έχουν ακόμη αναλυθεί, αποκωδικοποιηθεί, ούτε καν ερμηνευθεί. Ακόμη και το μικρότερο λάθος στην συγκεκριμένη διαδικασία, μπορεί να οδηγήσει σε ζημιές στο τσιπ μνήμης, το οποίο θα είχε ως αποτέλεσμα τα δεδομένα να χάνονταν οριστικά. Ως εκ τούτου, οι ειδικοί συμβουλεύουν την προσφυγή στην μέθοδο chip-off όταν: α) έχουν ήδη επιχειρηθεί άλλες μέθοδοι εξαγωγής, β) είναι σημαντικό να διατηρηθεί η τρέχουσα κατάσταση της μνήμης της συσκευής, γ) το τσιπ μνήμης είναι το μόνο στοιχείο στην κινητή συσκευή που δεν έχει σπάσει.

  Η όλη διαδικασία αποτελείται από τα πέντε στάδια παρακάτω:

  1. τον εντοπισμό της τοπολογίας του τσιπ μνήμης της συσκευής,
  2. την φυσική εξαγωγή του τσιπ (αφαιρώντας το για παράδειγμα),
  3. τη διασύνδεση του τσιπ χρησιμοποιώντας κάποιο λογισμικό ανάγνωσης / προγραμματισμού,
  4. την ανάγνωση και τη μεταφορά των δεδομένων από το τσιπ στον υπολογιστή,
  5. την ερμηνεία των δεδομένων που αποκτήθηκαν (χρησιμοποιώντας αντίστροφη μηχανική).

- η μέθοδος micro read: η συγκεκριμένη μέθοδος αναφέρεται στη χειροκίνητη λήψη όλης της όψης μέσα από τους φακούς ενός ηλεκτρονικού μικροσκοπίου και στην ανάλυση των δεδομένων που εμφανίζονται στο τσιπ μνήμης, ειδικότερα στις φυσικές πύλες του τσιπ. Με λίγα λόγια, η micro read είναι μια μέθοδος που απαιτεί υψηλό επίπεδο

εμπειρογνωμοσύνης, είναι αρκετά δαπανηρή και χρονοβόρα και προορίζεται συνήθως για περιπτώσεις που σχετίζονται με θέματα εθνικής ασφάλειας.

Όσο η τεχνολογία εξελίσσεται, οι μέθοδοι γίνονται περισσότερο τεχνικά απαιτητικές με προσανατολισμό τους πάντα, την διασφάλιση της ακεραιότητας των αποκτηθέντων στοιχείων (το λεγόμενο και forensic soundness).

Οι ολοένα αυξανόμενες απαιτήσεις λοιπόν, έχει παρατηρηθεί ότι έχουν ως συνέπεια τα ακόλουθα:

- την άνοδο στις τιμές των εμπορικά διαθέσιμων εργαλείων,
- την αύξηση της χρονικής περιόδου που απαιτείται για να πραγματοποιηθεί η εξέταση,
- η τεχνική κατάρτιση των ερευνητών πρέπει να εμπλουτίζεται συνεχώς μέσω της έρευνας και της επιπρόσθετης εκπαίδευσης όπου αυτό χρειάζεται,
- μερικές μέθοδοι απόκτησης τείνουν να γίνουν ακόμη πιο επεμβατικές.[13][14]

3.4 Τρόποι εξαγωγής δεδομένων της κάρτας μνήμης(sd card)

Η πλειοψηφία των τρόπων εξαγωγής που εφαρμόζονται από τον ψηφιακό ερευνητή για την εσωτερική μνήμη του τηλεφώνου, μπορούν να εφαρμοστούν και στην περίπτωση της εξαγωγής δεδομένων από την κάρτα μνήμης της συσκευής. Συνεπώς, η χειροκίνητη απόκτηση (manual acquisition), η λογική απόκτηση (logical acquisition) και η φυσική απόκτηση (physical acquisition) έχουν και εδώ πρακτική εφαρμογή και αντίστοιχα αποτελέσματα. Επίσης, η κάρτα μνήμης μπορεί να τοποθετηθεί απευθείας στον υπολογιστή σε ειδική θύρα ανάγνωσης καρτών (card reader) και να επιτευχθεί η εξαγωγή των δεδομένων μέσω ειδικών προγραμμάτων λογισμικού. Στο σημείο αυτό, αξίζει βέβαια να σημειωθεί ότι ο συνιστώμενος τρόπος εξέτασης της κάρτας μνήμης είναι ο τελευταίος (αφαίρεσή της από την συσκευή και εξέτασή της ως φυσική μνήμη για καλύτερα αποτελέσματα).

3.5 Τρόποι εξαγωγής δεδομένων της κάρτας SIM

Η εξαγωγή των δεδομένων από την κάρτα SIM πρέπει να πραγματοποιείται έχοντας αφαιρέσει την κάρτα SIM από την συσκευή. Οποιοσδήποτε άλλος τρόπος, καλό είναι να αποφεύγετε για να διασφαλίζεται η πληρέστερη εξαγωγή των δεδομένων. Υπάρχουν ειδικές συσκευές ανάγνωσης καρτών SIM μέσω των οποίων επιτυγχάνεται αρχικά η ανάλυση και στην συνέχεια η εξαγωγή των δεδομένων. Βέβαια, το σύνηθες είναι μία κάρτα SIM να προστατεύεται από κωδικό πρόσβασης (το γνωστό σε όλους μας κωδικό PIN), κάτι το οποίο δυσκολεύει την όλη

[13] https://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/the-mobile-forensics-process-steps-types/

[14] https://en.wikipedia.org/wiki/Mobile_device_forensics

διαδικασία. Αν ισχύει αυτό, ο ψηφιακός ερευνητής ζητάει από τον χρήστη του κινητού τον κωδικό PIN και σε περίπτωση που εκείνος δεν συνεργαστεί, απευθύνεται στον εκάστοτε πάροχο κινητής τηλεφωνίας ζητώντας τον κωδικό PUK για το ξεκλείδωμα της κάρτας.

3.6 Τρόποι εξαγωγής δεδομένων της μνήμης RAM

Η εξαγωγή των δεδομένων από την μνήμη RAM είναι μία πολύπλοκη διαδικασία και απαιτεί εξειδικευμένες γνώσεις, ειδικά εργαλεία και συγκεκριμένα βήματα για να υλοποιηθεί όσο το δυνατόν καλύτερα. Σε επόμενο κεφάλαιο, θα γίνει αναλυτικότερη περιγραφή όσον αφορά την εξαγωγή δεδομένων από την μνήμη RAM.

3.7 Πληροφορίες που μπορούν να ανακτηθούν

Στην ενότητα 3.2, έγινε αναφορά και παρουσίαση των διαφόρων σημείων στα οποία μπορούν να αποθηκευτούν δεδομένα αλλά και πληροφορίες σχετικά με την χρήση μίας συσκευής και τις προτιμήσεις του χρήστη της. Οι πληροφορίες που ενδέχεται να ενδιαφέρουν τον ψηφιακό ερευνητή ποικίλλουν και αποτελούνται από τα ακόλουθα:

- τις επαφές του χρήστη,
- τις πρόσφατες κλήσεις(ανεξαρτήτως αν είναι εξερχόμενες, ληφθείσες ή αναπάντητες),
- τα μηνύματα τύπου SMS και MMS (εισερχόμενα και εξερχόμενα),
- τις ηχογραφήσεις,
- τις φωτογραφίες και τα βίντεο του χρήστη,
- το ημερολόγιο και τυχόν σημειώσεις ή υπενθυμίσεις,
- την ηλεκτρονική αλληλογραφία του χρήστη,
- τις αναζητήσεις και το ιστορικό επισκεψιμότητας στο διαδίκτυο,
- τα αρχεία και τα έγγραφα που υπάρχουν τοπικά στην συσκευή,
- τα αναγνωριστικά στοιχεία για τον χρήστη και την συσκευή,
- τα δεδομένα που σχετίζονται με το ιστορικό της γεωγραφικής τοποθεσίας του χρήστη,
- τα αρχεία που έχει διαμοιράσει / διαμοιραστεί με άλλους χρήστες,
- τα δεδομένα από διάφορες εφαρμογές που σχετίζονται με τα μέσα κοινωνικής δικτύωσης (Facebook, Instagram, Viber, Whatsapp κτλ.),
- τα αρχεία που έχουν αποθηκευτεί στο σύννεφο (cloud),

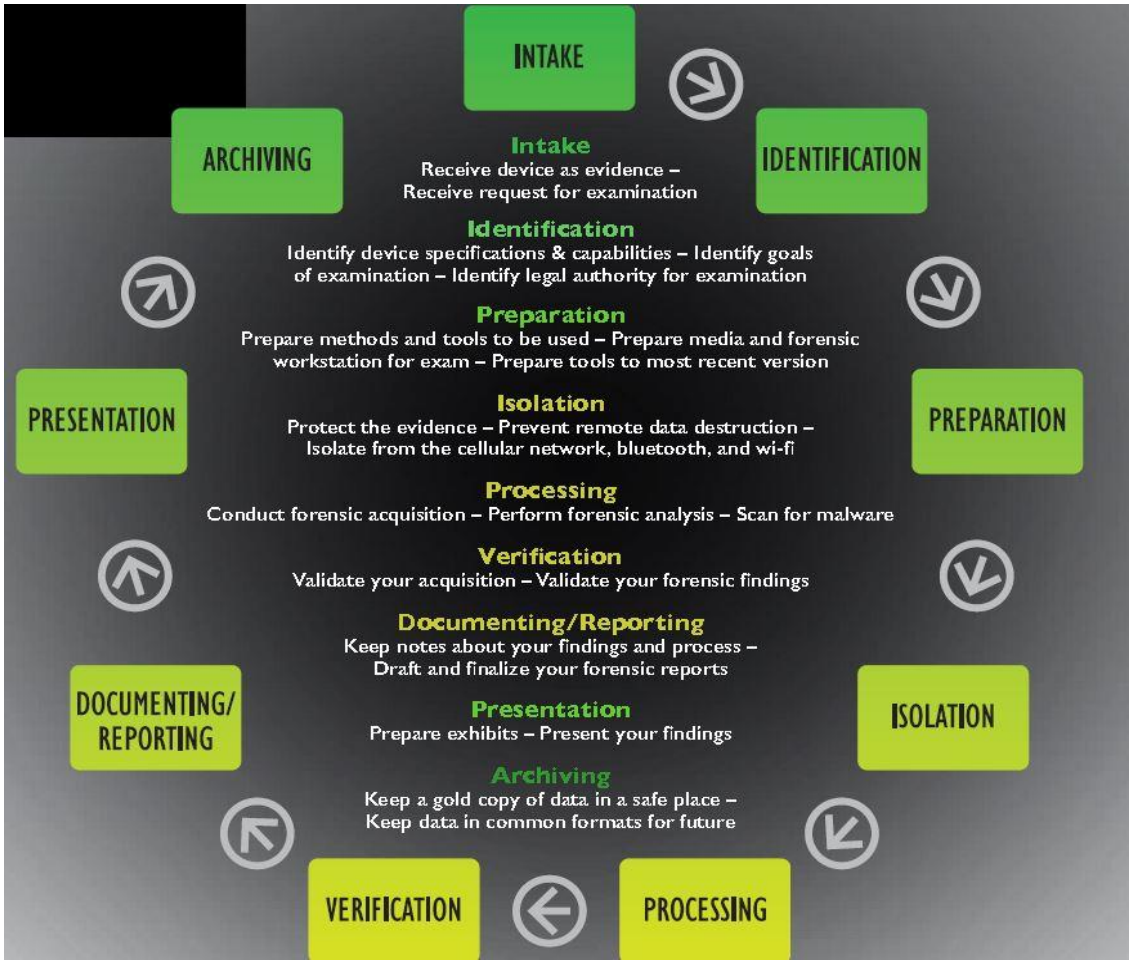
- τα δεδομένα από διάφορες εγκατεστημένες εφαρμογές,
- τα διαγραμμένα δεδομένα όλων των παραπάνω.[15]

[15] https://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/common-mobile-forensics-tools-and-techniques/

4. Βήματα εξέτασης ψηφιακών πειστηρίων κινητών συσκευών

4.1 Εισαγωγή

Τις περισσότερες φορές, μία εξέταση ψηφιακών πειστηρίων θα χρειαστεί την συνδυαστική εφαρμογή δύο ή περισσότερων εργαλείων για την εξαγωγή και την ανάλυση των δεδομένων από την κινητή συσκευή. Κάτι το οποίο χρήζει ιδιαίτερης προσοχής σε μία εξέταση, είναι να μπορεί ο ψηφιακός ερευνητής να επαληθεύσει την ακρίβεια και την ακεραιότητα των δεδομένων που έχουν ανακτηθεί. Τα βήματα που ακολουθούνται λοιπόν για την διεξαγωγή μίας εξέτασης ψηφιακών πειστηρίων είναι συνολικά εννέα και απεικονίζονται χαρακτηριστικά στην εικόνα που ακολουθεί (βλ. εικόνα 4).



Εικόνα 4. Οι εννέα φάσεις της ψηφιακής εξέτασης κινητών συσκευών.
(Πηγή: https://digital-forensics.sans.org/media/mobile-device-forensic-process-v3.pdf)

4.2 1η φάση: Intake(προσκόμιση)

Στο πρώτο στάδιο λοιπόν της εξέτασης, ο ψηφιακός εξεταστής παίρνει στην κατοχή του την συσκευή ως αποδεικτικό στοιχείο και του ζητείται να προβεί στην ψηφιακή της εξέταση. Η αρχική αυτή φάση περιλαμβάνει κυρίως το γραφειοκρατικό κομμάτι της διαδικασίας καθώς και βασικά στοιχεία που αφορούν τον ιδιοκτήτη της εκάστοτε συσκευής, με τι γεγονός έχει συνδεθεί η χρήση της συσκευής αλλά και τι είδους δεδομένα χρειάζεται να ανακτηθούν από την ψηφιακή εξέταση.

4.3 2η φάση: Identification(αναγνώριση)

Στο στάδιο αυτό, ο εξεταστής καλείται να αναγνωρίσει το μοντέλο και τις τεχνικές προδιαγραφές της συσκευής. Απαραίτητη προϋπόθεση είναι η αναλυτική καταγραφή των παραπάνω στοιχείων. Αυτό είναι κάτι που βοηθά τον εξεταστή να προσδιορίσει και να καταλήξει στα εργαλεία που θα μπορέσει να χρησιμοποιήσει για την εξέταση της συσκευής καθώς και να διαλέξει την μέθοδο ανάλυσης που θα ακολουθήσει. Το στάδιο αυτό είναι καθοριστικό για την επιλογή των κατάλληλων εργαλείων και των μεθόδων που εν τέλει θα χρησιμοποιηθούν για την ψηφιακή εξέταση της συσκευής.

4.4 3η φάση: Preparation(προετοιμασία)

Στο συγκεκριμένο στάδιο, ο εξεταστής προετοιμάζει την μέθοδο και τα εργαλεία που σκοπεύει να χρησιμοποιήσει. Η φάση της προετοιμασίας περιλαμβάνει εκτός των άλλων την στοχευμένη αναζήτηση οδηγιών για την υπό εξέταση συσκευή, την επιλογή των κατάλληλων μεθόδων και εργαλείων που θα χρησιμοποιήσει ο εξεταστής, την προετοιμασία του μηχανήματος που θα χρησιμοποιηθεί για την εξέταση με απώτερο σκοπό την εξασφάλιση του απαραίτητου εξοπλισμού, των καλωδίων που θα χρειαστούν, του κατάλληλου λογισμικού και των οδηγών(drivers) για την ψηφιακή εξέταση.

4.5 4η φάση: Isolation(απομόνωση)

Στο στάδιο αυτό, ο εξεταστής οφείλει να φροντίσει για την προστασία των δεδομένων και την απομόνωση της συσκευής από κάθε μορφής συνδεσιμότητα που θα μπορούσε να τα επηρεάσει (π.χ. σύνδεση Wi-Fi, σύνδεση Bluetooth, σύνδεση δεδομένων παρόχου). Η απομόνωση της συσκευής μπορεί να επιτευχθεί με αρκετούς τρόπους:

- την χρήση ειδικών Faraday τσαντών,
- την χρήση ειδικών jamming συσκευών,
- την ενεργοποίηση της λειτουργίας πτήσης (airplane mode).

Μέσω της απομόνωσης, ο εξεταστής αποτρέπει την εγγραφή / προσθήκη νέων δεδομένων στο τηλέφωνο μέσω τυχόν εισερχόμενων κλήσεων και μηνυμάτων κειμένου, την διαγραφή των

δεδομένων μέσω απομακρυσμένης πρόσβασης(το λεγόμενο remote wipe) καθώς και την τυχαία αντικατάσταση των υφιστάμενων δεδομένων καθώς λαμβάνονται νέες κλήσεις και μηνύματα κειμένου στην συσκευή. Η λειτουργία πτήσης θεωρείται ο πιο εύκολος και αποδοτικός τρόπος για να επιτύχει το καλύτερο αποτέλεσμα ο ψηφιακός εξεταστής.

4.6 5η φάση: Processing(επεξεργασία)

Στο στάδιο της επεξεργασίας περιλαμβάνονται η απόκτηση(acquisition) και η ανάλυση(analysis) των αποτελεσμάτων. Εάν η συσκευή περιλαμβάνει την κάρτα SIM ή περιέχει μία κάρτα μνήμης, τότε θα πρέπει να αφαιρεθούν από το τηλέφωνο πριν την πραγματοποίηση της εξέτασης της συσκευής και να υποβληθούν σε ξεχωριστή ανάλυση χρησιμοποιώντας τις κλασσικές μεθόδους απόκτησης και ανάλυσης ψηφιακών πειστηρίων, ούτως ώστε να διασφαλιστούν οι πληροφορίες ημερομηνίας και ώρας για τα αρχεία που είναι αποθηκευμένα στην κάρτα μνήμης και να μην τροποποιηθούν κατά την διαδικασία της εξέτασης. Η εκτέλεση πολλαπλών ειδών εξαγωγής δεδομένων από την ίδια συσκευή μπορεί να φανεί χρήσιμη στον εξεταστή και για την αποκωδικοποίηση των δεδομένων που λαμβάνονται από το τηλέφωνο.

4.7 6η φάση: Verification(επαλήθευση)

Στο συγκεκριμένο στάδιο, ο εξεταστής επαληθεύει την απόκτηση των δεδομένων που πραγματοποίησε στην προηγούμενη φάση καθώς και τα εγκληματολογικά ευρήματα που ανέκτησε. Αυτή η φάση περιλαμβάνει την διαδικασία της διασφάλισης των δεδομένων που εξάγονται από την κινητή συσκευή τα οποία πρέπει να ταιριάζουν με τα δεδομένα που εμφανίζει η ίδια η συσκευή. Η επαλήθευση των δεδομένων που εξάγονται μπορεί να πραγματοποιηθεί με διάφορους τρόπους. Μερικοί από αυτούς είναι οι εξής:

- έλεγχος της δεκαεξαδικής μορφής Hex (Hexadecimal): αρκετά εργαλεία προσφέρουν την δυνατότητα στον εξεταστή να επιβεβαιώσει την ακεραιότητα των δεδομένων αναλύοντας την μορφή Hex. Η εν λόγω μέθοδος απαιτεί άρτια τεχνική κατάρτιση και εμπειρία από την μεριά του εξεταστή.
- έλεγχος των hash τιμών: οι hash τιμές μπορούν να χρησιμοποιηθούν για την επαλήθευση και την ακεραιότητα των εξαγόμενων δεδομένων.
- έλεγχος και ανάλυση με πολλαπλά εργαλεία: το ιδανικό είναι να χρησιμοποιούνται διαφορετικά εργαλεία για την εξαγωγή αποτελεσμάτων, ούτως ώστε να μπορεί ο εξεταστής να συγκρίνει και να αξιολογήσει τα αποτελέσματα.

4.8 7η φάση: Documentation / Reporting(τεκμηρίωση / αναφορά)

Το στάδιο αυτό, περιλαμβάνει την σύνταξη και την οριστικοποίηση της αναφοράς - έκθεσης της ψηφιακής εξέτασης καθώς και την τεκμηρίωση των πορισμάτων. Στην αναφορά που συντάσσεται από τον εξεταστή, καταγράφονται μεταξύ των άλλων τα παρακάτω:

- η διαδικασία που χρησιμοποιήθηκε για την εξαγωγή δεδομένων από το τηλέφωνο,
- τα εργαλεία που χρησιμοποιήθηκαν,
- το είδος των δεδομένων που εν τέλει ανακτήθηκαν.

Όλα τα στοιχεία που σχετίζονται με την ψηφιακή εξέταση θα πρέπει να τεκμηριώνονται με ακρίβεια στις αναφορές του εξεταστή. Η τεκμηρίωση της εξέτασης θα πρέπει να πραγματοποιείται καθ' όλη τη διαδικασία και ο εξεταστής οφείλει να κρατάει αναλυτικές σημειώσεις σχετικά με τα βήματα που ακολούθησε κατά την ψηφιακή εξέταση.

4.9 8η φάση: Presentation(παρουσίαση)

Στο συγκεκριμένο στάδιο, περιλαμβάνεται η παρουσίαση των ευρημάτων μαζί με πληροφορίες σχετικά με την πηγή, την ημερομηνία και την ώρα, τα δεδομένα που εξήχθησαν από την συσκευή καθώς και τις εικόνες και τα βίντεο που αποτελούν μέρος των αποδεικτικών στοιχείων, όπως αυτά ανακτήθηκαν από το κινητό τηλέφωνο. Η παρουσίαση πρέπει να είναι λεπτομερής και αναλυτική, για να μην μπορούν να αμφισβητηθούν τα αποτελέσματα της εξέτασης.

4.10 9η φάση: Archiving(αρχειοθέτηση)

Τελευταίο και εξίσου σημαντικό στάδιο με τα προηγούμενα, είναι η αρχειοθέτηση. Είναι απαραίτητο τα δεδομένα να διατηρούνται σε μορφή τέτοια, ούτως ώστε να μπορούν να χρησιμοποιηθούν για μελλοντική αναφορά και για την απαραίτητη τήρηση των αρχείων. Η διατήρηση των δεδομένων που εξάγονται και τεκμηριώνονται από το κινητό τηλέφωνο, αποτελεί αναπόσπαστο και κρίσιμο μέρος της συνολικής διαδικασίας.

5. Εργαλεία εξέτασης ψηφιακών πειστηρίων σε περιβάλλον Android

5.1 Εισαγωγή

Υπάρχει πληθώρα εργαλείων ψηφιακών πειστηρίων τα οποία ανάλογα με τις ανάγκες της ψηφιακής έρευνας και εξέτασης προσφέρουν αρκετές δυνατότητες και επιλογές. Κριτήρια επιλογής συνήθως αποτελούν το κόστος, το πλήθος των υποστηριζόμενων συσκευών (αν μιλάμε για εμπορικά εργαλεία) καθώς και η εξοικείωση του ψηφιακού ερευνητή με αυτά.

5.2 Εργαλεία λογικής απόκτησης (logical acquisition)

Η μέθοδος και τα βασικά χαρακτηριστικά της λογικής απόκτησης έχουν αναλυθεί σε προηγούμενο κεφάλαιο (βλ. ενότητα 3.3). Υπάρχουν αρκετά διαθέσιμα εμπορικά και open source εργαλεία τα οποία μπορεί να χρησιμοποιήσει ο ψηφιακός εξεταστής για να πραγματοποιήσει την εξέταση μίας Android συσκευής. Ακολουθεί αναφορά σε μερικά από τα εργαλεία που εξυπηρετούν την προαναφερθείσα μέθοδο.

5.2.1 Το εργαλείο SAFT

Το SAFT είναι ένα ευκολόχρηστο εργαλείο ψηφιακής εξέτασης κινητών συσκευών που χρησιμοποιείται για την εξαγωγή αρκετών χρήσιμων πληροφοριών από την συσκευή, όπως:

- τα αρχεία καταγραφής των κλήσεων,
- τα αρχεία καταγραφής των μηνυμάτων τύπου SMS / MMS,
- την λίστα επαφών,
- τα logs περιήγησης αρχείων,
- το ιστορικό του προγράμματος περιήγησης και τους σελιδοδείκτες,
- τα logs διαφόρων εφαρμογών (Facebook, Twitter, Viber, Whatsapp, Skype κτλ.),
- τα μηνύματα ηλεκτρονικού ταχυδρομείου,
- το ιστορικό τοποθεσίας,
- το ημερολόγιο.

Το συγκεκριμένο εργαλείο έχει την δυνατότητα δημιουργίας αναφορών σε αρχεία τύπου Microsoft Word.

5.2.2 Το εργαλείο LiME

Το εργαλείο LiME (Linux Memory Extractor, προηγούμενη ονομασία DMD) είναι ένα Loadable Kernel Module (LKM), το οποίο επιτρέπει την απόκτηση της «ζωντανής» μνήμης RAM από συσκευές που είναι βασισμένες στον πυρήνα του Linux, όπως είναι και οι συσκευές τύπου Android. Το συγκεκριμένο εργαλείο υποστηρίζει την απόκτηση της μνήμης είτε στο σύστημα αρχείων της συσκευής, είτε πάνω από το δίκτυο.

Το LiME είναι μοναδικό στο είδος του, καθώς είναι το πρώτο εργαλείο που επιτρέπει την πλήρη απόκτηση της μνήμης RAM από συσκευές τύπου Android. Επίσης, σημαντικό είναι το γεγονός ότι το LiME ελαχιστοποιεί την αλληλεπίδραση του εξεταστή με τις διεργασίες που τρέχουν στον χώρο του χρήστη (user space) ή στον χώρο του πυρήνα (kernel space) γεγονός που του επιτρέπει να εξάγει αποτυπώματα μνήμης που είναι πιο αξιόπιστα από εγκληματολογικής απόψεως σε σχέση με άλλα εργαλεία που έχουν σχεδιαστεί για την απόκτηση της μνήμης από συσκευές τύπου Linux. Το LiME αποτελεί ένα εξαιρετικό εργαλείο για την απόκτηση της «ζωντανής» μνήμης των συσκευών Android και παρέχει εικόνα τόσο για τα δομημένα όσο και για τα αδόμητα δεδομένα της μνήμης RAM, των κλειδιών κρυπτογράφησης, των δεδομένων από τις εφαρμογές, τα αποσπάσματα τυχόν επικοινωνιών, των ανοιχτών αρχείων, των διαδικασιών, των εφαρμογών και των δομών του πυρήνα και του δικτύου.

5.2.3 Το εργαλείο Nandroid Backup

Το Nandroid Backup είναι μία άλλη μέθοδος για την εξαγωγή του συνολικού συστήματος αρχείων της συσκευής χρησιμοποιώντας το ομώνυμο εργαλείο. Αυτά τα αντίγραφα ασφαλείας μπορούν να δημιουργηθούν εκκινώντας την συσκευή σε προσαρμοσμένη ανάκτηση(custom recovery) και επιλέγοντας την αντίστοιχη επιλογή στο μενού. Το αντίγραφο ασφαλείας Nandroid μπορεί να δημιουργηθεί μόνο αν ο Bootloader έχει ξεκλειδωθεί και η προσαρμοσμένη ανάκτηση(custom recovery) είναι εγκατεστημένη ή η συσκευή είναι rooted. Για να ολοκληρωθεί η διαδικασία, πρέπει να χρησιμοποιηθούν το πακέτο Busybox και η εφαρμογή αντιγράφων ασφαλείας Nandroid.

5.2.4 Η σουίτα εργαλείων OSAF-TK (Open Source Android Forensics – Tool Kit)

Το OSAF-TK είναι ένα κιτ εργαλείων ανοιχτού κώδικα Android που γράφτηκε από την έκδοση 11.10 του Ubuntu ειδικά για την ανάλυση κακόβουλου λογισμικού Android. Το συγκεκριμένο κιτ εργαλείων διαθέτει όλα τα απαραίτητα για την εξέταση κώδικα και την ανάλυση εφαρμογών. Με την χρήση της συγκεκριμένης σουίτας εργαλείων, ο ψηφιακός εξεταστής μπορεί να εντοπίσει μεταξύ των άλλων μία κακόβουλη εφαρμογή και να αναλύσει περαιτέρω τις δραστηριότητές της.

5.2.5 Το Santoku Linux

Το Santoku Linux αποτελεί ένα δωρεάν κιτ εργαλείων λογισμικού ανοιχτού κώδικα και είναι αφιερωμένο στα ψηφιακά πειστήρια κινητών συσκευών, την ανάλυση κακόβουλου λογισμικού και τις δοκιμές ασφάλειας για κινητές συσκευές. Το Santoku περιέχει ένα σύνολο εργαλείων που μπορούν να καλύψουν ένα μεγάλο φάσμα ενεργειών, όπως:

- το λεγόμενο firmware flashing(υποστηρίζοντας πολλούς διαφορετικούς κατασκευαστές),
- την δυνατότητα απεικόνισης NAND,
- τις κάρτες πολυμέσων,
- την μνήμη RAM.

Αυτό το πακέτο εφαρμογών περιέχει αρκετά χρήσιμα βοηθητικά προγράμματα και σκριπτ που χρησιμοποιούνται κυρίως για την ψηφιακή εγκληματολογία κινητών συσκευών και περιλαμβάνει επίσης δωρεάν εκδόσεις ορισμένων εμπορικά διαθέσιμων εργαλείων ψηφιακής εγκληματολογίας. Το πακέτο εφαρμογών περιέχει εξομοιωτές για κινητές συσκευές, βάσεις δεδομένων κακόβουλου λογισμικού και διάφορα βοηθητικά προγράμματα για την προσομοίωση υπηρεσιών δικτύου σε δυναμική ανάλυση. Το συγκεκριμένο πακέτο περιλαμβάνει επίσης σκριπτ που απαριθμούν τις λεπτομέρειες της εφαρμογής, αυτοματοποιούν τα δυαδικά αρχεία αποκρυπτογράφησης, ανιχνεύουν κοινά θέματα ασφαλείας σε εφαρμογές καθώς και εργαλεία αποσυναρμολόγησης(disassembly) και επανασύνταξης(recompilation).

5.2.6 Το εργαλείο Andriller

Το εργαλείο Andriller είναι ένα πρόγραμμα λογισμικού που περιλαμβάνει μία συλλογή εργαλείων ψηφιακής εγκληματολογίας για κινητές συσκευές. Εκτελείτε σε λειτουργία read-only, με τρόπο ο οποίος είναι εγκληματολογικά αποδεκτός (εξασφαλίζει το λεγόμενο forensic soundness) και ο τρόπος απόκτησης που χρησιμοποιεί δεν επηρεάζει τα δεδομένα των συσκευών Android. Μερικά από τα κύρια χαρακτηριστικά που διαθέτει είναι τα εξής:

- ισχυρό «σπάσιμο» κλειδώματος οθόνης για μοτίβο, κωδικό PIN και κωδικό πρόσβασης,
- προσαρμοσμένους αποκωδικοποιητές(custom decoders) για την εξαγωγή δεδομένων από εφαρμογές Android,
- βάσεις δεδομένων για την αποκωδικοποίηση επικοινωνιών,
- δυνατότητα εξαγωγής δεδομένων από rooted και non rooted συσκευές,
- δυνατότητα λήψης στιγμιότυπων οθόνης.

Το σημαντικότερο χαρακτηριστικά του συγκεκριμένου εργαλείου είναι αναμφίβολα η δυνατότητα εξαγωγής δεδομένων από non rooted συσκευές. Τέλος, το εργαλείο δίνει την δυνατότητα στον ψηφιακό εξεταστή να παράγει report με τα αποτελέσματα σε μορφή HTML ή σε αρχεία Excel.

5.2.7 Το εργαλείο LiMEaide

Το LiMEaide είναι ένα πρόγραμμα γραμμένο σε Python που έχει σχεδιαστεί για την απομακρυσμένη απόκτηση της μνήμης RAM μίας συσκευής που τρέχει Linux και δημιουργεί ένα volatility προφίλ το οποίο ο εξεταστής μπορεί να αναλύσει αργότερα τοπικά στον υπολογιστή του. Το συγκεκριμένο εργαλείο απλοποιεί αρκετά την διαδικασία της απομακρυσμένης ψηφιακής εξέτασης μίας συσκευής. Ο τρόπος λειτουργίας του LiMEaide είναι αρκετά απλός.

Πρώτα, ο εξεταστής πραγματοποιεί μία σύνδεση μέσω SSH με τη συσκευή - στόχο. Στην συνέχεια, μεταφέρονται τα απαραίτητα αρχεία που θα χρειαστούν για το χτίσιμο του custom rom. Έπειτα δημιουργείται το Loadable Kernel Module (LKM) μέσω του LiME και μέσω αυτού, γίνεται η απόκτηση της μνήμης RAM. Στην συνέχεια μεταφέρεται το αποκτηθέν αρχείο πίσω στο μηχάνημα που πραγματοποιεί την εξέταση. Τέλος, χτίζεται το απαραίτητο Volatility προφίλ που θα χρειαστεί για την ανάλυση των εξαχθέντων αποτελεσμάτων.

5.3 Εργαλεία φυσικής απόκτησης (physical acquisition)

Η μέθοδος και τα βασικά χαρακτηριστικά της φυσικής απόκτησης έχουν αναλυθεί και αυτά σε προηγούμενο κεφάλαιο (βλ. ενότητα 3.3). Εδώ θα κάνουμε αναφορά κυρίως σε εμπορικά εργαλεία τα οποία μπορεί να χρησιμοποιήσει ο ψηφιακός εξεταστής για να πραγματοποιήσει την εξέταση μίας Android συσκευής.

5.3.1 Το εργαλείο UFED Ultimate

Πρόκειται για ένα από τα πιο γνωστά εμπορικά εργαλεία της εταιρίας Cellebrite. Το Unified Forensic Examination Device (UFED) είναι ίσως το πιο διαδεδομένο και ένα από τα καλύτερα εμπορικά διαθέσιμα εργαλεία που υπάρχουν στον τομέα της ψηφιακής εγκληματολογίας, υποστηρίζοντας συνολικά πάνω από 26.179 προφίλ συσκευών (στοιχεία Οκτωβρίου 2018). Μπορεί να ανακτήσει δεδομένα με μεγαλύτερη ταχύτητα και ακρίβεια από ένα μεγάλο φάσμα κινητών συσκευών, εφαρμογών και κοινωνικών δικτύων. Το UFED μπορεί να συνδεθεί με το κινητό τηλέφωνο μέσω καλωδίου USB, τεχνολογίας υπερύθρων (IR) ή BlueTooth (BT). Μπορεί να αποκτήσει δεδομένα τόσο λογικά όσο και φυσικά από μία κινητή συσκευή και ουσιαστικά μπορεί να αναλύσει κάθε τμήμα της μνήμης μίας συσκευής χρησιμοποιώντας προχωρημένες τεχνικές λογικής απόκτησης, απόκτησης συστήματος αρχείων καθώς και τεχνικές φυσικής απόκτησης. Μέσω του συγκεκριμένου εργαλείου, ο εξεταστής μπορεί να ανακτήσει τα ακόλουθα:

- δυνατότητα παράκαμψης του κλειδώματος της οθόνης ανεξαρτήτως του τρόπου κλειδώματος(κωδικός, κωδικός PIN, μοτίβο),
- τις επαφές του χρήστη,
- τα μηνύματα τύπου MMS / SMS,
- τα αρχεία καταγραφής των κλήσεων,
- δεδομένα χρήσεως των εφαρμογών,
- τυχόν φωτογραφίες και βίντεο αποθηκευμένα στην συσκευή,
- κρυφά και διεγραμμένα αρχεία της συσκευής.

5.3.2 Το εργαλείο Belkasoft Evidence Center

Πρόκειται για μία σουίτα εργαλείων που χρησιμοποιείται για την απόκτηση, τον εντοπισμό, την εξαγωγή και την ανάλυση των ψηφιακών στοιχείων που είναι αποθηκευμένα σε υπολογιστές, σε

κινητές συσκευές, στην μνήμη RAM και στο σύννεφο(cloud). Μερικά από τα κύρια χαρακτηριστικά του συγκεκριμένου εργαλείου είναι τα εξής:

- το λεγόμενο «σκάλισμα» δεδομένων(data carving) επιτρέπει στον ψηφιακό εξεταστή να εντοπίσει στοιχεία τα οποία έχουν διαγραφεί, καταστραφεί ή απλά δεν έχουν αποθηκευτεί ποτέ τοπικά στο σκληρό δίσκο (όπως το αρχείο σελίδας(page file), το αρχείο αδρανοποίησης(hibernation file), το περιεχόμενο της μνήμης RAM κτλ.).
- δυνατότητα ανάκτησης των κατεστραμμένων και μη ολοκληρωμένων βάσεων δεδομένων SQLite, καθώς και επαναφορά των διεγραμμένων εγγραφών και των αρχείων ιστορικού που έχουν εκκαθαριστεί.
- δυνατότητα ανάλυσης της ζωντανής μνήμης RAM και εξαγωγή πληροφοριών όπως η ιδιωτική περιήγηση του χρήστη, το διεγραμμένο ιστορικό του προγράμματος περιήγησης, οι συζητήσεις από εφαρμογές κοινωνικής δικτύωσης, το ιστορικό χρήσης εφαρμογών τύπου cloud και πολλά άλλα.
- με την χρήση των ενσωματωμένων δυνατοτήτων όπως το File System Explorer, το Hex Viewer και το Type Converter, ο ψηφιακός εξεταστής έχει την δυνατότητα να εξετάσει σε βάθος το περιεχόμενο των αρχείων και των φακέλων στη συσκευή.
- με την δυνατότητα BelkaScript, ο ψηφιακός εξεταστής έχει την δυνατότητα να γράψει τα δικά του σκριπτ, προκειμένου να αυτοματοποιήσει ορισμένες από τις συνήθεις λειτουργίες και να επεκτείνει περαιτέρω τη λειτουργικότητα του προϊόντος.

5.3.3 Το εργαλείο MOBILedit Forensic Express

Το εργαλείο MOBILedit Forensic Express χρησιμοποιείται για την εξαγωγή δεδομένων από κινητές συσκευές (δεδομένα τα οποία είναι αποθηκευμένα είτε τοπικά, είτε στο cloud) και προσφέρει στον εξεταστή την δυνατότητα λεπτομερούς ανάλυσης των εξαγόμενων δεδομένων καθώς και την δυνατότητα δημιουργίας αναφορών. Χρησιμοποιώντας το εν λόγω εργαλείο, ο ψηφιακός εξεταστής μπορεί να πραγματοποιήσει τόσο φυσική όσο και λογική απόκτηση των δεδομένων. Μερικά από τα στοιχεία που μπορούν να ανακτηθούν με την χρήση του συγκεκριμένου εργαλείου είναι τα παρακάτω:

- μηνύματα τύπου SMS/ MMS,
- μηνύματα ηλεκτρονικού ταχυδρομείου,
- μηνύματα από εφαρμογές κοινωνικών δικτύων,
- αναλυτικά αρχεία καταγραφής των κλήσεων,
- τυχόν κρυφά και διεγραμμένα μηνύματα ή δεδομένα της συσκευής,
- φωτογραφίες, ηχογραφήσεις και βίντεο αποθηκευμένα στην συσκευή,
- ανάκτηση των δεδομένων που σχετίζονται με την τοποθεσία της συσκευής και δυνατότητα γραφικής απεικόνισης των αποτελεσμάτων,
- ιστορικό περιήγησης στο Διαδίκτυο,
- σελιδοδείκτες από τα προγράμματα περιήγησης που είναι εγκατεστημένα στη συσκευή,
- αρχεία καταγραφής του συστήματος (system logs).

5.3.4 Το εργαλείο Oxygen Forensic Detective

Το συγκεκριμένο εμπορικά διαθέσιμο εργαλείο είναι από τα κορυφαία στο είδος του και το εύρος των συσκευών που υποστηρίζει ξεπερνάει τα 19.500 προφίλ συσκευών (στοιχεία Οκτωβρίου 2018). Μερικές από τις κύριες δυνατότητες που διαθέτει το συγκεκριμένο εργαλείο είναι οι εξής:

- απενεργοποίηση ή προσπέλαση του κλειδώματος της οθόνης σε μεγάλη γκάμα συσκευών,
- ανάκτηση των επαφών,
- ανάκτηση των μηνυμάτων,
- ανάκτηση του ημερολογίου και των σημειώσεων,
- ανάκτηση των κωδικών πρόσβασης διάφορων εφαρμογών,
- πρόσβαση σε δεδομένα cloud εφαρμογών,
- ανάκτηση δεδομένων τοποθεσίας της συσκευής και γραφική απεικόνιση των αποτελεσμάτων για ευκολότερη κατανόηση από τον εξεταστή.

5.3.5 Το εργαλείο XRY – Extract

Άλλο ένα πολύ γνωστό εμπορικό εργαλείο το οποίο υποστηρίζει εκτός από την μέθοδο της φυσικής απόκτησης και την λογική απόκτηση. Υποστηρίζει συνολικά πάνω από 25.000 προφίλ συσκευών (στοιχεία Οκτωβρίου 2018) και ορισμένα βασικά χαρακτηριστικά του συγκεκριμένου εργαλείου είναι τα ακόλουθα:

- απενεργοποίηση ή ανάκτηση του κωδικού κλειδώματος της οθόνης,
- πρόσβαση σε προστατευμένα και διεγραμμένα αρχεία,
- ανάκτηση των επαφών,
- ανάκτηση των μηνυμάτων τύπου SMS/MMS,
- ανάκτηση του ημερολογίου και των σημειώσεων,
- ανάκτηση αρχείων εικόνας,
- υποστήριξη πάνω από 2.100 εφαρμογών, συμπεριλαμβάνοντας όλες τις γνωστές εφαρμογές κοινωνικών δικτυών.

5.3.6 Το εργαλείο Paraben E3: DS

Αποτελεί και αυτό ένα εμπορικά διαθέσιμο εργαλείο το οποίο υποστηρίζει πολλαπλές μεθόδους απόκτησης των δεδομένων (λογικής, φυσικής, cloud). Το εν λόγω εργαλείο υποστηρίζει συνολικά πάνω από 29.000 προφίλ συσκευών (στοιχεία Οκτωβρίου 2018) και εμπλουτίζεται συνεχώς με καινούρια. Όπως και τα προηγούμενα εργαλεία που αναφέρθηκαν, υποστηρίζει και αυτό πληθώρα επιλογών μεταξύ των οποίων:

- παράκαμψη του τρόπου κλειδώματος της οθόνης(ανεξαρτήτως μεθόδου),
- διαχωρισμός των ανακτηθέντων αρχείων ανάλογα με την κατηγορία στην οποία υπάγονται,
- ανάκτηση διάφορων τύπων διεγραμμένων αρχείων,

- δυνατότητα αυτόματης ανίχνευσης της υπό εξέτασης συσκευής και εμφάνιση επιλογών που είναι συμβατές με την εκάστοτε συσκευή.

5.3.7 Τα εργαλεία AccessData MPE+ and nFIELD

Συνδυάζοντας το εργαλείο MPE+ και το nFIELD της AccessData, ο ψηφιακός εξεταστής μπορεί να εξετάσει επιτυχώς πάνω από 10.000 προφίλ συσκευών (στοιχεία Οκτωβρίου 2018) και να συλλέξει τα παρακάτω στοιχεία:

- το ιστορικό τηλεφωνικών κλήσεων,
- τυχόν αρχεία ήχου που εμπεριέχονται στην συσκευή,
- αρχεία φωτογραφιών ή βίντεο,
- μηνύματα κειμένου SMS (δυνατότητα ανάκτησης ακόμα και διεγραμμένων μηνυμάτων),
- ιστορικό πλοήγησης στο διαδίκτυο και ιστορικό λήψεων.

Χάρη στους πύργους κινητής τηλεφωνίας και τις διάφορες τεχνολογίες που βοηθούν στον εντοπισμό της τοποθεσίας του χρήστη (GPS, Bluetooth, Wi-Fi), υπάρχει η δυνατότητα σε πραγματικό χρόνο να καθοριστεί κατά προσέγγιση η θέση του κατόχου της συσκευής σε συγκεκριμένη χρονική στιγμή.

5.4 Εργαλεία ανάλυσης της μνήμης (memory analysis)

Εφόσον έχουν εξαχθεί επιτυχώς από την συσκευή τα υπό εξέταση δεδομένα, ο ψηφιακός εξεταστής χρειάζεται να τα αναλύσει και να εξάγει συμπεράσματα ανάλογα με τα στοιχεία που θα προκύψουν. Τα περισσότερα εμπορικά εργαλεία που έχουν ήδη αναφερθεί, προσφέρουν την δυνατότητα στον ψηφιακό εξεταστή να εξετάσει αναλυτικά την μνήμη της συσκευής.

5.5 Εργαλεία ανάλυσης της ζωντανής μνήμης (live memory analysis)

Για να επιτύχει το καλύτερο δυνατό αποτέλεσμα και να καταφέρει επιτυχώς την ανάλυση της ζωντανής μνήμης, ο εξεταστής οφείλει να διαλέξει το κατάλληλο εργαλείο που θα τον βοηθήσει να ερμηνεύσει πλήρως τις εξαχθέντες πληροφορίες. Ακολουθεί αναφορά και σύντομη περιγραφή των πιο γνωστών εργαλείων ανάλυσης της ζωντανής μνήμης.

5.5.1 Το Volatility Framework

Το Volatility αποτελεί μία συλλογή εργαλείων ανοιχτού κώδικα και είναι γραμμένο στην γλώσσα Python. Χρησιμοποιείται για την εξαγωγή ψηφιακών πειστηρίων από δείγματα ζωντανής μνήμης RAM. Οι τεχνικές εξαγωγής εκτελούνται τελείως ανεξάρτητα από το σύστημα που ερευνάται αλλά προσφέρουν ορατότητα στην κατάσταση του χρόνου εκτέλεσης του συστήματος. Το Volatility έχει ως στόχο να εισαγάγει τους ανθρώπους στις τεχνικές και τις πολυπλοκότητες που σχετίζονται με την εξαγωγή ψηφιακών πειστηρίων από δείγματα ζωντανής μνήμης RAM και παρέχει μία πλατφόρμα για περαιτέρω εργασία σε αυτό το συναρπαστικό και ταυτόχρονα απαιτητικό πεδίο.

5.5.2 Το εργαλείο Autopsy

Το εργαλείο Autopsy είναι μια ψηφιακή πλατφόρμα εγκληματολογίας και αποτελεί την γραφική διεπαφή του χρήστη με την σουίτα «The Sleuth Kit» και άλλα εργαλεία ψηφιακής εγκληματολογίας. Όσον αφορά τις συσκευές Android, υπάρχει το Android Analyzer, το οποίο είναι ένα module που επιτρέπει στον ψηφιακό εξεταστή να αναλύσει τα αρχεία SQLite καθώς και άλλου είδους αρχεία από μία συσκευή Android. Το συγκεκριμένο εργαλείο θα μπορεί να εφαρμοστεί σχεδόν σε όλα τα physical dumps από τις περισσότερες συσκευές Android (να σημειωθεί ότι το συγκεκριμένο εργαλείο δεν παρέχει μέθοδο απόκτησης).

Ενεργοποιώντας το συγκεκριμένο module λοιπόν, ο εξεταστής θα πρέπει να είναι σε θέση να εξάγει τα ακόλουθα:

- τα μηνύματα κειμένου τύπου SMS/MMS,
- τα αρχεία καταγραφής του μητρώου κλήσεων,
- τις επαφές του χρήστη,
- τα δεδομένα γεωγραφικής τοποθεσίας GPS από το πρόγραμμα περιήγησης και την εφαρμογή χάρτες της Google,
- τα δεδομένα γεωγραφικής τοποθεσίας GPS από τα αρχεία cache.wifi και cache.cell.

5.5.3 Το Rekall Framework

Το Rekall αποτελεί και αυτό μία συλλογή εργαλείων ανοιχτού κώδικα και είναι γραμμένο στην γλώσσα Python. Όπως και το Volatility, χρησιμοποιείται για την εξαγωγή ψηφιακών πειστηρίων από δείγματα μνήμης RAM. Το Rekall μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα υποστηρίζει την γλώσσα Python. Είναι συμβατό με όλες τις μεγάλες πλατφόρμες λειτουργικών συστημάτων (Windows, Linux, Mac OS). Το Rekall, εκτός από την ανάλυση, παρέχει επίσης και την δυνατότητα απόκτησης της μνήμης για όλα τα μεγάλα λειτουργικά συστήματα.

6. Ψηφιακή εγκληματολογία «ζωντανής» μνήμης (Live Memory Forensics)

6.1 Εισαγωγή

Αν θέλαμε να δώσουμε έναν ορισμό για την ψηφιακή εγκληματολογία «ζωντανής» μνήμης, θα λέγαμε ότι αποτελεί την εξέταση της φυσικής μνήμης (RAM) για την υποστήριξη της ψηφιακής εγκληματολογίας, της ανάλυσης κακόβουλου λογισμικού και άλλων εξιδεικευμένων ενεργειών. Το βασικό πλεονέκτημα σε σχέση με τους άλλους τύπους ψηφιακής εγκληματολογίας(όπως τη δικτυακή ή των αποθηκευτικών συσκευών) είναι ότι ένα μεγάλο μέρος των εργασιών που εκτελείται σε μία συσκευή, εμφανίζεται μόνο στη μνήμη. Συνεπώς, χάνεται η πρόσβαση σε πολλές χρήσιμες πληροφορίες όταν το σύστημα απενεργοποιείται και δεν υπάρχει τρόπος ανακάλυψης ή ανάκτησής τους.

6.2 Προδιαγραφές εργαλείων για την απόκτηση της μνήμης RAM

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, υπάρχει μια σειρά εργαλείων και μεθόδων τα οποία είναι διαθέσιμα για τη λήψη των memory dumps. Υπάρχουν ορισμένες απαιτήσεις που κάθε τέτοιο εργαλείο οφείλει να πληροί. Οι απαραίτητες προϋποθέσεις που πρέπει να τηρούνται λοιπόν είναι οι ακόλουθες:

1. Το εργαλείο πρέπει οπωσδήποτε να τρέχει σε λειτουργία Kernel Mode. Η λειτουργία αυτή είναι απαραίτητη σε ένα εργαλείο ψηφιακής εγκληματολογίας που πραγματοποιεί μεθόδους απόκτησης της μνήμης RAM. Πολλές εφαρμογές προστατεύουν προληπτικά κατά της απόκτησης (dumping) τα σετ μνήμης που τις αφορούν. Συνεπώς, ένα εργαλείο το οποίο τρέχει σε User Mode είναι απλά καταστροφικό σε τέτοιες περιπτώσεις. Σε ιδανικές συνθήκες, εργαλεία τέτοιου τύπου θα διαβάσουν μηδενικά ή τυχαία δεδομένα αντί για τις πραγματικές πληροφορίες που υπάρχουν. Στο χειρότερο σενάριο όμως, όταν τρέχει στο παρασκήνιο μία antidebugging προστασία, τότε θα εφαρμοστούν αυτόματα άμεσα μέτρα προκειμένου να καταστραφούν ολοσχερώς οι όποιες προστατευμένες πληροφορίες υπάρχουν και εν συνεχεία θα πραγματοποιηθεί το κλείδωμα και η απενεργοποίηση ή τυχόν επανεκκίνηση της συσκευής, κάτι το οποίο θα καθιστούσε οποιαδήποτε περαιτέρω ανάλυση ανέφικτη. Για να αποτραπεί αυτό, οι ψηφιακοί ερευνητές οφείλουν να επιλέξουν το κατάλληλο εργαλείο ανάλογα με την προς εξέταση συσκευή.

2. Το εργαλείο πρέπει να αφήνει το μικρότερο δυνατό αποτύπωμα στην συσκευή. Όσο μικρότερο είναι το αποτύπωμα που αφήνει ένα εργαλείο απόκτησης μνήμης, τόσο το καλύτερο. Όλα τα εργαλεία αφήνουν ίχνη κατά την διαδικασία απόκτησης, κάτι το οποίο ενδεχομένως να καταστρέψει και ορισμένα αποδεικτικά στοιχεία. Όσο

λιγότερο συμβαίνει κάτι τέτοιο, τόσο πιο αξιόπιστα θεωρούνται τα αποτελέσματα της ψηφιακής εξέτασης.

3. Το εργαλείο πρέπει να διαθέτει φορητότητα. Τα εργαλεία απόκτηση της μνήμης πρέπει να είναι φορητά (portable) και έτοιμα για να τρέξουν από τον ερευνητή σε ανεξάρτητες συσκευές (π.χ. usb flash drive ή απομακρυσμένη τοποθεσία δικτύου). Εργαλεία που χρειάζεται να εγκατασταθούν στην συσκευή δεν είναι αποδεκτά για προφανείς λόγους.

4. Το εργαλείο πρέπει να έχει πρόσβαση μόνο για ανάγνωση (read only access). Οποιοδήποτε αξιόλογο ψηφιακό εργαλείο δεν θα γράψει ποτέ κάτι στην μνήμη της συσκευής, γιατί κάτι τέτοιο θα τροποποιούσε τα προς εξέταση δεδομένα και θα καθιστούσε τα αποτελέσματα της ψηφιακής εξέτασης αναξιόπιστα.

6.3 Διαφορές μεταξύ «dead» και «live» ψηφιακής εγκληματολογίας

Προτού παρουσιάσουμε τις διαφορές μεταξύ των δύο μεθόδων ψηφιακής εγκληματολογίας, ας δούμε περιληπτικά λίγα πράγματα για την καθεμία.

6.3.1 Στατική ανάλυση («dead» analysis)

Αποτελεί τον πιο παραδοσιακό τρόπο ανάλυσης στην ψηφιακή εγκληματολογία και επικεντρώνεται στην εξέταση ενός ακριβούς αντίγραφου του δίσκου για να αποκτήσει πρόσβαση στα περιεχόμενα της μνήμης, όπως τα αρχεία που διαγράφονται, το ιστορικό περιήγησης στο διαδίκτυο, τα τμήματα αρχείων (file fragments), τις συνδέσεις δικτύου, τα ανοιγμένα αρχεία, το ιστορικό σύνδεσης των χρηστών κ.τ.λ.

Απώτερος σκοπός είναι να δημιουργήσει ο εξεταστής μια χρονική γραμμή η οποία θα του δώσει μία εικόνα σχετικά με τις δραστηριότητες που έγιναν στο υπό εξέταση σύστημα πριν τον τερματισμό της λειτουργίας του. Στην στατική ανάλυση, χρησιμοποιούνται διάφορα εργαλεία για την απόκτηση της μνήμης και για το ξεκαθάρισμα των ανακτηθέντων στοιχείων. Αυτό συμβάλλει σε τυχόν περαιτέρω ανάλυση των στοιχείων καθώς και στην παρουσίασή τους από τον ψηφιακό εξεταστή. Τα δεδομένα ανακτώνται με τη χρήση διαφορετικών ειδών εξωτερικών συσκευών όπως καλώδια USB, εξωτερικούς σκληρούς δίσκους ή δίσκους CD και DVD και στη συνέχεια τα δεδομένα αυτά μεταφέρονται στο εγκληματολογικό εργαστήριο όπου και αναλύονται περαιτέρω. Εφόσον ολοκληρωθούν τα προαναφερθέντα, τα στοιχεία που έχουν εξαχθεί αποτελούν αποδεικτικά στοιχεία.

6.3.2 Δυναμική ανάλυση («live» analysis)

Εξαιτίας των νέων προκλήσεων που αντιμετωπίζουν οι εξεταστές ψηφιακής εγκληματολογίας, τα τελευταία χρόνια, υπάρχει μία αυξανόμενη χρήση της «ζωντανής» ανάλυσης. Εδώ ο εξεταστής συλλέγει και αναλύει τις πληροφορίες διατηρώντας το υπό εξέταση σύστημα λειτουργικό. Με την χρήση της συγκεκριμένης μεθόδου, πληροφορίες όπως διεργασίες που εκτελούνται, ανοικτές συνδέσεις δικτύου και κλειδιά κρυπτογράφησης μπορούν να ανακτηθούν, κάτι το οποίο δεν θα ήταν δυνατό με την χρήση της στατικής ανάλυσης.

6.3.3 Διαφορές μεταξύ των δύο προσεγγίσεων

Η πρώτη βασική διαφορά που υπάρχει μεταξύ «dead» και «live» ανάλυσης, είναι το γεγονός ότι κατά την διαδικασία της πρώτης, η αντιγραφή των αρχείων γίνεται χωρίς την βοήθεια του λειτουργικού συστήματος της υπό εξέτασης συσκευής. Αντίθετα, η αντιγραφή των αρχείων στη δεύτερη περίπτωση γίνεται χρησιμοποιώντας το λειτουργικό σύστημα της συσκευής, κάτι το οποίο εμπεριέχει το ρίσκο της διαφοροποίησης των δεδομένων ή της εξαγωγής πειραγμένων δεδομένων, εάν έχουν ληφθεί στην συσκευή μέτρα κατά της ψηφιακής εξέτασης. Αυτός είναι και ο βασικότερος λόγος για τον οποίο κυρίως στο παρελθόν, ένας ψηφιακός εξεταστής δεν θα χρησιμοποιούσε την μέθοδο της «ζωντανής» ανάλυσης.

Μία ακόμα βασική διαφορά μεταξύ των δύο μεθόδων, είναι οι πληροφορίες που μπορούν να εξαχθούν με την χρήση τους. Με την μέθοδο της «live» ανάλυσης, ο εξεταστής μπορεί να αποκτήσει πρόσβαση σε πληροφορίες που μέχρι πρότινος με την χρήση της πιο διαδεδομένης μεθόδου (της λεγόμενης και «dead» ανάλυσης) δεν είχε την δυνατότητα, όπως τυχόν κλειδιά κρυπτογράφησης της συσκευής, φανερές και κρυφές διεργασίες που τρέχουν στην μνήμη RAM κατά την στιγμή της απόκτησης, προηγμένο κακόβουλο λογισμικό που έχει προγραμματιστεί για να τρέχει και να αφήνει τυχόν ίχνη αποκλειστικά στην μνήμη RAM της συσκευής και διάφορα άλλα.

Επίσης, κάτι τελευταίο που αξίζει να αναφερθεί, είναι ότι κατά βάση ένας ψηφιακός εξεταστής μπορεί να αναλύσει τους ίδιους τύπους δεδομένων χρησιμοποιώντας τεχνικές «live» και «dead» ανάλυσης. Η ουσιαστική διαφορά που παρουσιάζεται, είναι η αξιοπιστία των εξαγόμενων αποτελεσμάτων. Αυτό συμβαίνει γιατί οι τεχνικές «live» ανάλυσης βασίζονται σε εφαρμογές που τρέχουν σε πραγματικό χρόνο και μπορεί να έχουν τροποποιηθεί για να παράγουν εσφαλμένα δεδομένα.

6.3.4 Προβλήματα/προκλήσεις ψηφιακής εγκληματολογίας μνήμης σε συσκευές Android

Η εξέταση της ψηφιακής εγκληματολογίας μνήμης παρουσιάζει αρκετές ιδιαιτερότητες καθώς και δυσκολίες που καλείται ο ψηφιακός εξεταστής να προσπεράσει.

Το συνεχώς μεταβαλλόμενο περιβάλλον μίας συσκευής η οποία παραμένει σε λειτουργία καθ' όλη την διάρκεια της ψηφιακής εξέτασης «ζωντανής» μνήμης, είναι το πρώτο και σημαντικότερο πρόβλημα που καλείται ο ψηφιακός εξεταστής να επιλύσει. Υπάρχει ενδεχόμενο τα δεδομένα της συσκευής να αλλοιωθούν ή να διαφοροποιηθούν κατά την διαδικασία της απόκτησης, γι' αυτό και ο εξεταστής οφείλει να εφαρμόσει τα κατάλληλα αντίμετρα προκειμένου να αποτρέψει ή να μετριάσει στο ελάχιστο την αλλοίωση των δεδομένων.

Μία εξίσου σημαντική μεταβλητή είναι η ακεραιότητα των αποκτηθέντων δεδομένων. Η ψηφιακή εγκληματολογία «ζωντανής» μνήμης εμπεριέχει την πιθανότητα αλλοίωσης των υπό εξέταση δεδομένων και αποτελεί πρόκληση για τον εξεταστή να μπορεί να αποδείξει ότι τα δεδομένα που θα παρουσιαστούν στο δικαστήριο είναι αξιόπιστα και δεν έχουν μεταβληθεί.

Επίσης, σε πραγματικές συνθήκες, δεν είναι πάντα εύκολο να πληρούνται όλες οι προϋποθέσεις για να εξαχθεί επιτυχώς και με τη λιγότερη δυνατή αλλοίωση η μνήμη Ram της υπό εξέτασης συσκευής. Το ιδανικό σενάριο για τον ψηφιακό εξεταστή θα περιλάμβανε τα εξής:

η συσκευή να είναι rooted, να έχει ενεργοποιημένη τη δυνατότητα USB Debugging για πρόσβαση στη συσκευή και ο πυρήνας Linux να έχει τροποποιηθεί και να είναι «φιλικός» στην χρήση της εντολής insmod (η εντολή insmod επιτρέπει την εισαγωγή ενός module μέσα στον πυρήνα του λειτουργικού συστήματος Linux). Δυστυχώς, τα περισσότερα από τα παραπάνω δεν συναντώνται εύκολα σε πραγματικές συνθήκες, με αποτέλεσμα ο εξεταστής να κινδυνεύει να αλλοιώσει την κατάσταση της μνήμης RAM προσπαθώντας να φέρει την συσκευή στην επιθυμητή κατάσταση.[16]

[16] Marthie Lessing, Basie von Solms (2008), Live Forensic Acquisition as Alternative to Traditional Forensic Processes

7. Εξέταση «ζωντανής» μνήμης κινητής συσκευής Android

7.1 Εισαγωγή

Στο πρακτικό μέρος της παρούσας διπλωματικής, θα εξετάσουμε ένα δείγμα μνήμης από συσκευή Android 4.1.2 το οποίο αποκτήθηκε με την χρήση του εργαλείου LiME και θα το αναλύσουμε περαιτέρω με την χρήση του εργαλείου Volatility.

7.2 Διαδικασία εξέτασης «ζωντανής» μνήμης

Για την ψηφιακή μας εξέταση, θα αναλύσουμε ένα δείγμα μνήμης το οποίο όπως έχουμε ήδη αναφέρει στην Εισαγωγή του 7ου κεφαλαίου, έχει αποκτηθεί με την χρήση του εργαλείου LiME, που αποτελεί ένα εργαλείο λογικής απόκτησης (περισσότερα στοιχεία για τη μέθοδο της λογικής απόκτησης έχουν παρουσιαστεί στο κεφάλαιο 3.2) και το οποίο θεωρείται το καλύτερο εργαλείο για εξαγωγή της «ζωντανής» μνήμης μίας συσκευής, όπως αναλύθηκε και στην ενότητα 5.2.2. Για την ανάλυσή μας λοιπόν, θα χρησιμοποιήσουμε το λειτουργικό Kali Linux το οποίο είναι βασισμένο στη διανομή Debian του Linux και έκδοσης 4.13.0 όπως απεικονίζεται στην παρακάτω φωτογραφία (βλ. εικόνα 5).

```
root@kali:~# uname -a
Linux kali 4.13.0-kali1-amd64 #1 SMP Debian 4.13.10-1kali2 (2017-11-08) x86_64 GNU/Linux
```

Εικόνα 5. Η έκδοση του Kali Linux που θα χρησιμοποιηθεί

Για να ξεκινήσουμε την ανάλυσή μας, πρώτα πρέπει να εγκαταστήσουμε το Volatility. Το Volatility επιλέχθηκε επειδή είναι ανοιχτού κώδικα και περιλαμβάνει μεγάλη συλλογή εργαλείων, ενώ ταυτόχρονα υποστηρίζει πληθώρα λειτουργικών συστημάτων και τύπους αρχείων (όπως hibernation αρχεία, crash dump αρχεία, αρχεία τύπου .lime, VMware snapshot αρχεία κτλ.)Εκτελούμε αρχικά την εντολή όπως φαίνεται παρακάτω (βλ. εικόνα 6).

```
root@kali:~# git clone https://github.com/volatilityfoundation/volatility
Cloning into 'volatility'...
remote: Counting objects: 26602, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 26602 (delta 2), reused 3 (delta 1), pack-reused 26596
Receiving objects: 100% (26602/26602), 20.08 MiB | 915.00 KiB/s, done.
Resolving deltas: 100% (19109/19109), done.
```

Εικόνα 6. Εγκατάσταση του Volatility Framework

Στη συνέχεια μεταβαίνουμε στο φάκελο εγκατάστασης που δημιουργήθηκε. Στο μονοπάτι volatility/volatility/plugins/overlays/linux/ αντιγράφουμε το profile του Volatility για την συσκευή που θέλουμε να αναλύσουμε για να ξεκινήσουμε την εξέτασή μας.

```
root@kali:~# cd volatility/volatility/plugins/overlays/linux/
root@kali:~/volatility/volatility/plugins/overlays/linux# ls -al
total 508
drwxr-xr-x 2 root root   4096 Aug 20 21:34 .
drwxr-xr-x 5 root root   4096 Aug 20 20:15 ..
-rw-r--r-- 1 root root  26000 Aug 20 21:01 elf.py
-rw-r--r-- 1 root root  28891 Aug 20 21:34 elf.pyc
-rw-r--r-- 1 root root 278263 Aug 20 18:28 goldfish-2.6.29.zip
-rw-r--r-- 1 root root      0 Aug 20 20:11 __init__.py
-rw-r--r-- 1 root root    149 Aug 20 20:15 __init__.pyc
-rw-r--r-- 1 root root  85724 Aug 20 20:11 linux.py
-rw-r--r-- 1 root root  79854 Aug 20 20:15 linux.pyc
```

Εικόνα 7. Αντιγραφή του Volatility profile για την υπό εξέταση συσκευή

Εκτελώντας την εντολή ls -al παρατηρούμε ότι το αρχείο προστέθηκε επιτυχώς στο μονοπάτι του Volatility.

Θέλοντας να επιβεβαιώσουμε ότι το profile που εισάγαμε αναγνωρίζεται επιτυχώς από το Volatility, εκτελούμε την εντολή όπως φαίνεται κάτωθι που μας επιβεβαιώνει ότι το profile υπάρχει και είναι διαθέσιμο για χρήση.

```
root@kali:~# cd volatility/
root@kali:~/volatility# python vol.py --info | grep Linux
Volatility Foundation Volatility Framework 2.6
Linuxgoldfish-2_6_29ARM - A Profile for Linux goldfish-2.6.29 ARM
LinuxAMD64PagedMemory          - Linux-specific AMD 64-bit address space.
linux_aslr_shift           - Automatically detect the Linux ASLR shift
linux_banner               - Prints the Linux banner information
linux_yarascan             - A shell in the Linux memory image
```

Εικόνα 8. Αποτέλεσμα εντολής python vol.py --info | grep Linux

Είμαστε έτοιμοι λοιπόν να ξεκινήσουμε να αναλύουμε το δείγμα μνήμης της συσκευής. Αρχικά, θα τρέξουμε δύο δικτυακές εντολές προκειμένου να δούμε τον πίνακα ARP της υπό εξέτασης συσκευής καθώς και τις πληροφορίες που σχετίζονται με τις διευθύνσεις IP, τις διευθύνσεις MAC, τα στοιχεία των Interfaces που χρησιμοποιούνται καθώς και αν είναι ενεργό το promiscuous mode σε κάποιο από τα interfaces.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_arp
Volatility Foundation Volatility Framework 2.6
[::                                    ] at 00:00:00:00:00:00    on lo
[10.0.2.3                              ] at 52:54:00:12:35:03    on eth0
[10.0.2.2                              ] at 52:54:00:12:35:02    on eth0
```

Εικόνα 9. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_arp

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_ifconfig
Volatility Foundation Volatility Framework 2.6
Interface        IP Address           MAC Address        Promiscous Mode
---------------- -------------------- ------------------ ---------------
lo               127.0.0.1            00:00:00:00:00:00  False
eth0             10.0.2.15            52:54:00:12:34:56  False
```

Εικόνα 10. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_ifconfig

Εφόσον το promiscuous mode είναι False, καταλαβαίνουμε ότι δεν γίνεται κάποιο sniffing πάνω στην συσκευή.

Άλλη μία δικτυακή εντολή που μπορούμε να εκτελέσουμε είναι η ακόλουθη:

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_route_cache
Volatility Foundation Volatility Framework 2.6
Interface        Destination          Gateway
---------------- -------------------- -------
eth0             95.211.162.18        10.0.2.2
eth0             74.125.237.203       10.0.2.2
eth0             74.125.237.202       10.0.2.2
eth0             54.225.150.210       10.0.2.2
eth0             10.0.2.3             10.0.2.3
lo               10.0.2.15            10.0.2.15
lo               0.0.0.0              0.0.0.0
eth0             54.243.82.218        10.0.2.2
eth0             107.22.187.100       10.0.2.2
eth0             74.125.237.170       10.0.2.2
eth0             173.194.79.108       10.0.2.2
eth0             74.125.237.172       10.0.2.2
eth0             107.22.187.100       10.0.2.2
lo               10.0.2.15            10.0.2.15
eth0             74.125.237.171       10.0.2.2
lo               10.0.2.15            10.0.2.15
eth0             74.125.237.204       10.0.2.2
eth0             107.22.211.9         10.0.2.2
eth0             10.0.2.3             10.0.2.3
eth0             173.194.79.109       10.0.2.2
eth0             74.125.237.202       10.0.2.2
lo               10.0.2.15            10.0.2.15
lo               10.0.2.15            10.0.2.15
eth0             74.125.129.109       10.0.2.2
eth0             192.168.43.221       10.0.2.2
lo               10.0.2.15            10.0.2.15
eth0             192.168.43.221       10.0.2.2
eth0             75.101.143.120       10.0.2.2
eth0             74.125.237.203       10.0.2.2
eth0             54.243.77.51         10.0.2.2
eth0             184.73.220.212       10.0.2.2
eth0             184.73.220.212       10.0.2.2
eth0             75.101.143.120       10.0.2.2
eth0             8.8.8.8              10.0.2.2
eth0             107.22.211.9         10.0.2.2
eth0             54.243.77.51         10.0.2.2
eth0             173.194.79.108       10.0.2.2
lo               10.0.2.15            10.0.2.15
lo               10.0.2.15            10.0.2.15
eth0             54.243.82.218        10.0.2.2
lo               10.0.2.15            10.0.2.15
eth0             54.243.43.116        10.0.2.2
eth0             54.225.150.210       10.0.2.2
eth0             74.125.237.172       10.0.2.2
```

Εικόνα 11. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_route_cache

Τρέχοντας την συγκεκριμένη εντολή, μπορούμε να δούμε με ποιες IP διευθύνσεις έχει επικοινωνήσει η συσκευή μας στο παρελθόν.

Στη συνέχεια, θα τρέξουμε ορισμένες εντολές που σχετίζονται με τις διεργασίες που έτρεχαν πάνω στην συσκευή την στιγμή που πραγματοποιήθηκε η απόκτηση της μνήμης, προσπαθώντας να αναγνωρίσουμε τυχόν «περίεργες» διεργασίες που τρέχουν στο παρασκήνιο.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_pslist
Volatility Foundation Volatility Framework 2.6
Offset     Name                 Pid             PPid            Uid             Gid    DTB        Start Time
---------- -------------------- --------------- --------------- --------------- ------ ---------- ----------
0xf3812c00 init                 1               0               0               0      0x33a88000 2014-02-25 04:56:15 UTC+0000
0xf3812800 kthreadd             2               0               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3812400 ksoftirqd/0          3               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3812000 events/0             4               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3819c00 khelper              5               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3819800 suspend              6               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3819400 kblockd/0            7               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3819000 cqueue               8               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf385ac00 kseriod              9               2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf385a800 kmmcd                10              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf385a400 pdflush              11              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf385a000 pdflush              12              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a35c00 kswapd0              13              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a35800 aio/0                14              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a58400 mtdblockd            24              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a58800 kstriped             25              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a58c00 hid_compat           26              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a56800 rpciod/0             29              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a56c00 mmcqd                30              2               0               0      ---------- 2014-02-25 04:56:15 UTC+0000
0xf3a56000 ueventd              31              1               0               0      0x33a94000 2014-02-25 04:56:15 UTC+0000
0xf3a56400 servicemanager       32              1               1000            1000   0x33c0c000 2014-02-25 04:56:18 UTC+0000
0xf3a35000 vold                 33              1               0               0      0x33ca4000 2014-02-25 04:56:18 UTC+0000
0xf3a58000 netd                 35              1               0               0      0x33c14000 2014-02-25 04:56:18 UTC+0000
0xf3bf9c00 debuggerd            36              1               0               0      0x33c18000 2014-02-25 04:56:18 UTC+0000
0xf3bf9800 rild                 37              1               1001            1001   0x33c20000 2014-02-25 04:56:18 UTC+0000
0xf3bf9400 surfaceflinger       38              1               1000            1003   0x33c24000 2014-02-25 04:56:18 UTC+0000
0xf3bf9000 zygote               39              1               0               0      0x33c4c000 2014-02-25 04:56:18 UTC+0000
0xf3c32c00 drmserver            40              1               1019            1019   0x33c50000 2014-02-25 04:56:18 UTC+0000
0xf3c32800 mediaserver          41              1               1013            1005   0x33c58000 2014-02-25 04:56:18 UTC+0000
0xf3c32400 installd             42              1               0               0      0x33c5c000 2014-02-25 04:56:18 UTC+0000
0xf3c32000 keystore             43              1               1017            1017   0x33c64000 2014-02-25 04:56:18 UTC+0000
0xf3c41c00 qemud                44              1               0               0      0x33c68000 2014-02-25 04:56:18 UTC+0000
0xf3c41000 sh                   47              1               2000            1007   0x33c98000 2014-02-25 04:56:18 UTC+0000
0xf3c7ec00 adbd                 48              1               0               0      0x33c74000 2014-02-25 04:56:19 UTC+0000
0xf3d25000 system_server        220             39              1000            1000   0x33b60000 2014-02-25 04:56:41 UTC+0000
0xf3e2f400 ndroid.systemui      276             39              -               10033  0x33e78000 2014-02-25 04:56:57 UTC+0000
0xf3ef1400 m.android.phone      308             39              1001            1001   0x33f38000 2014-02-25 04:57:03 UTC+0000
0xf3f57400 ndroid.launcher      324             39              -               10014  0x33f8c000 2014-02-25 04:57:03 UTC+0000
0xf3fef400 d.process.acore      354             39              -               10010  0x28c18000 2014-02-25 04:57:12 UTC+0000
0xf3ef1800 tlook.Z7:engine      462             39              -               10051  0x24c48000 2014-02-25 04:57:37 UTC+0000
0xe4c5dc00 putmethod.latin      477             39              -               10036  0x24c88000 2014-02-25 04:57:40 UTC+0000
0xe4d75800 d.process.media      530             39              -               10007  0x24d98000 2014-02-25 04:57:45 UTC+0000
0xe4c5d800 twitter.android      554             39              -               10045  0x24df8000 2014-02-25 04:57:46 UTC+0000
0xe4d75000 m.android.email      568             39              -               10030  0x24e8c000 2014-02-25 04:57:48 UTC+0000
```

Εικόνα 12. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_pslist (1/2)

0xe4d75000	m.android.email	568	39	-	10030	0x24e8c000	2014-02-25 04:57:48 UTC+0000
0xe100dc00	er:bdservice_v1	671	39	-	10049	0x21048000	2014-02-25 04:57:52 UTC+0000
0xe4e52400	er.forecast.pro	727	39	-	10056	0x33ed0000	2014-02-25 04:58:05 UTC+0000
0xe5b68000	ngs.android.pop	928	39	-	10059	0x28dec000	2014-02-25 05:06:45 UTC+0000
0xe1118c00	libestool2.so	959	956	-	10059	0x28f08000	2014-02-25 05:06:52 UTC+0000
0xe8d5e400	ackageinstaller	975	39	-	10018	0x28f30000	2014-02-25 05:07:05 UTC+0000
0xe8f0d400	id.defcontainer	988	39	-	10029	0x28ee8000	2014-02-25 05:07:08 UTC+0000
0xe8d65800	com.svox.pico	1003	39	-	10015	0x21120000	2014-02-25 05:07:34 UTC+0000
0xe4d55400	cm.aptoide.pt	1016	39	-	10057	0x24d68000	2014-02-25 05:07:39 UTC+0000
0xe4d75400	.quicksearchbox	1036	39	-	10006	0x24d3c000	2014-02-25 05:08:18 UTC+0000
0xe4f5b800	om.devhd.feedly	1095	39	-	10046	0x28d2c000	2014-02-25 05:09:43 UTC+0000
0xe1030800	bnix.pdf.reader	1141	39	-	10052	0x21000000	2014-02-25 05:10:14 UTC+0000
0xe102d800	org.jtb.httpmon	1185	39	-	10061	0x21024000	2014-02-25 05:10:56 UTC+0000
0xe8ffd800	com.android.mms	1221	39	-	10019	0x28da4000	2014-02-25 05:12:19 UTC+0000
0xe8ebd000	sh	1255	1200	-	10061	0x28db4000	2014-02-25 05:12:28 UTC+0000
0xe8e60000	ndroid.contacts	1321	39	-	10010	0x25b78000	2014-02-25 05:14:43 UTC+0000
0xe5982c00	einfinity.photo	1368	39	-	10047	0x25ac8000	2014-02-25 05:15:27 UTC+0000
0xe4e97c00	ndroid.settings	1420	39	1000	1000	0x24ec4000	2014-02-25 05:16:03 UTC+0000
0xf3fef000	sh	1454	48	0	0	0x28e64000	2014-02-25 05:16:24 UTC+0000
0xf3f9ac00	sh	1461	1454	0	0	0x28e58000	2014-02-25 05:16:25 UTC+0000
0xe8cb4c00	insmod	1468	1461	0	0	0x28c90000	2014-02-25 05:16:44 UTC+0000

Εικόνα 13. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_pslist (2/2)

Τα πεδία που εμφανίζονται στις παραπάνω φωτογραφίες είναι τα εξής: το offset που αποτελεί ουσιαστικά την τρέχουσα θέση ενός αρχείου όπως αυτό διατηρείται στον πυρήνα του λειτουργικού, το όνομα της διεργασίας, το Pid (Process ID), το Ppid (Parent process ID), το Uid (User ID), το Gid (Group ID), το DTB (Directory Table Base) και τέλος το Start Time που αποτελεί την ώρα εκκίνησης της διεργασίας.

Μία αντίστοιχη εντολή που μπορούμε να εκτελέσουμε για να μας επιστρέψει μία συγκεντρωμένη και πιο στοχευμένη εικόνα είναι η εντολή psaux, όπως φαίνεται παρακάτω.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_psaux
Volatility Foundation Volatility Framework 2.6
Pid    Uid    Gid    Arguments
1      0      0      /init
2      0      0      [kthreadd]
3      0      0      [ksoftirqd/0]
4      0      0      [events/0]
5      0      0      [khelper]
6      0      0      [suspend]
7      0      0      [kblockd/0]
8      0      0      [cqueue]
9      0      0      [kseriod]
10     0      0      [kmmcd]
11     0      0      [pdflush]
12     0      0      [pdflush]
13     0      0      [kswapd0]
14     0      0      [aio/0]
24     0      0      [mtdblockd]
25     0      0      [kstriped]
26     0      0      [hid_compat]
29     0      0      [rpciod/0]
30     0      0      [mmcqd]
31     0      0      /sbin/ueventd
32     1000   1000   /system/bin/servicemanager
33     0      0      /system/bin/vold
35     0      0      /system/bin/netd
36     0      0      /system/bin/debuggerd
37     1001   1001   /system/bin/rild
38     1000   1003   /system/bin/surfaceflinger
39     0      0      zygote /bin/app_process -Xzygote /system/bin --zygote --start-system-server
40     1019   1019   /system/bin/drmserver
41     1013   1005   /system/bin/mediaserver
42     0      0      /system/bin/installd
43     1017   1017   /system/bin/keystore /data/misc/keystore
44     0      0      /system/bin/qemud
47     2000   1007   /system/bin/sh
48     0      0      /sbin/adbd
220    1000   1000   system_server
276    10033  10033  com.android.systemui
308    1001   1001   com.android.phone
324    10014  10014  com.android.launcher
354    10010  10010  android.process.acore
462    10051  10051  com.outlook.Z7:engine
477    10036  10036  com.android.inputmethod.latin
530    10007  10007  android.process.media
554    10045  10045  com.twitter.android
568    10030  10030  com.android.email
671    10049  10049  com.lidroid.fileexplorer:bdservice_v1
```

Εικόνα 14. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_psaux (1/2)

```
727   10056  10056  local.weather.forecast.pro
928   10059  10059  com.estrongs.android.pop
959   10059  10059  /data/data/com.estrongs.android.pop/files/libestool2.so 39623 /data/data/com.estrongs.android.pop/files/comm/tool_port
975   10018  10018  com.android.packageinstaller
988   10029  10029  com.android.defcontainer
1003  10015  10015  com.svox.pico
1016  10057  10057  cm.aptoide.pt
1036  10006  10006  com.android.quicksearchbox
1095  10046  10046  com.devhd.feedly
1141  10052  10052  com.foobnix.pdf.reader
1185  10061  10061  org.jtb.httpmon
1221  10019  10019  com.android.mms
1255  10061  10061  sh
1321  10010  10010  com.android.contacts
1368  10047  10047  com.blueinfinity.photo
1420  1000   1000   com.android.settings
1454  0      0      /system/bin/sh -
1461  0      0      sh
1468  0      0      insmod lime.ko path=/sdcard/mem.dmp format=lime
```

Εικόνα 15. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_psaux (2/2)

Έχοντας λοιπόν τα αποτελέσματα των δύο παραπάνω εντολών, παρατηρούμε τα εξής ενδιαφέροντα αποτελέσματα:

- υπάρχουν τέσσερα διαφορετικά PID's που σχετίζονται με την εκτέλεση του shell (47,1255,1454,1461). Το πρώτο PID (47) δείχνει να σχετίζεται με την εκτέλεση του adb (Android Debug Bridge). Τα δύο τελευταία PID's (1454,1461) φαίνεται ότι μπορεί να σχετίζονται με την εκτέλεση της εντολής insmod (όπως έχουμε αναφέρει ήδη, η εντολή insmod επιτρέπει την εισαγωγή ενός module μέσα στον πυρήνα του Linux) και κατά πάσα πιθανότητα σχετίζεται με την εισαγωγή και εκτέλεση του LiME για την απόκτηση της μνήμης από την συσκευή. Για να επιβεβαιώσουμε τον ισχυρισμό μας, εκτελούμε την εντολή lsmod, η οποία μας επιστρέφει μία λίστα με τα loaded kernel modules πάνω στην συσκευή.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsmod
Volatility Foundation Volatility Framework 2.6
bf0010d0 lime 8696
```

Εικόνα 16. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsmod

Πράγματι λοιπόν διαπιστώνουμε ότι το LiME έχει φορτωθεί στην υπό εξέταση συσκευή μας και εκτελώντας την εντολή psaux συγκεκριμένα για την διεργασία insmod, μπορούμε να δούμε επακριβώς και το μονοπάτι που ορίστηκε για να αποθηκευτεί το dump της μνήμης από το LiME.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_psaux |grep insmod
Volatility Foundation Volatility Framework 2.6
1468   0      0      insmod lime.ko path=/sdcard/mem.dmp format=lime
```

Εικόνα 17. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_psaux |grep insmod

- Η διεργασία με το PID 1255, παρατηρούμε ότι έχει το ίδιο UID με κωδικό 10061 με την διεργασία με PID 1185 και είναι κάτι το οποίο θα εξετάσουμε περαιτέρω στη συνέχεια.

Για να αντιληφθούμε καλύτερα λοιπόν τον συσχετισμό μεταξύ των δύο διεργασιών, θα εκτελέσουμε την εντολή lsof και για τα δύο PID's. Η συγκεκριμένη εντολή μας επιστρέφει μία λίστα με το μονοπάτι και την περιγραφή του αρχείου (file descriptor) για κάθε διεργασία που θα δώσουμε σαν παράμετρο.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1255
Volatility Foundation Volatility Framework 2.6
Offset           Name                     Pid      FD      Path
---------------- ------------------------ -------- -------- ----
0x00000000e8ebd000 sh                        1255        0 pipe:[4791]
0x00000000e8ebd000 sh                        1255        1 pipe:[4792]
0x00000000e8ebd000 sh                        1255        2 pipe:[4793]
0x00000000e8ebd000 sh                        1255        8 /dev/  properties
```

Εικόνα 18. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1255

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1185
Volatility Foundation Volatility Framework 2.6
Offset             Name                           Pid      FD       Path
------------------ ------------------------------ -------- -------- ----
0x00000000e102d800 org.jtb.httpmon                    1185        0 /dev/null
0x00000000e102d800 org.jtb.httpmon                    1185        1 /dev/null
0x00000000e102d800 org.jtb.httpmon                    1185        2 /dev/null
0x00000000e102d800 org.jtb.httpmon                    1185        3 /dev/log/main
0x00000000e102d800 org.jtb.httpmon                    1185        4 /dev/log/radio
0x00000000e102d800 org.jtb.httpmon                    1185        5 /dev/log/events
0x00000000e102d800 org.jtb.httpmon                    1185        6 /system/framework/core.jar
0x00000000e102d800 org.jtb.httpmon                    1185        7 /system/framework/core-junit.jar
0x00000000e102d800 org.jtb.httpmon                    1185        8 /dev/__properties__
0x00000000e102d800 org.jtb.httpmon                    1185        9 /dev/binder
0x00000000e102d800 org.jtb.httpmon                    1185       10 /system/framework/bouncycastle.jar
0x00000000e102d800 org.jtb.httpmon                    1185       11 /system/framework/ext.jar
0x00000000e102d800 org.jtb.httpmon                    1185       12 /system/framework/framework.jar
0x00000000e102d800 org.jtb.httpmon                    1185       13 /system/framework/android.policy.jar
0x00000000e102d800 org.jtb.httpmon                    1185       14 /system/framework/services.jar
0x00000000e102d800 org.jtb.httpmon                    1185       15 /system/framework/apache-xml.jar
0x00000000e102d800 org.jtb.httpmon                    1185       16 /system/framework/framework.jar
0x00000000e102d800 org.jtb.httpmon                    1185       17 /system/framework/framework-res.apk
0x00000000e102d800 org.jtb.httpmon                    1185       18 /system/etc/system_fonts.xml
```

Εικόνα 19. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1185 (1/3)

```
0x00000000e102d800 org.jtb.httpmon                    1185       19 /system/etc/fallback_fonts.xml
0x00000000e102d800 org.jtb.httpmon                    1185       20 /system/framework/core.jar
0x00000000e102d800 org.jtb.httpmon                    1185       21 /dev/urandom
0x00000000e102d800 org.jtb.httpmon                    1185       22 pipe:[4570]
0x00000000e102d800 org.jtb.httpmon                    1185       23 /dev/cpuctl/apps/tasks
0x00000000e102d800 org.jtb.httpmon                    1185       24 /dev/cpuctl/apps/bg_non_interactive/tasks
0x00000000e102d800 org.jtb.httpmon                    1185       25 socket:[4567]
0x00000000e102d800 org.jtb.httpmon                    1185       26 pipe:[4568]
0x00000000e102d800 org.jtb.httpmon                    1185       27 pipe:[4568]
0x00000000e102d800 org.jtb.httpmon                    1185       28 pipe:[4570]
0x00000000e102d800 org.jtb.httpmon                    1185       29 /anon_inode:/[eventpoll]
0x00000000e102d800 org.jtb.httpmon                    1185       30 /dev/ashmem
0x00000000e102d800 org.jtb.httpmon                    1185       31 /dev/ashmem
0x00000000e102d800 org.jtb.httpmon                    1185       32 /data/app/org.jtb.httpmon-1.apk
0x00000000e102d800 org.jtb.httpmon                    1185       33 /data/app/org.jtb.httpmon-1.apk
0x00000000e102d800 org.jtb.httpmon                    1185       34 /data/app/org.jtb.httpmon-1.apk
0x00000000e102d800 org.jtb.httpmon                    1185       35 pipe:[4586]
0x00000000e102d800 org.jtb.httpmon                    1185       36 pipe:[4586]
0x00000000e102d800 org.jtb.httpmon                    1185       37 /anon_inode:/[eventpoll]
0x00000000e102d800 org.jtb.httpmon                    1185       38 socket:[4594]
0x00000000e102d800 org.jtb.httpmon                    1185       39 /data/data/org.jtb.httpmon/files/UpdateService.jar
0x00000000e102d800 org.jtb.httpmon                    1185       40 /data/data/org.jtb.httpmon/files/UpdateService.jar
0x00000000e102d800 org.jtb.httpmon                    1185       41 /data/data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.jar
0x00000000e102d800 org.jtb.httpmon                    1185       42 /data/data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.jar
0x00000000e102d800 org.jtb.httpmon                    1185       43 /727/task/1538
```

Εικόνα 20. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1185 (2/3)

```
0x00000000e102d800 org.jtb.httpmon          1185      43 /727/task/1538
0x00000000e102d800 org.jtb.httpmon          1185      46 /1185/task/1531
0x00000000e102d800 org.jtb.httpmon          1185      47 /data/org.jtb.httpmon/shared_prefs/org.jtb.httpmon_preferences.xml.bak
0x00000000e102d800 org.jtb.httpmon          1185      48 /1185/task/1531
0x00000000e102d800 org.jtb.httpmon          1185      49 pipe:[4791]
0x00000000e102d800 org.jtb.httpmon          1185      50 pipe:[4792]
0x00000000e102d800 org.jtb.httpmon          1185      51 []
0x00000000e102d800 org.jtb.httpmon          1185      52 pipe:[4793]
0x00000000e102d800 org.jtb.httpmon          1185      53 /727/task/1538
0x00000000e102d800 org.jtb.httpmon          1185      55 /meminfo
0x00000000e102d800 org.jtb.httpmon          1185      56 pipe:[6535]
0x00000000e102d800 org.jtb.httpmon          1185      58 pipe:[6535]
0x00000000e102d800 org.jtb.httpmon          1185      59 /anon_inode:/[eventpoll]
0x00000000e102d800 org.jtb.httpmon          1185      60 /system/batterystats.bin
```

Εικόνα 21. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_lsof -p 1185 (3/3)

Από τα αποτελέσματα της εντολής lsof για την διεργασία με PID 1185, παρατηρούμε δύο περίεργα αρχεία με κατάληψη .jar (Java αρχεία δηλαδή) τα οποία είναι αποθηκευμένα σε φάκελο με αρχεία της εφαρμογής org.jtb.httpmon. Η συγκεκριμένη εφαρμογή είναι μία εφαρμογή του λειτουργικού συστήματος Android για την παρακολούθηση της κατάστασης διαφόρων HTTP servers. Όπως φαίνεται στην εικόνα 20 λοιπόν, οι εγγραφές 39-42 σχετίζονται με τα αρχεία rathrazdaeizaztaxchj.jar και UpdateService.jar. Το όνομα του 1ου αρχείου είναι αρκετά περίεργο και το όνομα του 2ου παραπέμπει σε αρχείο που προσπαθεί να μιμηθεί legitimate αρχείο για αναβάθμιση εφαρμογής. Λαμβάνοντας σαν δεδομένο όμως ότι στο λειτουργικό σύστημα Android οποιαδήποτε εφαρμογή χρειάζεται να αναβαθμιστεί το κάνει μέσω του επίσημου Play Store της Google, μπορούμε να πούμε με αρκετή σιγουριά ότι η συγκεκριμένη διεργασία είναι ύποπτη.

Θέλοντας να επικεντρώσουμε την ανάλυση μας στην ύποπτη διεργασία που εντοπίσαμε, εκτελούμε την εντολή dentry_cache η οποία μας επιστρέφει το filesystem που υπάρχει στη μνήμη για κάθε ενεργό σημείο mount και μπορεί να ανακτήσει ακόμα και ονόματα αρχείων τα οποία έχουν διαγραφεί στο παρελθόν. Το αποτέλεσμα που επιστρέφει η συγκεκριμένη εντολή έχει το ακόλουθο format:

MD5|name|inode|mode_as_string|UID|GID|size|atime|mtime|ctime|crtime

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dentry_cache | grep org\.jtb\.httpmon
Volatility Foundation Volatility Framework 2.6
0|data/org.jtb.httpmon/files/UpdateService.dex|1220|0|10061|10061|3688|4083860824|4083860832|0|4083860840
0|data/org.jtb.httpmon|1077|0|10061|10061|2048|4084219248|4084219256|0|4084219264
0|app/org.jtb.httpmon-1.odex|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/cache|946|0|10061|10061|2048|4083091928|4083091936|0|4083091944
0|dalvik-cache/data@app@org.jtb.httpmon-1.apk@classes.dex|1094|0|1000|10061|102144|4084265800|4084265808|0|4084265816
0|app/org.jtb.httpmon-1.apk|1063|0|1000|1000|124408|4084268760|4084268768|0|4084268776
0|data/org.jtb.httpmon/shared_prefs|1214|0|10061|10061|2048|4083107752|4083107760|0|4083107768
0|data/org.jtb.httpmon/shared_prefs/org.jtb.httpmon_preferences.xml|481|0|10061|10061|996|4083829632|4083829640|0|4083829648
0|data/org.jtb.httpmon/cache/com.android.renderscript.cache|1204|0|10061|10061|2048|4083070392|4083070400|0|4083070408
0|data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.jar|1230|0|10061|10061|37661|4082926344|4082926352|0|4082926360
0|data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.odex|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/files/UpdateService.jar|1210|0|10061|10061|1993|4083926456|4083926464|0|4083926472
0|data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.dex|1240|0|10061|10061|96920|4082925752|4082925760|0|4082925768
0|data/org.jtb.httpmon/files/UpdateService.odex|0|0|0|0|0|0|0|0
0|Download/[Megafileupload]org.jtb.httpmon.apk|174|0|1000|1015|124408|4084033896|4084033904|0|4084033912
0|data/org.jtb.httpmon/files/a/k|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/files|774|0|10061|10061|2048|4084291792|4084291800|0|4084291808
0|data/org.jtb.httpmon/files/a/p|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/files/a/a|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/files/a|789|0|10061|10061|2048|4084386000|4084386008|0|4084386016
0|data/org.jtb.httpmon/shared_prefs/org.jtb.httpmon_preferences.xml.bak|0|0|0|0|0|0|0|0
0|data/org.jtb.httpmon/lib|1084|0|1000|1000|2048|4084265504|4084265512|0|4084265520
```

Εικόνα 22. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dentry_cache | grep org\.jtb\.httpmon

Τα αποτελέσματα που παρατηρούμε έχοντας το μηδέν(0) σε όλα τα πεδία, αφορούν εγγραφές οι οποίες δεν κατάφεραν να γίνουν επιτυχώς initialized. Αν εντοπίζαμε στα αποτελέσματα της εντολής εγγραφές που επέστρεφαν λανθασμένους ή non-ascii χαρακτήρες, αυτό θα σήμαινε ότι το directory που θα σχετιζόταν με την εν λόγω εγγραφή θα είχε διαγραφεί αλλά το μέρος του ονόματος του αρχείου από την δομή δεδομένων δεν θα είχε διαγραφεί επιτυχώς.

Θέλοντας να ανακτήσουμε περισσότερες πληροφορίες σχετικά με την συγκεκριμένη διεργασία, εκτελούμε την εντολή proc_maps δίνοντας σαν παράμετρο το όνομα του 1ου αλλά και του 2ου αρχείου που έχουμε εντοπίσει ως ύποπτα, η οποία μας επιστρέφει πληροφορίες αναφορικά με την μνήμη της συγκεκριμένης διεργασίας, περιλαμβάνοντας τα heaps, τα stacks, και τις shared libraries.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_proc_maps
-p 1185 | grep 'UpdateService.jar'
Volatility Foundation Volatility Framework 2.6
0x00000000e102d800     1185 org.jtb.httpmon      0x0000000049b80000 0x0000000049b81000 r--          0x0
 31      1       1210 /data/data/org.jtb.httpmon/files/UpdateService.jar
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_proc_maps
-p 1185 | grep 'rathrazdaeizaztaxchj.jar'
Volatility Foundation Volatility Framework 2.6
0x00000000e102d800     1185 org.jtb.httpmon      0x000000004b883000 0x000000004b884000 r--        0x9000
 31      1       1230 /data/data/org.jtb.httpmon/files/rathrazdaeizaztaxchj.jar
```

Εικόνα 23. Αποτελέσματα των εντολών python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_proc_maps -p 1185 | grep 'UpdateService.jar' και python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_proc_maps -p 1185 | grep 'rathrazdaeizaztaxchj.jar'

Γνωρίζοντας πλέον επακριβώς την θέση των αρχείων στη μνήμη, θα τα κάνουμε dump στον σκληρό δίσκο για να μπορέσουμε να τα αναλύσουμε περαιτέρω χρησιμοποιώντας την εντολή dump_map.

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dump_map
-p 1185 --dump-dir /root/Downloads -s 0x0000000049b80000
Volatility Foundation Volatility Framework 2.6
Task       VM Start   VM End         Length Path
---------- ---------- ---------- ---------- ----
      1185 0x49b80000 0x49b81000     0x1000 /root/Downloads/task.1185.0x49b80000.vma
```

Εικόνα 24. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dump_map -p 1185 --dump-dir /root/Downloads -s 0x0000000049b80000

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dump_map
-p 1185 --dump-dir /root/Downloads -s 0x000000004b883000
Volatility Foundation Volatility Framework 2.6
Task       VM Start   VM End         Length Path
---------- ---------- ---------- ---------- ----
      1185 0x4b883000 0x4b884000     0x1000 /root/Downloads/task.1185.0x4b883000.vma
```

Εικόνα 25. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dump_map -p 1185 --dump-dir /root/Downloads -s 0x000000004b883000

Στα αρχεία που κάναμε μόλις dump, θα τρέξουμε την εντολή strings η οποία χρησιμοποιείται για να εξάγουμε τις παραμέτρους των εκτυπώσιμων(printables) χαρακτήρων μέσα από ένα φάκελο ή ένα συγκεκριμένο αρχείο.

```
root@kali:~/volatility# strings -a /root/Downloads/task.1185.0x4b883000.vma
S7 D
@~ow
META-INF/MANIFEST.MF
META-INF/maven/com.metasploit/Metasploit-JavaPayload/pom.propertiesPK
META-INF/maven/com.metasploit/Metasploit-JavaPayload/pom.xmlPK
classes.dexPK
```

Εικόνα 26. Αποτέλεσμα εντολής strings -a /root/Downloads/task.1185.0x4b883000.vma

```
root@kali:~/volatility# strings -a /root/Downloads/task.1185.0x49b80000.vma
META-INF/MANIFEST.MF
LK-.
:UZ)
classes.dexm
nRJ)4.mS(N
P%.p@B
`bMs
y3^O,'A%
IVs-
$h?_
z0[V
lb i
Fc$M
^vj,
Sr<R
qD>3
5if@
META-INF/MANIFEST.MF
classes.dexPK
```

Εικόνα 27. Αποτέλεσμα εντολής strings -a /root/Downloads/task.1185.0x49b80000.vma

Όπως φαίνεται στην εικόνα 26, εντοπίζουμε την ακολουθία Metasploit-JavaPayload οπότε μπορούμε με ασφάλεια να συμπεράνουμε ότι το κακόβουλο αρχείο που χρησιμοποιήθηκε από την συγκεκριμένη διεργασία είναι το Metasploit-JavaPayload. Σε περίπτωση που θέλαμε να εκτελέσουμε την εντολή για όλο το φάκελο και όχι για κάθε ένα αρχείο ξεχωριστά, το αποτέλεσμα της εντολής θα ήταν το κάτωθι που ουσιαστικά περιλαμβάνει συγκεντρωμένη την πληροφορία και για τα δύο υπό εξέταση αρχεία.

```
root@kali:~/volatility# strings -a /root/Downloads/*.vma
META-INF/MANIFEST.MF
LK-.
:UZ)
classes.dexm
nRJ)4.mS(N
P%.p@B
`bMs
y3^O,'A%
IVs-
$h?_
z0[V
lb i
Fc$M
^vj,
Sr<R
qD>3
5if@
META-INF/MANIFEST.MF
classes.dexPK
S7 D
@~ow
META-INF/MANIFEST.MF
META-INF/maven/com.metasploit/Metasploit-JavaPayload/pom.propertiesPK
META-INF/maven/com.metasploit/Metasploit-JavaPayload/pom.xmlPK
classes.dexPK
```

Εικόνα 28. Αποτέλεσμα εντολής strings -a /root/Downloads/*.vma

```
root@kali:~/volatility# python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM
 linux_dentry_cache | grep httpmon | grep '.apk'
Volatility Foundation Volatility Framework 2.6
0|dalvik-cache/data@app@org.jtb.httpmon-1.apk@classes.dex|1094|0|1000|10061|102144|40842658
00|4084265808|0|4084265816
0|app/org.jtb.httpmon-1.apk|1063|0|1000|1000|124408|4084268760|4084268768|0|4084268776
0|Download/[Megafileupload]org.jtb.httpmon.apk|174|0|1000|1015|124408|4084033896|4084033904
|0|4084033912
```

Εικόνα 29. Αποτέλεσμα εντολής python vol.py -f /root/memory.dmp --profile=Linuxgoldfish-2_6_29ARM linux_dentry_cache | grep httpmon | grep '.apk'

Χρήσιμο ενδεχομένως θα ήταν να ανακτήσουμε και το μονοπάτι στο οποίο έγινε το κατέβασμα και η αποθήκευση του κακόβουλου αρχείου. Βλέποντας στην εικόνα 29 λοιπόν, παρατηρούμε ότι το κακόβουλο αρχείο είχε αποθηκευτεί στην ακόλουθη τοποθεσία: *«Download/[Megafileupload]org.jtb.httpmon.apk».*

Με τις εντολές που εκτελέστηκαν για να αναλύσουμε την μνήμη μας λοιπόν, έχουμε εξάγει αρκετά χρήσιμα συμπεράσματα τα οποία μας βοήθησαν να κατανοήσουμε τις κακόβουλες διεργασίες που έτρεχαν πάνω στη συσκευή κατά τη στιγμή της απόκτησης, το κακόβουλο λογισμικό από το οποίο είχε μολυνθεί η υπό εξέταση συσκευή καθώς και την σύνδεση μεταξύ τους.

Σημείωση: Κατά την διαδικασία ανάλυσης της μνήμης, υπήρξαν εντολές οι οποίες χρησιμοποιήθηκαν αλλά τα αποτελέσματα που επέστρεψαν δεν ήταν χρήσιμα ή βοηθητικά στην εξέταση μας. Τα screenshots από τις συγκεκριμένες εντολές δεν παρατίθενται.

This page intentionally left blank

8. Επίλογος

8.1 Σύνοψη

Συνοψίζοντας την διπλωματική εργασία, προσπαθήσαμε να προσεγγίσουμε και να αναλύσουμε αρκετούς τομείς που αποτελούν την «Ψηφιακή Εγκληματολογία μνήμης σε συσκευές Android» τόσο από την θεωρητική, όσο και από την πρακτική τους όψη εστιάζοντας στο πρακτικό κομμάτι στην ανάλυση της ζωντανής μνήμης από μία συσκευή Android.

Αρχικά, στο θεωρητικό μέρος έγινε ανάλυση του περιβάλλοντος Android (λειτουργικό σύστημα, αρχιτεκτονική, σύστημα αρχείων κτλ.) και στη συνέχεια αναλύθηκαν οι τρόποι εξαγωγής των δεδομένων από κινητές συσκευές τύπου Android. Στη συνέχεια, περιεγράφηκαν αναλυτικά τα βήματα που ακολουθούνται για την πραγματοποίηση μίας ψηφιακής εξέτασης κινητής συσκευής. Έπειτα, έγινε περιγραφή και ανάλυση διάφορων εμπορικών αλλά και εργαλείων ανοιχτού κώδικα που χρησιμοποιούνται στην ψηφιακή εγκληματολογία και αναφέρθηκαν επιγραμματικά οι πληροφορίες που μπορούν να ανακτηθούν με την χρήση τους. Προχωρώντας στην εργασία, πραγματοποιήθηκε εκτενέστερη αναφορά στην ψηφιακή εγκληματολογία μνήμης και αναφέρθηκαν οι διαφορές μεταξύ «ζωντανής(live)» και «νεκρής(dead)» ψηφιακής εγκληματολογίας, καθώς και τυχόν προβλήματα/προκλήσεις που συνεπάγεται η ψηφιακή εγκληματολογία ζωντανής μνήμης σε περιβάλλον Android. Όλα αυτά ακολουθήθηκαν από μία πρακτική εξέταση δείγματος μνήμης από συσκευή Android, δείχνοντας στην πράξη τι είδους πληροφορίες είναι εφικτό να εξάγουμε από την ανάλυση της ζωντανής μνήμης μίας συσκευής.

8.2 Συμπεράσματα

Αυτό που θέλει να προσφέρει η συγκεκριμένη διατριβή στον αναγνώστη, είναι αρχικά να τον βοηθήσει να αποκτήσει μία σφαιρική εικόνα για την ψηφιακή εγκληματολογία μνήμης σε συσκευές Android, βοηθώντας τον να κατανοήσει τόσο τις διαφορετικές μεθόδους εξαγωγής δεδομένων από τις κινητές συσκευές, όσο και την πληθώρα εργαλείων που υπάρχουν στην αγορά, για να μπορεί να καταλήξει ανάλογα με τον τύπο εξέτασης αλλά και τα δεδομένα που θέλει να εξάγει στο κατάλληλο.

Ο κλάδος της ψηφιακής εγκληματολογίας ζωντανής μνήμης σε κινητές συσκευές Android είναι ιδιαίτερα δύσκολος και απαιτητικός και σε πραγματικές συνθήκες, αρκετές φορές η εξέταση της ζωντανής μνήμης είναι ανέφικτη. Αυτό οφείλεται στα προαπαιτούμενα που πρέπει να πληροί η υπό εξέταση συσκευή, κάτι το οποίο καθιστά τον τύπο εξέτασης σχεδόν απαγορευτικό για τον ψηφιακό εξεταστή. Παρόλο λοιπόν που οι πληροφορίες που μπορεί ο ψηφιακός εξεταστής να ανακτήσει από την ζωντανή μνήμη μίας συσκευής είναι πολύ σημαντικές και υπό άλλες συνθήκες μη ανακτήσιμες, δυστυχώς τις περισσότερες φορές ο ψηφιακός εξεταστής θα αναγκαστεί να παρακάμψει τελείως την εξέταση αυτού του τύπου και απλά θα αρκεστεί στις πληροφορίες που μπορεί να ανακτήσει με την ανάλυση της «νεκρής» μνήμης.

Κάτι το οποίο δυσκόλεψε αρκετά το πρακτικό κομμάτι της συγκεκριμένης διατριβής, είναι η έλλειψη λεπτομερούς documentation και συγκεκριμένων βημάτων που απαιτούνται, προκειμένου ο ψηφιακός εξεταστής να εξάγει επιτυχώς την «ζωντανή» μνήμη από μία συσκευή

με την χρήση του εργαλείου LiME. Αποτέλεσμα αυτού, ήταν η διατριβή να επικεντρωθεί κατά βάση στο κομμάτι της ανάλυσης της μνήμης και όχι στον τρόπο με τον οποίο η μνήμη αποκτάται από την υπό εξέταση συσκευή. Ίσως αυτό να είναι κάτι το οποίο χρήζει βελτίωσης από την μεριά του προγραμματιστή αλλά και της κοινότητας ανοιχτού λογισμικού.

Ελπίζουμε στο μέλλον, να βρεθούν ευκολότεροι τρόποι εξαγωγής της «ζωντανής» μνήμης από τις φορητές συσκευές προκειμένου να καταφέρει ο συγκεκριμένος τρόπος εξέτασης να γίνει ευρέως αποδεκτός, γιατί οι πληροφορίες που μπορούν να ανακτηθούν με την χρήση της συγκεκριμένη μεθόδου είναι πολύτιμες και κρίσιμες για την ορθότερη και αποτελεσματικότερη εξέταση της ψηφιακής εγκληματολογίας μίας συσκευής.

9. Βιβλιογραφία – Πηγές

Michael Hale Ligh, Andrew Case, Jamie Levy, Aaron Walters (2014), *The Art of Memory Forensics*: *Detecting Malware and Threats in Windows, Linux, and Mac Memory*

Mamoona Rafique, M.N.A.Khan (2013), *Exploring Static and Live Digital Forensics: Methods, Practices and Tools*. International Journal of Scientific & Engineering Research Volume 4, Issue 10, October-2013 ISSN 2229-5518

Johannes Stuettgen and Michael Cohen Robust (2014), *Linux Memory Acquisition with Minimal Target Impact*

Andri P. Heriyanto (2013), *Procedures and Tools For Acquisition And Analysis Of Volatile Memory On Android Smartphones*, Edith Cowan University

Andri Heriyanto, Craig Valli, Peter Hannay (2015), *Comparison of Live Response, Linux Memory Extractor (LiME) and Mem tool for acquiring android's volatile memory in the malware incident*, Edith Cowan University

Holger Macht (2013), *Live Memory Forensics on Android with Volatility*, Diploma Thesis in Department of Computer Science Friedrich-Alexander University Erlangen-Nuremberg

Vrizlynn Thing, Kian-Yong Ng and Ee-Chien Chang (2010), *Live Memory Forensics of Mobile Phones*

Dimitris Apostolopoulos, Giannis Marinakis, Christoforos Ntantogian, Christos Xenakis (2013), *Discovering authentication credentials in volatile memory of Android mobile devices*

Rushita Dave, Nilay R. Mistry, Dr. M. S.Dahiya (2014), *Volatile Memory Based Forensic Artifacts & Analysis Institute of Forensic Science*, Gujarat Forensic Sciences University, Gandhinagar, Gujarat - India

Venkateswara Rao V., A. S. N. Chakravarthy (2016), *Survey on Android Forensic Tools and Methodologies*

Firdous Kausar and Tadani Nasser Alyahya (2016), *Analysis of Physical Image Acquisition Forensic Tools for Android Smartphones*, IJCSNS International Journal of Computer Science and Network Security, VOL.16 No.11, November 2016

Marthie Lessing, Basie von Solms (2008), *Live Forensic Acquisition as Alternative to Traditional Forensic Processes*

Mahesh Kolhe, Purnima Ahirao (2017) *Live Vs Dead Computer Forensic Image Acquisition*, Mahesh Kolhe et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (3), 2017, 455-457

Jeff Lessard, Gary C. Kessler (2010), *Android Forensics: Simplifying Cell Phone Examinations*, SMALL SCALE DIGITAL DEVICE FORENSICS JOURNAL VOL. 4, NO.1, SEPTEMBER 2010, ISSN# 1941-6164

Det. Cynthia A. Murphy (2013), *Developing Process for Mobile Device Forensics*

Simon Broenner, Hans Höfken and Marko Schuba (2016), *Streamlining Extraction and Analysis of Android RAM Images*, Department of Electrical Engineering and Information Technology, Aachen University of Applied Sciences

ISAK MRKAIĆ (2016), *Android Forensic Using Some Open Source Tools*, University of Donja Gorica

https://digital-forensics.sans.org/summit-archives/2012/android-mind-reading-memory-acquisition-and-analysis-with-lime-and-volatility.pdf

https://www.sans.org/reading-room/whitepapers/forensics/forensic-analysis-android-practical-case-36317

http://www.ucd.ie/cci/cync/Acquisition%20and%20Analysis%20of%20Android%20Memory.pdf

https://www.computersecuritystudent.com/HOME/index.html

http://gowrishankarnath.com/wp-content/uploads/2015/12/android-Architecture.png

http://www.idc.com/promo/smartphone-market-share/os

http://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/the-mobile-forensics-process-steps-types/

https://en.wikipedia.org/wiki/Mobile_device_forensics

http://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/common-mobile-forensics-tools-and-techniques/

https://securitycommunity.tcs.com/infosecsoapbox/articles/2015/11/04/mobile-forensics-data-acquisition-methods

https://www.imf-conference.org/imf2008/IMF2008-06_Live%20Forensic%20Acquisition%20as%20Alternative%20to%20Traditional%20Forensics%20-%20Marthie%20Lessing.pdf

https://digital-forensics.sans.org/media/DFIR-Smartphone-Forensics-Poster.pdf

http://www.uobabylon.edu.iq/eprints/publication_4_13681_1356.pdf

http://cradpdf.drdc-rddc.gc.ca/PDFS/unc160/p800486_A1b.pdf

http://opensecuritytraining.info/AndroidForensics_files/Android%20Forensics%20and%20Security%20Testing%20Course_PUBLIC_RELEASE.pptx.pdf

https://www.nist.gov/sites/default/files/documents/forensics/6-Mahalik_OSMF.pdf

https://www.addictivetips.com/mobile/android-partitions-explained-boot-system-recovery-data-cache-misc/

https://comserv.cs.ut.ee/home/files/pappas_cybersecurity_2017.pdf?study=ATILoputoo&reference=BE2138E95B31179324FF14E71176FCDB482D24DD

https://www.cvedetails.com/top-50-products.php?year=0

https://www.cvedetails.com/top-50-products.php?year=2016

https://www.cvedetails.com/top-50-products.php?year=2017

https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224

https://developer.android.com/about/dashboards/index.html

https://en.wikipedia.org/wiki/Open_Handset_Alliance

https://www.techopedia.com/definition/27805/digital-forensics

https://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%AE%CE%BD%CE%B1%CF%82_Linux

https://developer.android.com/guide/platform/index.html

https://developer.android.com/guide/platform/images/android-stack_2x.png

https://en.wikipedia.org/wiki/Android_version_history

http://www.slideshare.net/nikoskapios/live-memory-forensics-on-android-devices

http://www.malware-analyzer.com/memory-forensics

https://www.certsi.es/en/blog/mobile-forensic-analyses-tools

http://forensicswiki.org/wiki/Tools:Memory_Imaging

https://www.oxygen-forensic.com/en/compare

https://media.cellebrite.com/wp-content/uploads/2018/04/UFEDPA7.3_ReleaseNotes.pdf

http://www.signalsec.com/saft/

https://www.msab.com/products/xry/

http://www.rekall-forensic.com/

https://www.digitalforensics.com/blog/android-forensic-analysis-with-autopsy/

https://www.andriller.com/

https://belkasoft.com/ec

https://github.com/google/rekall

http://www.mobiledit.com/forensic-express