ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ «Ασφάλεια Ψηφιακών Συστημάτων»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΣΥΓΚΡΙΤΙΚΗ ΨΗΦΙΑΚΗ ΔΙΚΑΝΙΚΗ ΑΝΑΛΥΣΗ ΤΩΝ WINDOWS 7,8,10»

Σπουδαστής:

ΤΕΜΠΕΛΗΣ ΙΩΑΝΝΗΣ ΜΤΕ1538

Επιβλέπων Καθηγητής:

Δρ. ΝΤΑΝΤΟΓΙΑΝ ΧΡΙΣΤΟΦΟΡΟΣ

ΦΕΒΡΟΥΑΡΙΟΣ 2018

ΠΕΡΙΛΗΨΗ

Στη παρούσα εργασία παρουσιάζονται οι διαφορετικές τεχνολογικές μέθοδοι που χρησιμοποιούνται από τις εκδόσεις των Windows 7, 8 (8.1) και 10 από την σκοπιά των ψηφιακών πειστηρίων. Βλέπουμε πως τροποποιήθηκαν οι τεχνολογίες ξεκινώντας από την έκδοση 7 και ποια καινούρια στοιχεία και δυνατότητες ενσωματώθηκαν στις επόμενες εκδόσεις των Windows, από την 8 (8.1) έως και τη 10. Επίσης εξετάζουμε πως αυτές οι αλλαγές επηρεάζουν τον αναλυτή ψηφιακών πειστηρίων όσον αφορά τη συλλογή και αξιολόγηση της πληροφορίας.

Λέξεις κλειδιά: Ασφάλεια, Windows, ψηφιακά πειστήρια

Στον πατέρα μου Σταύρο† (5.4.2018)

Περιεχόμενα

1. Forensics Artifacts

Το λειτουργικό σύστημα Microsoft Windows είναι ίσως το πιο μελετημένο λειτουργικό σύστημα από τη σκοπιά των ψηφιακών πειστηρίων (digital forensics). Σε όλα τις εκδόσεις Windows, το σύστημα δημιουργεί και φυλάσσει διαφόρων τύπων δεδομένα, τα οποία σχετίζονται με τη δραστηριότητα του χρήστη στο λειτουργικό σύστημα. Οι περισσότερες έρευνες ψηφιακών πειστηρίων ασχολούνται με κυρίως με δεδομένα τα οποία παράγονται από το χρήστη, δηλαδή δημιουργηθέντα αρχεία, διαγραμμένα ή/και κρυπτογραφημένα αρχεία και όχι τόσο με αντικείμενα (artifacts) τα οποία δημιουργούνται από το ίδιο το λειτουργικό σύστημα ως συνέπεια των ενεργειών του χρήστη.

Τα artifacts είναι αντικείμενα σε ένα λειτουργικό σύστημα τα οποία περιέχουν πληροφορία σχετική με τις δραστηριότητες του χρήστη. Η τοποθεσία και ο τύπος της πληροφορίας η οποία περιέχεται σε αυτά διαφέρει μεταξύ των διαφορετικών λειτουργικών συστημάτων. Η μελέτη αυτών έγκειται στο να προσδιοριστούν, να υποστούν την κατάλληλη επεξεργασία και να αναλυθούν ώστε να επικυρώσουν ή να αποδείξουν μια παρατήρηση κατά τη διάρκεια μια ψηφιακής ανάλυσης. Παρ' όλα αυτά, απουσία της πληροφορίας σε ένα συγκεκριμένο artifact δεν σημαίνει ότι δεν υπήρχε ανθρώπινη δραστηριότητα στο σύστημα. Υπάρχουν αρκετά artifacts τα οποία συνδυαζόμενα μπορούν να παρέχουν σημαντική γνώση για το τι επιτελέστηκε σε ένα υπολογιστικό σύστημα. Ο τύπος και η τοποθεσία αυτών μπορεί να διαφέρει ανάμεσα σε διαφορετικές εκδόσεις του ίδιου λειτουργικού συστήματος.

Παρακάτω παρουσιάζεται μια ενδεικτική λίστα από artifacts τα οποία αξίζουν προσοχής και μελέτης για τις εκδόσεις Windows 7, 8 (8.1) και 10.

- Registry
- Windows Event logs
- Volume shadow copies
- MFT (NTFS file system)
- Shellbags
- Jump lists
- USNJRNL
- User-assist keys
- Prefetch files
- Link files

- Bitlocker
- USB activity

1.1 Registry

Η registry παρέχει πάρα πολλές πληροφορίες για έναν αναλυτή ψηφιακών πειστηρίων. Είναι μια ιεραρχική βάση δεδομένων, η οποία περιέχει ρυθμίσεις συστήματος οι οποίες ορίζονται είτε από το ίδιο το σύστημα είτε από το χρήστη. Είναι ένα αποθετήριο, το οποίο καταγράφει τις δραστηριότητες του χρήστη με τη μορφή registry εγγραφών. Στα Windows είναι οργανωμένη σε Hives, Keys και Sub-Keys το οποία περιέχουν κάποιες τιμές. Μπορεί να αναδομηθεί off-line από τα hive αρχεία, όπως "SYSTEM, SOFTWARE, SAM, SECURITY, USERDIFF" τα οποία βρίσκονται στο "C:\Windows\System32".

Πληροφορίες όπως το όνομα του υπολογιστή, ημερομηνίες εγκατάστασης λειτουργικού συστήματος, προγράμματα που εγκαταστάθηκαν/απεγκαταστάθηκαν, συσκευές USB που είναι συνδεδεμένες, αρχεία που προσπελάστηκαν πρόσφατα, IP διευθύνσεις, πληροφορία που διαμοιράζεται δικτυακά, πληροφορία σχετικά με τη δραστηριότητα του χρήστη και τη δραστηριότητα κάποιου malware, μπορούν να αντληθούν από την ανάλυση της registry. Προς τούτο μπορούν να χρησιμοποιηθούν εργαλεία όπως το Regripper, regview, Mitec Registry Viewer, αλλά και πολλά εμπορικά διαθέσιμα, τα οποία μπορούν να κάνουν parse τα αρχεία registry και να οπτικοποιήσουν την πληροφορία που περιέχεται σε αυτά.

1.2 Windows Event Logs

Αυτά τα αρχεία καταγράφουν τις ειδοποιήσεις και τις αφυπνίσεις που παράγονται ως αποτέλεσμα της δραστηριότητας του χρήστη. Αναφέρονται ως "events" και προσδιορίζονται από ένα event ID. Αυτά μπορούν να προβληθούν από μια ενσωματωμένη στα Windows εφαρμογή, το event viewer ενώ το σύστημα είναι ενεργό. Σε offline κατάσταση (π.χ. μελέτη ενός image) αυτά τα αρχεία βρίσκονται στο "C:\Windows\System32\winevt\Logs". Τα πιο κοινά αρχεία που εξετάζονται είναι τα SecEvent.evtx, SysEvent.evtx, AppEvent.evtx. Η πληροφορία που μπορεί να αποκτηθεί από αυτά είναι login/logout οτυ χρήστη, δραστηριότητα εφαρμογών και χρήστη, αλλαγές σε προκαθορισμένες ρυθμίσεις, αλλαγές που έχουν γίνει από malware.

1.3 Volume Shadow Copies

Είναι τα αντίγραφα ασφαλείας (snapshots), τα οποία περιοδικά δημιουργούνται για κάθε μονάδα δίσκου από το Volume Shadow Copy service. Αποτελούν κατά κάποιο τρόπο ένα ιστορικό και περιέχουν χρήσιμη πληροφορία για τα αρχεία και τα δεδομένα. Εξετάζοντας τέτοια αντίγραφα ασφαλείας μπορούμε να πάρουμε πληροφορίες για προηγούμενες εκδόσεις αρχείων, φακέλων του σκληρού δίσκου. Εκδόσεις αυτών των αντιγράφων βρίσκονται σε registry hives και βάσεις δεδομένων SQL-lite. Εργαλεία όπως το shadow explorer ή το Internet Evidence Finder μπορούν να κάνουν mount αυτά τα ατνίγραφα για περαιτέρω ανάλυση

1.4 Master File Table (MFT)

Είναι μια βάση δεδομένων του NTFS η οποία διατηρεί πληροφορία για τα αρχεία και τους φακέλους στη μονάδα του δίσκου. Μπορούμε να αντλήσουμε πληροφορίες για το μέγεθος των αρχείων, το path, τα διεγραμμένα αρχεία κλπ. Κατά την ανάλυση malware μπορούν να χρησιμεύσουν για να εντοπιστούν IOCs σχετικά με τη δραστηριότητα του malware. Εφαρμογές όπως το FTK ή το X-ways και εργαλεία όπως το MFT ripper, MFTDump ή το AnalyzeMFT μπορούν να κάνουν parse το MFT και να οπτικοποιήσουν το περιεχόμενό του στον αναλυτή ψηφιακών πειστηρίων.

1.5 Windows Shellbags

Περιέχουν πληροφορίες για τις θεάσεις, το μέγεθος, το σχήμα, τα εικονίδια και τη θέση των παραθύρων του windows explorer. Η εξέταση των shellbags μπορεί να παρέχει πληροφορία για τους φακέλους, το ιστορικό θέασης του χρήστη, λεπτομέρειες για φακέλους που έχουν διαγραφεί. Τα shellbags είναι οργανωμένα στο BagMRU key στη registry και το που είναι αυτό στη registry εξαρτάται από την έκδοση των Windows. Λόγω του ότι περιέχουν πληροφορία ακόμα και για φακέλους που έχουν διαγραφεί, η ανάλυση και το parsing αυτών των artifacts μπορεί να χρησιμοποιηθεί για να αναδομηθεί η δραστηριότητα του χρήστη.

1.6 Jump lists

Οι Jumplists είναι μια από τις καινοτομίες των Windows 7 και βοηθά το χρήστη να δει όλα τα πρόσφατα χρησιμοποιημένα αρχεία βάσει της κατηγορίας αρχείων. Επίσης επιτρέπει στο χρήστη να κάνει pin τα αρχεία στην taskbar για πιο εύκολη πρόσβαση σε αυτά. Οι Jumplists είναι σε μορφή "*.automaticDestinations-ms" στο path του χρήστη C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations" και σε αρχεία "*.customDestination-ms" σε μορφή binary. Ο αναλυτής μπορεί μέσω αυτών να προσδιορίσει ποια αρχεία και ποιες εφαρμογές δημιούργησε και χρησιμοποίησε ο χρήστης.

1.7 USNJRNL

Το αρχείο USNJRNL (Update Sequence Number Journal), γνωστό και ως NTFS Change Journal φυλάσσει όλες τις αλλαγές για ένα αρχείο. Τέτοιο αρχείο διατηρεί κάθε NTFS volume και αποθηκεύεται στο αρχείο "$Extend\$UsnJrnl". Αυτό το αρχείο μπορεί να φανεί χρήσιμο σε έναν αναλυτή για να ανακαλύψει τις αλλαγές που έχουν γίνει στο λειτουργικό σύστημα. Η εξέταση λοιπόν αυτού του αρχείου μπορεί να αποκαλύψει ονόματα αρχείων και φακέλων, τα αντίστοιχα MFT record numbers, την τροποποίηση που έχουν υποστεί τα αρχεία, χρονική στιγμή αυτών των τροποποιήσεων, αιτιολογία που αυτές έγιναν, το Security ID και πως προέκυψαν οι τροποποιήσεις αυτές.

1.8 Userassist keys

Είναι ένα σύνολο από κλειδιά της registry το οποία περιέχουν πληροφορία για τις εφαρμογές και τις συντομέυσεις οι οποίες προσπελάστηκαν από το χρήστη μέσω του γραφικού περιβάλλοντος των Windows. Περιέχεται επίσης πληροφορία για το πλήθος των εκτελέσεων, την ημερομηνία τελευταίας εκτέλεσης, επομένως γνωρίζουμε αν κάποιος έκανε μια ενέργεια ή όχι. Τα κλειδιά αυτά βρίσκονται στο path "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist".

1.9 Prefetch (Superfetch) files

Αυτά τα αρχεία δημιουργούνται από το ίδιο το λειτουργικό σύστημα κάθε φορά που μια εφαρμογή εκτελείται από μια συγκεκριμένη τοποθεσία για πρώτη φορά. Χρησιμεύουν για επιταχύνουν τη διαδικασία εκκίνησης μιας εφαρμογής. Αυτά τα αρχεία λαμβάνουν προκαθορισμένη ονομασία και μορφή και το όνομά τους αποτελείται από το όνομα της εφαρμογής, ένα hash που υποδεικνύει την τοποθεσία από την οποία αυτή εκτελέστηκε και μια επέκταση αρχείου ".PF". Είναι αποθηκευμένα στο φάκελο "C:\Windows\Prefetch". Το αξιοσημείωτο είναι ότι τα αρχεία αυτά διατηρούνται ακόμα κι όταν η εφαρμογή διαγράφτηκε ή απεγκαταστάθηκε. Σε συνδυασμό με μια χρονική ανάλυση συμβάντων, ο αναλυτής προσδιορίζει ποιες εφαρμογές εκτελέστηκαν στο σύστημα.

1.10 Link files

Τα Link files είναι οι γνωστές συντομεύσεις και δημιουργούνται τόσο από το χρήστη όσο και από το ίδιο το λειτουργικό σύστημα, για αρχεία που χρησιμοποιούνται συχνά ή προσπελαύνονται από διαφορετική τοποθεσία π.χ. αφαιρούμενες αποθηκευτικές μονάδες, network shares. Έτσι επικυρώνεται η ύπαρξη των αρχείων τα οποία προσπέλασε ο χρήστης και τα metadata περιέχουν πληροφορία δημιουργίας, προσπέλασης, τροποποίησης και τις αντίστοιχες χρονικές στιγμές του πρωτότυπου αρχείου, πληροφορία για το δίσκο του συστήματος στον οποίο ήταν το αρχείο (π.χ. volume serial number), πληροφορίες δικτύου αν ήταν δικτυακά διαμοιραζόμενα, μέγεθος αρχείου κ.ά.. Πληθώρα εργαλείων μπορεί να κάνει parse αυτά τα αρχεία.

1.11 USB Activity

Οι φορητές αποθηκευτικές συσκευές είναι μια από τις βασικότερες απειλές ασφάλειας. Αυτές οι συσκευές μπορούν να συνδεθούν σε ένα σύστημα για να εκτελέσουν κακόβουλες ενέργειες. Ανάμεσα σε αυτές τις ενέργειες μπορεί να είναι κλοπή προσωπικών και εταιρικών δεδομένων, μεταφορά εμπιστευτικών δεδομένων, διασπορά κακόβουλου λογισμικού, κ.ά.. Τα οπτικά μέσα αποθήκευσης και οι δισκέτες έχουν αντικατασταθεί από USB drives, τα οποία, ενώ είναι αρκετά μικρά σε μέγεθος, παρέχουν μεγάλη

χωρητικότητα. Οι έρευνες ψηφιακών πειστηρίων γύρω από τις συσκευές USB στηρίζονται σε δύο άξονες. Ο πρώτος είναι να αναλυθεί η registry ώστε να βρεθούν «υπολείμματα» από τη δραστηριότητα του USB και ο δεύτερος είναι να βρεθούν ίχνη από όλες τις συσκευές που έχουν συνδεθεί στο σύστημα.

2. Recycle Bin

Όπως είναι γνωστό, ο κάδος ανακύκλωσης (Recycle Bin) είναι ένα εικονίδιο στην επιφάνεια εργασίας των Windows, ο οποίος αποθηκεύει προσωρινά διαγραμμένα αρχεία και μπορεί να ανακτήσει αυτά που μπορεί να διαγράφησαν κατά λάθος. Τα αρχεία που φυλάσσει είναι αυτά που διαγράφησαν από το windows explorer. Από προκαθορισμένη ρύθμιση, τα διαγραμμένα αρχεία που βρίσκονται σε αφαιρούμενα ή δικτυακά μέσα αποθήκευσης διαγράφονται μόνιμα. Τα αρχεία τα οποία βρίσκονται σε εσωτερικούς σκληρούς δίσκους, όταν διαγράφονται από τη γραμμή εντολών ή έχοντας πατημένο το πλήκτρο Shift διαγράφονται μόνιμα, χωρίς να αποθηκεύονται στον κάδο ανακύκλωσης. Μόνιμη διαγραφή σημαίνει για έναν αναλυτή ότι αυτός μπορεί να ανακτήσει το περιεχόμενό τους χρησιμοποιώντας εργαλεία data carving, όσο η περιοχή της μνήμης που διατηρούσε το αρχείο δεν έχει χρησιμοποιηθεί για άλλο αρχείο. Ο κάδος ανακύκλωσης αποτελεί σημαντική πηγή πληροφοριών για έναν αναλυτή. Αυτός μπορεί να δεί μέσα στον κάδο ανακύκλωσης τα αρχεία με το αρχικό τους όνομα και όταν αυτά επαναφέρονται επιστρέφουν στην αρχική τους τοποθεσία με το ίδιο πρωταρχικό όνομα. [1]

2.1 Αρχιτεκτονική

Η πραγματική τοποθεσία του κάδου ανακύκλωσης εξαρτάται από την έκδοση του λειτουργικού συστήματος και το σύστημα αρχείων. Στα windows 7,8 (8.1) η τοποθεσία είναι στο “C:\$Recycle.Bin”, η οποία είναι κρυφή στον τελικό χρήστη. Αυτός ο φάκελος βρίσκεται σε κάθε λογική μονάδα δίσκου στο σύστημα. Στο τμήμα δίσκου του συστήματος π.χ. C:\, ο φάκελος του κάδου ανακύκλωσης είναι $Recycle.Bin (μικρά ή κεφαλαία γράμματα), ενώ στα άλλα partitions εμφανίζεται ως $RECYCLE.BIN. Για τις εκδόσεις πριν τα Windows 7, ένα αρχείο που βρίσκεται στον κάδο ανακύκλωσης φυλάσσεται στη φυσική του τοποθεσία και μετονομάζεται ως D<αρχικό drive letter του αρχείου><#>.<αρχική επέκταση αρχείου>. Για παράδειγμα , αν διαγράψουμε ένα αρχείο με όνομα Readme.txt από το drive C:\ αυτό θα μετονομαστεί ως DC1.txt. Ένα κρυφό αρχείο INFO2 περιέχει τον αρχικό κατάλογο του αρχείου και το αρχικό όνομά του σε μορφή binary.

Στις επόμενες εκδόσεις των Windows, όταν ένα αρχείο διαγράφεται και το σύστημα αρχείων είναι NTFS η διαδικασία είναι διαφορετική. Η εγγραφή $MFT ενημερώνεται με

έναν καινούριο αριθμό εγγραφής για το parent folder (το οποίο τώρα είναι ο κάδος ανακύκλωσης αντί του αρχικού). Έπειτα το αρχείο αποκτά ένα νέο όνομα από έξι τυχαίους χαρακτήρες και την αρχική επέκτασή του. Τώρα το αρχείο μετονομάζεται ως $R<αλφαριθμητικό>.<αρχική επέκταση αρχείου> και μαζί με αυτό δημιουργείται ένα συνοδευτικό διαχειριστικό αρχείο με τους ίδιους έξι χαρακτήρες μαζί με την αρχική επέκταση ως $I<αλφαριθμητικό>.<αρχική επέκταση αρχείου> [2]. Για παράδειγμα, αν διαγραφεί το αρχείο Readme.txt, θα δημιουργηθούν δύο αρχεία, έστω το $RPRLG5.txt και το $IPRLG5.txt. Αυτοί οι δύο υποφάκελοι καθιστούν τον κάδο ανακύκλωσης και διαχωρίζονται ο ένας από τον άλλο με βάση το χρήση στον οποίο ανήκουν. Η ονομασία αυτών των φακέλων βασίζεται στο Security Identifier (SID) του χρήστη και είναι σημαντικό να αντιστοιχίζεται το SID με συγκεκριμένο χρήστη. Το SID είναι μια αλφαριθμητική συμβολοσειρά, η οποία χρησιμοποιείται από τα Windows για καθορίσει ένα συγκεκριμένο αντικείμενο, όπως έναν χρήστη ή μια ομάδα χρηστών. Αν και το προφίλ του χρήστη έχει δημιουργηθεί αρχικά, το SID δεν εμφανίζεται έως ότου ο χρήστης συνδεθεί στο σύστημα για πρώτη φορά.

Τα αρχεία που διαγράφονται από άλλους χρήστες στο ίδιο σύστημα δεν τοποθετούνται στον ίδιο κάδο ανακύκλωσης. Κάθε χρήστης έχει τον δικό του κάδο ανακύκλωσης. Αυτό γίνεται γιατί όταν ο χρήστης διαγράφει ένα αρχείο, αυτό εμφανίζεται με το SID του χρήστη. Κάθε χρήστης έχει το δικό του κάδο σε κάθε λογική μονάδα δίσκου. Για παράδειγμα, αν δύο χρήστες χρησιμοποιούν ένα σύστημα με πέντε partitions, κάθε partition έχει δύο υποφακέλους με δύο διαφορετικά SIDs, άρα συνολικά δέκα κάδοι ανακύκλωσης.

Σχήμα 1. Περιεχόμενα κάδου ανακύκλωσης

Στο παραπάνω σχήμα το γράμμα S υποδεικνύει ότι η συμβολοσειρά είναι ένα SID, το 1 είναι το Revision Level, το 5 είναι το Identifier Authority Level (στο συγκεκριμένο παράδειγμα το 5 αντιστοιχεί στο NT Authority). Το υπόλοιπο της συμβολοσειράς από το 21-5100…..693 είναι το Domain ή το Local Computer Identifier. Είναι μια μεταβλητή η

οποία καθορίζει ποιο τερματικό ή δίκτυο δημιούργησε αυτό το νούμερο. Στο τέλος, το 500 είναι το Relative ID κι εδώ είναι ο administrator. Οι προκαθορισμένες ρυθμίσεις επιτρέπουν μόνο έναν λογαριασμό να είναι ο administrator. Οι λογαριασμοί χρηστών ή ομάδων χρηστών που δεν είναι προκαθορισμένοι παίρνουν τιμή Relative ID από 1000 και πάνω. Στο σχήμα βλέπουμε ότι έχουν δημιουργηθεί δύο τέτοιοι λογαριασμοί αφού οι αντίστοιχες τιμές είναι 1000 και 1001. [5]

Είναι πολύ σημαντικό για τον αναλυτή να αντιστοιχίσει το SID με τον χρήστη. Τρέχουμε λοιπόν την εφαρμογή regedit και πλοηγούμαστε στο HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\ProfileList. Από εκεί επιλέγουμε το φάκελο που το όνομά του αντιστοιχεί στο SID του χρήστη που ψάχνουμε. Δεξιά στο παράθυρο θα δούμε μια λίστα από αρχεία, ένα από τα οποία ονομάζεται ProfileImagePath. Η αντίστοιχη πληροφορία κάτω από τη στήλη Data θα είναι το path για το profile του συγκεκριμένου χρήστη. το όνομα του τελευταίου φακέλου στο path θα είναι το username που σχετίζεται με το συγκεκριμένο SID. [5]

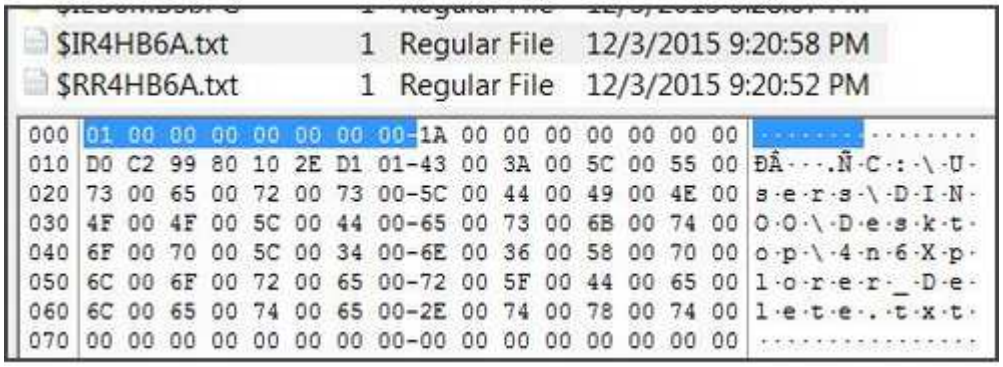
2.2 $I File

Τα $I αρχεία περιέχουν πολύτιμη πληροφορία ακόμα κι αν τα συσχετιζόμενα $R αρχεία αντικατασταθούν. Τα $I αρχεία φυλάσσουν τα παρακάτω metadata από το αρχικό αρχείο:

- ✓ Όνομα αρχείου που διαγράφηκε
- ✓ Μέγεθος αρχείου που διαγράφηκε
- ✓ Πλήρής διαδρομή φακέλου
- ✓ Δεδομένα που στάλθηκαν στον κάδο ανακύκλωσης

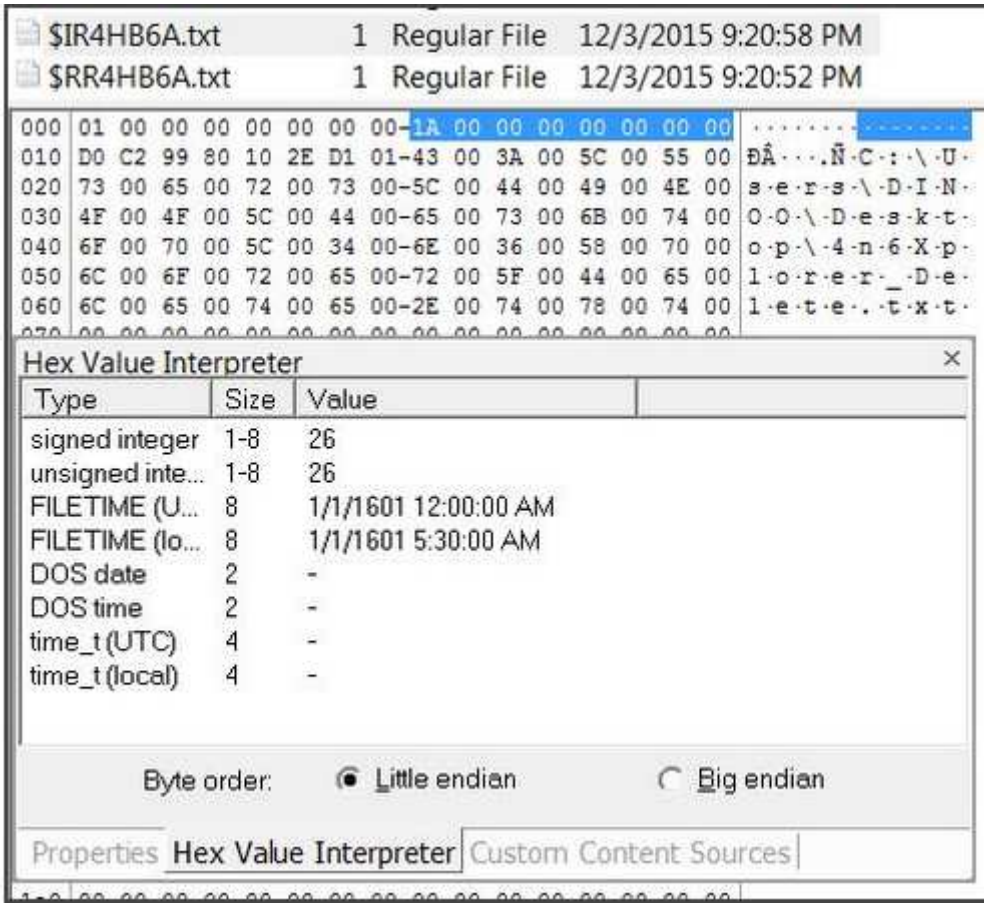
Έστω ότι διαγράφουμε ένα αρχείο το οποίο δημιουργεί τα $IR4HB6A.txt και αντίστοιχα $RR4HB6A.txt. Το $I αρχείο κρατά την παρακάτω πληροφορία:

α) Τα πρώτα 8 bytes (0-7bytes) είναι το $I header, το οποίο είναι πάντα 01 ακολουθούμενο από επτά ζεύγη 00.



Σχήμα 2. Αποκωδικοποίηση του $I αρχείου

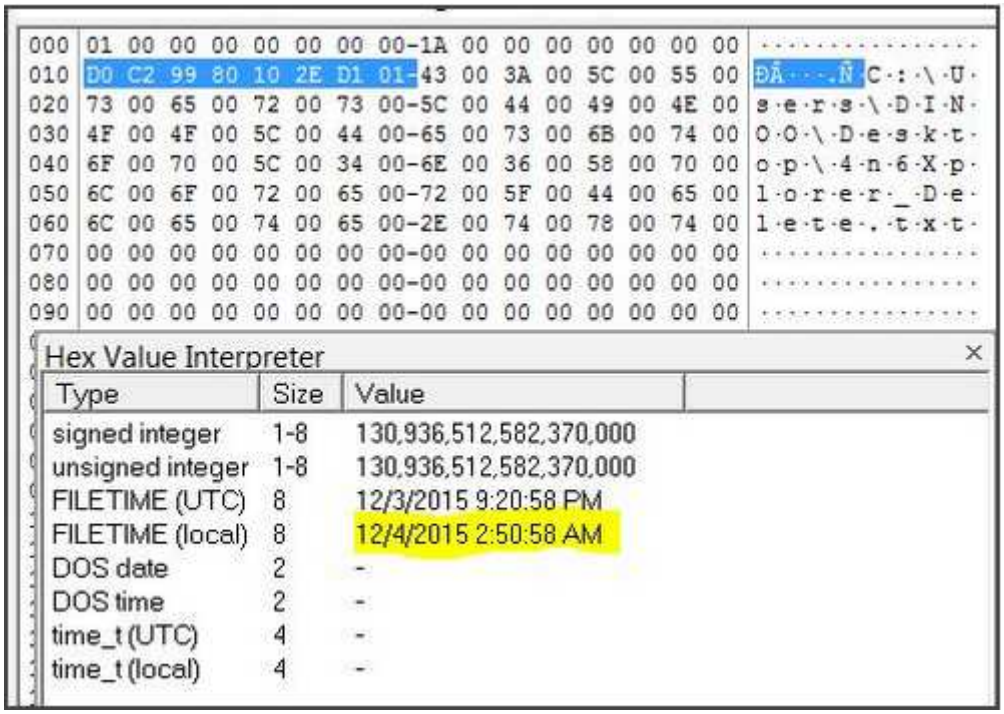
β) τα επόμενα 8 bytes (8-15bytes) είναι το μέγεθος του αρχείου στο δεκαεξαδικό σύστημα. Λόγω Little/Big Endian θα πρέπει να αντιστραφεί η τιμή σε 00 00 00 00 00 00 00 1A. Κατά τη μετατροπή στο δεκαδικό σύστημα η παραπάνω τιμή είναι 26bytes.



Σχήμα 3.

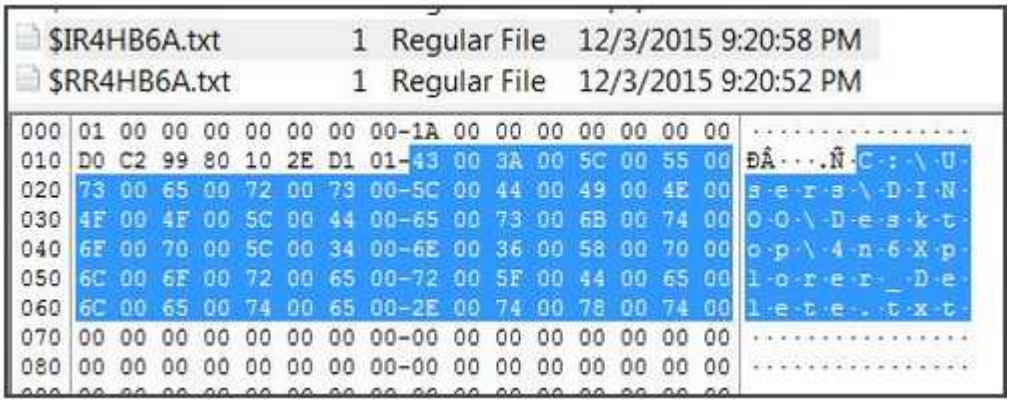
γ) τα επόμενα 8 bytes (16-23bytes) φυλάσσουν την ημερομηνία και ώρα που έγινε η διαγραφή του αρχείου σε μορφή Windows date/time και παρουσιάζονται ως η μετατόπιση (offset) από τα μεσάνυχτα 1η Ιανουαρίου 1601 εκφραζόμενη σε 100 nanoseconds. Επειδή η μορφή είναι Little Endian θα πρέπει να μετατραπεί σε Big Endian

οπότε στο σχήμα 4, στο Hex Value Interpreter του FTK παίρνουμε την ημερομηνία και ώρα.



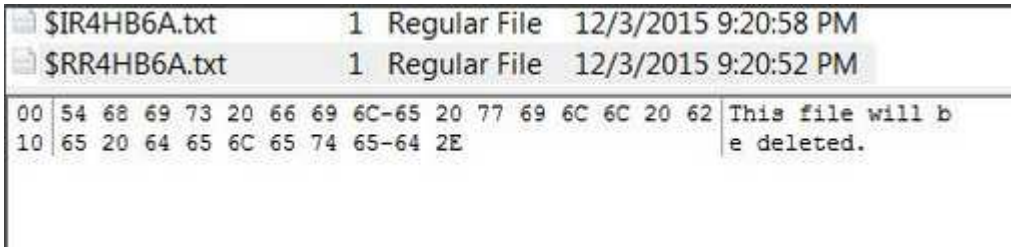
Σχήμα4. Ημερομηνία και ώρα του αρχείου που διαγράφηκε.

δ) Μετά το header, το μέγεθος του αρχείου και την ημερομηνία/ώρα είναι η διαδρομή φακέλων και το όνομα του αρχείου που ξεκινά από το 24ο byte, όπως φαίνεται στο σχήμα 5.



Σχήμα 5. Διαδρομή φακέλου και όνομα αρχείου.

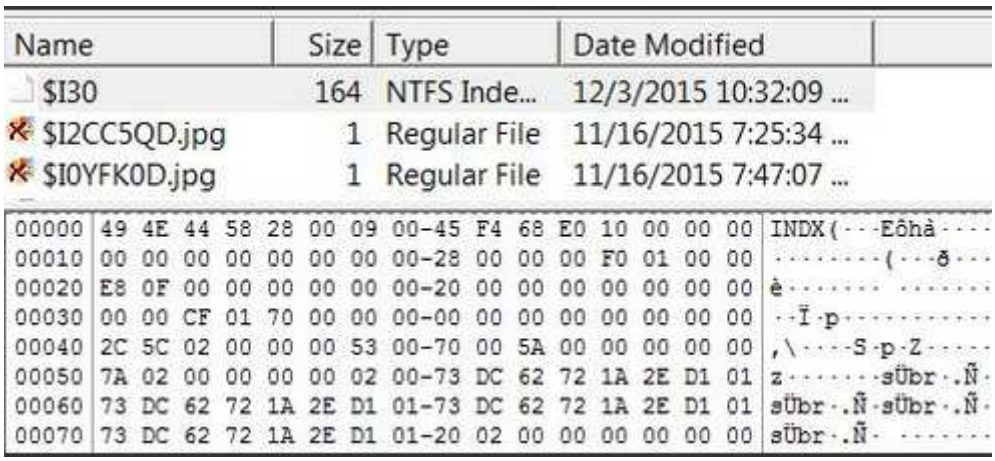
2.3 $R File



Σχήμα 6. Τα δεδομένα του διεγραμμένου αρχείου.

2.4 $I30 File

Το αρχείο $I30 είναι ένα NTFS Index Attribute, το οποίο μπορεί να βοηθήσει να προσδιοριστούν αρχεία που έχουν διαγραφεί ή αντικατασταθεί. Ακόμα κι αν το αρχικό αρχείο δεν υπάρχει πια μπορούμε να ανακτήσουμε το όνομά του, το μέγεθός του, τη διαδρομή στο δίσκο, τη χρονική στιγμή που δημιουργήθηκε, τη χρονική στιγμή που τροποποιήθηκε και προσπελάστηκε, κλπ. χρησιμοποιώντας ειδικό λογισμικό π.χ. EnCase EnPack.



Σχήμα 7. $I30 File Parsing

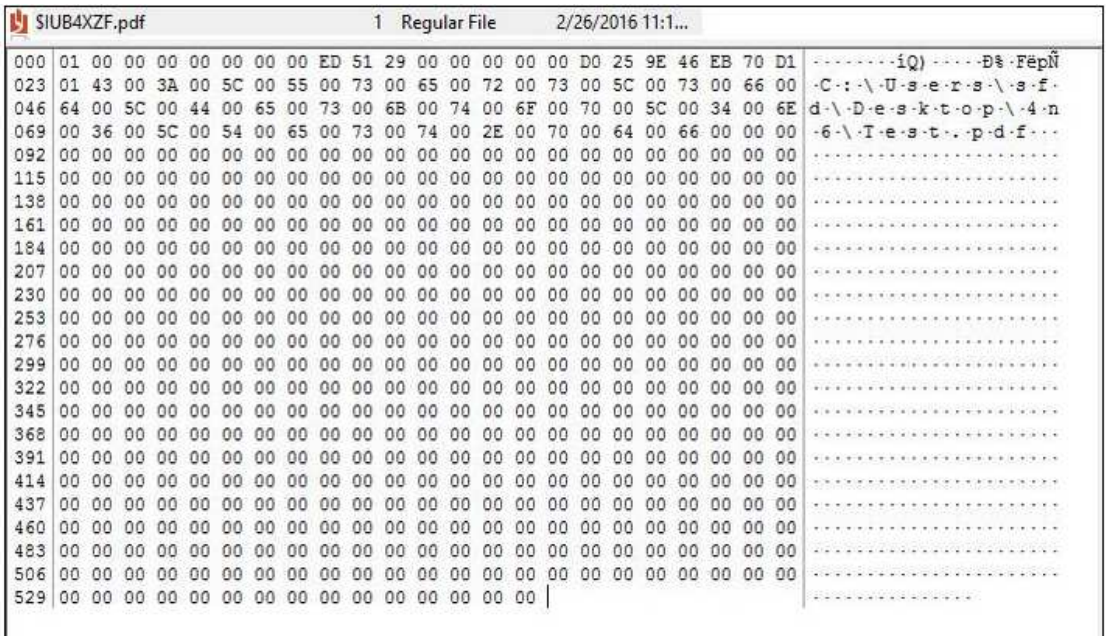
2.5 Ανάκτηση Αρχείων

Όταν ένας φάκελος ο οποίος περιέχει ένα αρχείο μετακινείται στον κάδο ανακύκλωσης τόσο ο φάκελος όσο και το αρχείο αποτελούν ξεχωριστές διαγραφές. Αντίστοιχα αρχεία $I και $R δημιουργούνται για καθένα από αυτά. Όταν ένα αρχείο ανακτάται τα αντίστοιχα αρχεία $I και $R διαγράφονται από τον κάδο και χρησιμοποιούνται για την ανακατασκευή αντιγράφου του αρχικού αρχείου πριν από τη διαγραφή του. Αν αυτό το αρχείο διαγραφεί εκ νέου τότε θα δημιουργηθούν καινούρια διαφορετικά αρχεία $I, $R.

Αν ο φάκελος που υπήρχε πριν από τη διαγραφή πλέον δεν υπάρχει κατά την επαναφορά, αυτός θα δημιουργηθεί ξανά, αλλά θα παραμείνει και στον κάδο ανακύκλωσης. Η ημερομηνία διαγραφής και δημιουργίας φυλάσσονται στο αρχείο $I κατά τη διαγραφή ενός αρχείου. Όταν όμως αυτό ανακτάται χάνεται η πληροφορία διαγραφής και προστίθεται η πληροφορία τροποποίησης και προσπέλασης. Για διαγραμμένο φάκελο όμως δεν ισχύει το ίδιο κατά την επαναφορά. Κατά την επαναφορά διατηρεί την ημερομηνία δημιουργίας αλλά δεν κρατά την πληροφορία τροποποίησης και προσπέλασης. Τέλος υπάρχει το αρχείο desktop.ini στον κάδο ανακύκλωσης, το οποίο παρέχει πληροφορία στο windows explorer για το πως θα απεικονίσει τα περιεχόμενα του φακέλου.

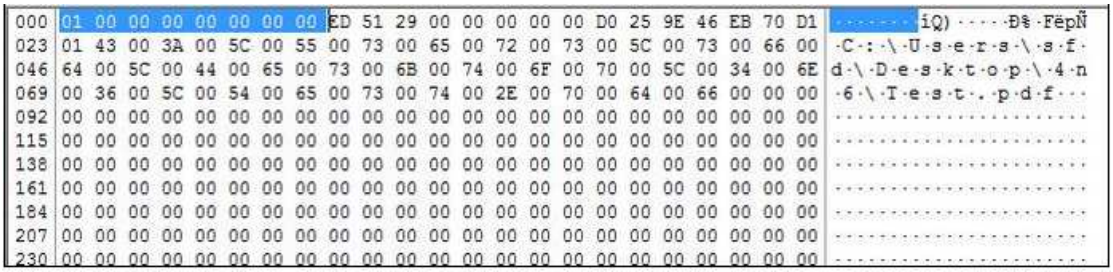
2.6 Windows 8.1 Recycle Bin

Από την ανάλυση στα Windows 8.1 δεν φαίνεται κάποια σημαντική διαφορά αναφορικά με τα Windows 7. Ο κάδος ανακύκλωσης είναι στη διαδρομή DRIVE:\$Recycle.Bin\SID, παρόλα αυτά υπάρχει μια διαφορά στην τελευταία οκτάδα bytes η οποία δείχνει τη διαδρομή στο δίσκο κι επίσης υπάρχουν κάποιες αλλαγές στη μορφή για τα Windows 10. [6]. Δημιουργούνται δύο αρχεία, τα Test.pdf μεγέθους 2645KB και Analysis.pptx μεγέθους 1227KB μέσα σε ένα φάκελο 4n6 στην επιφάνεια εργασίας, τα οποία διαγράφονται.



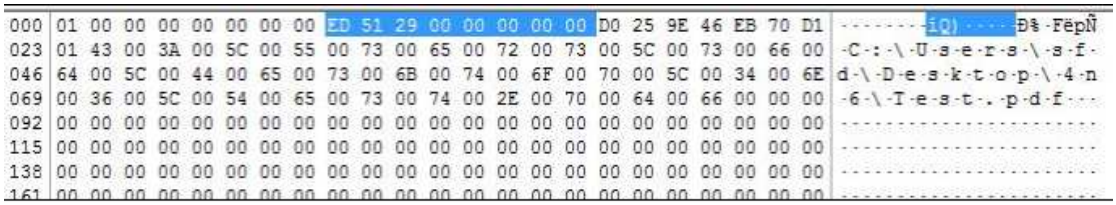
Σχήμα 8. FTK Imager screenshot του $I file για το αρχείο Test.pdf

α) Τα πρώτα 8 bytes (0-7bytes) είναι το header του $I file.



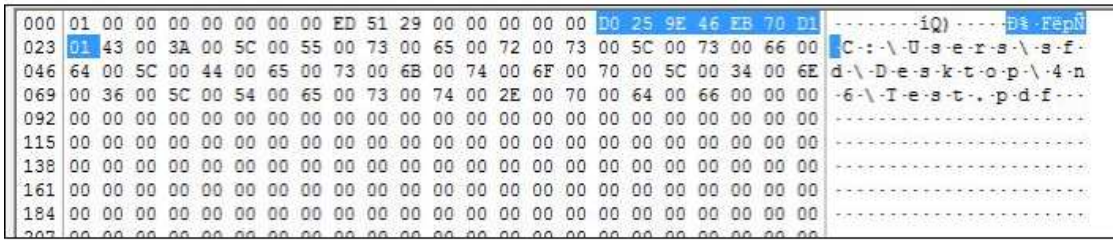
Σχήμα 9. $I file header

β) Τα επόμενα 8bytes (8-15bytes) είναι το μέγεθος του αρχείου



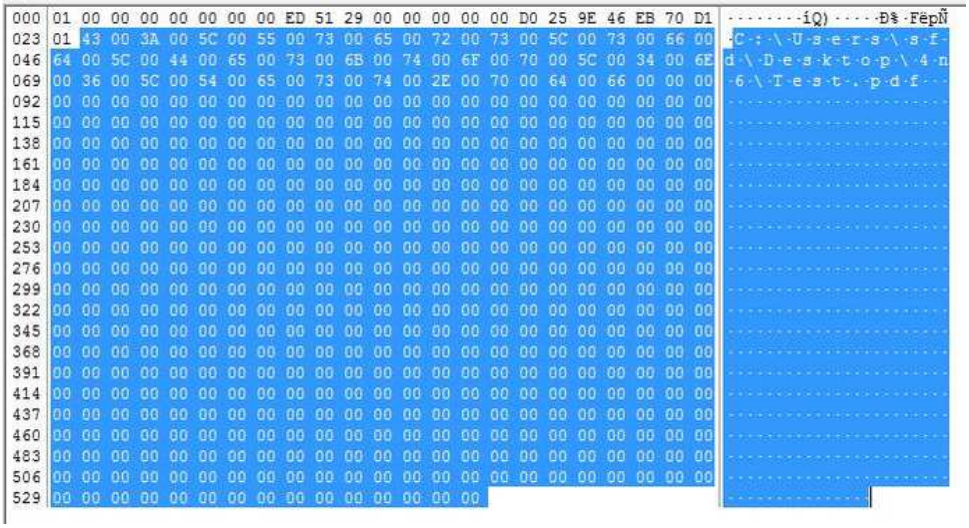
Σχήμα 10. Μέγεθος αρχείου σε bytes.

γ) Τα επόμενα 8bytes (16-23bytes) αποτυπώνουν τη χρονική στιγμή διαγραφής σε δεκαεξαδική μορφή.



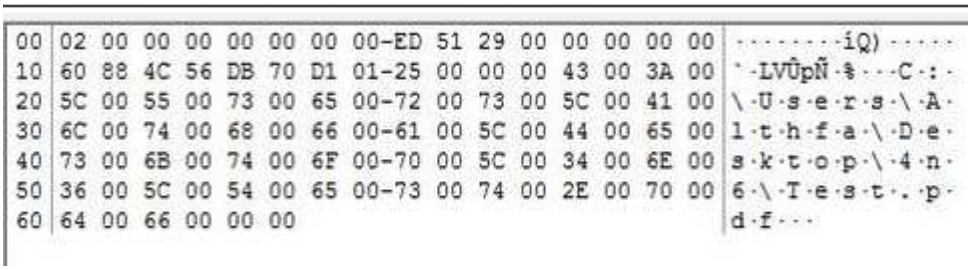
Σχήμα 11. Χρονική στιγμή διαγραφής αρχείου.

δ) τα υπόλοιπα bytes (24-543bytes) δεν εξαρτώνται πλέον από την αρχική διαδρομή στο δίσκο και το όνομα (Windows 7, 10). Είναι πάντα 520bytes ξεκινώντας από offset 24.



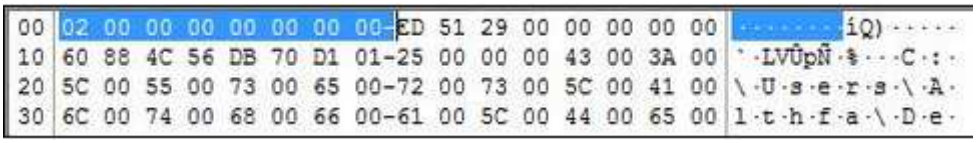
Σχήμα 12. Path, Filename

2.7 Windows 10 Recycle Bin



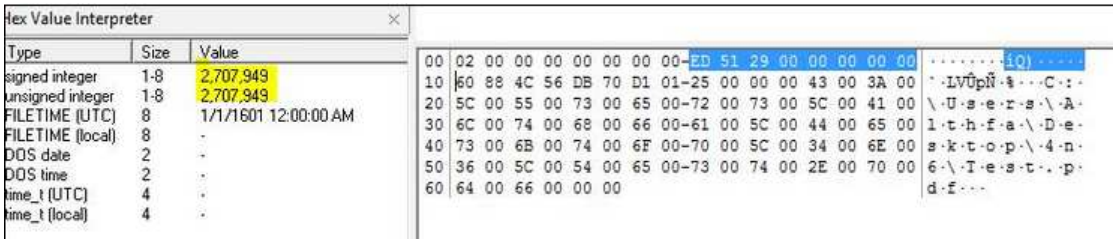
Σχήμα 13. FTK imager screenshot του $I file για το αρχείο Test.pdf (Windows 10).

α) τα πρώτα 8 bytes (0-7bytes) είναι το $I header και ξεκινά με την τιμή 02 (προς διαφοροποίηση με τις προηγούμενες εκδόσεις)



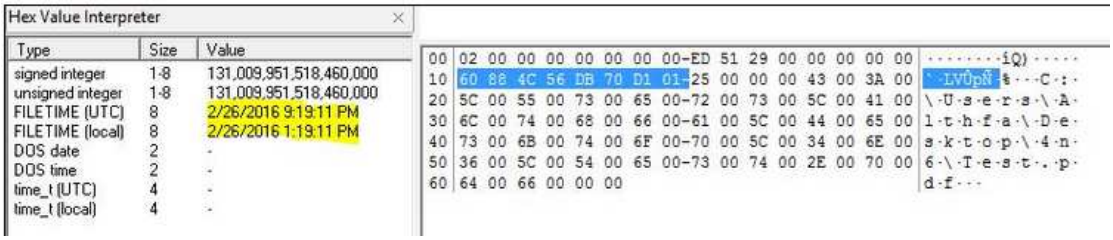
Σχήμα 14. $I file header.

β) τα επόμενα 8bytes (8-15bytes) υποδεικνύουν το μέγεθος του αρχείου σε bytes)



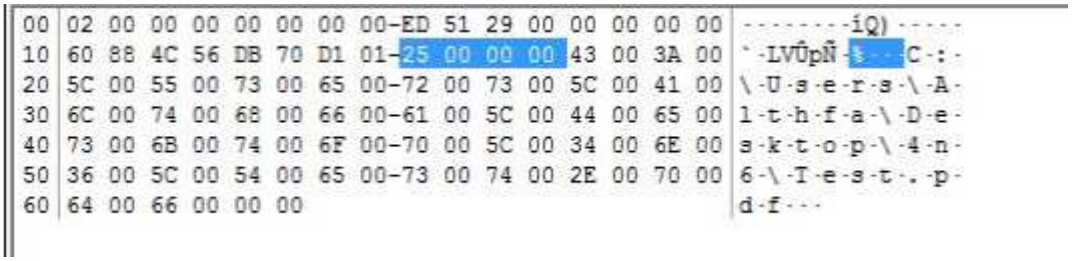
Σχήμα 15. Μέγεθος του αρχείου σε bytes.

γ) τα επόμενα 8bytes (16-23bytes) υποδεικνύουν τη χρονική στιγμή διαγραφής του αρχείου.



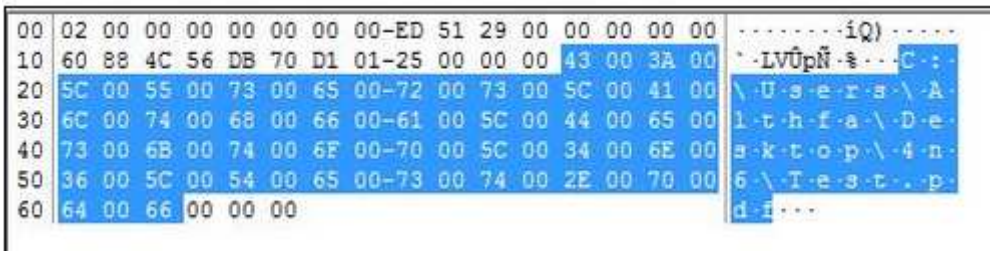
Σχήμα 16. Χρονική στιγμή διαγραφής αρχείου.

δ) τα επόμενα 4bytes (24-27bytes) είναι άγνωστο τι αντιπροσωπεύουν. Δεν αναφέρονται ούτε σε κάποιο metadata του αρχείου ούτε αντιστοιχίζονται σε κάποιο χαρακτηριστικό σχετικά με τη διαγραφή του.



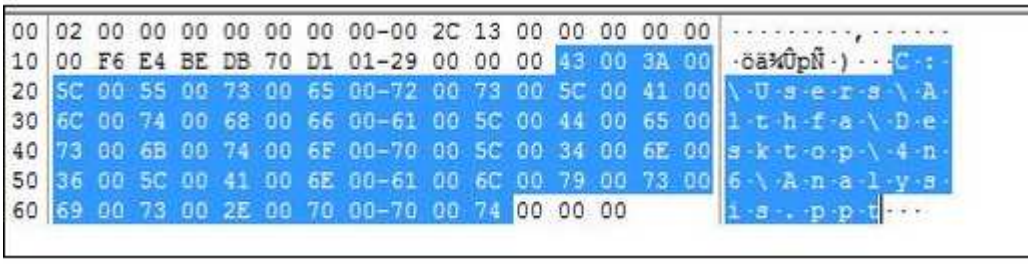
Σχήμα 17. 4bytes άγνωστης πληροφοριακής σημασίας.

ε) Τα υπόλοιπα bytes (28-μεταβλητό μήκος) αντιπροσωπεύουν την πλήρη διαδρομή στο δίσκο και το όνομα του αρχείου. Αυτή η τιμή αντιστοιχεί σε μεταβλητό μήκος αντίθετα με τα Windows 8 όπου είναι σταθερό πάντα στα 520bytes.



Σχήμα 18. Path, Filename.

στ) Το τέλος του αρχείου μαρκάρεται με 3bytes από συνεχόμενα 00.



Σχήμα 19. 3bytes 00 στο τέλος του αρχείου.

3. Volume Shadow Copies

3.1 Εισαγωγή

Το Shadow Copy ή αλλιώς το Volume Snapshot Service ή το Volume Shadow Copy Service (VSS) είναι μια τεχνολογία η οποία περιλαμβάνεται στα Windows και η οποία επιτρέπει τη λήψη χειροκίνητων ή αυτόματων αντιγράφων ασφαλείας ή snapshots από αρχεία ή ολόκληρους δίσκους ακόμα κι όταν αυτά χρησιμοποιούνται. Υλοποιείται μέσω ενός service των Windows, του Volume Shadow Copy Service. [7]

Τα Volume Shadow Copies (VSC) είναι πολύτιμα για έναν αναλυτή ψηφιακών πειστηρίων. Στα Windows 7 το Volume Shadow Copy είναι ένα service το Volume Shadow Service (VSS) το οποίο δημιουργεί snapshots από όλα τα αρχεία συμπεριλαμβανομένων και των αρχείων του χρήστη. Στα Windows 8 η Microsoft συμπεριέλαβε ένα καινούριο χαρακτηριστικό το File History, το οποίο κρατά αντίγραφα βιβλιοθηκών, επιφάνειας εργασίας, Αγαπημένων και επαφών μαζί με κάποια βασικά αρχεία συστήματος. Στα Windows 10 η λειτουργία Shadow Copy επανέρχεται. Η διαφορά μεταξύ του System Restore και του File History είναι ότι το System Restore μας επιτρέπει να μεταβούμε σε μια προηγούμενη κατάσταση, ενώ το File History επαναφέρει αρχεία και δεδομένα από κάποια στιγμή στο παρελθόν. Έτσι είναι δυνατή η διάσωση αρχείων όταν αυτά χαθούν ή καταστραφούν ή ακόμα και ολόκληρου του συστήματος όταν αυτό αποτύχει ή καταστραφεί. Δεν αποτελούν απλώς ένα ακόμα σύστημα λήψης αντιγράφων ασφαλείας γιατί τα VSCs λειτουργούν σε επίπεδο block μέσα στο σύστημα αρχείων. Δεν παρέχουν μόνο ιστορικά δεδομένα, αλλά επιπλέον ανάλυση βοηθά στη σύγκριση των αρχείων κατά τη διάρκεια μιας χρονικής περιόδου.

Για να λειτουργήσει το VSS θα πρέπει να σύστημα αρχείων να είναι NTFS. Το VSS επιβλέπει όλες τις αλλαγές που γίνονται σε ένα δίσκο για τον οποίο έχει ενεργοποιηθεί, σε blocks των 16KB. Όταν παρατηρηθεί αλλαγή σε οποιαδήποτε δεδομένα μέσα σε ένα block των 16KB, ολόκληρο το block αντιγράφεται στο Volume Shadow Copy File. Αυτά τα blocks χωρίζονται σε Index Blocks ή Data Blocks. Τα Index Blocks είναι δείκτες στα 16KB blocks τα οποία έχουν αντιγραφεί από το VSS. Τέλος τα αντίγραφα αυτά δύνανται να δημιουργηθούν τόσο σε εσωτερικούς όσο και σε εξωτερικούς ή δικτυακούς δίσκους. Υπάρχουν δύο τρόποι δημιουργίας αντιγράφων: αντιγραφή όλου του δίσκου ή αντιγραφή μόνο των αλλαγών στο συγκεκριμένο δίσκο. Για οποιαδήποτε μέθοδο από τις παραπάνω δύο δημιουργούνται δύο εικόνες: ο αρχικός δίσκος και ο Shadow Copy δίσκος.

Η διαφορά μεταξύ των δύο είναι ότι ο αρχικός δίσκος έχει attributes εγγραφής/ανάγνωσης ενώ ο Shadow Copy είναι μόνο για ανάγνωση. Η διαδικασία λήψης αντιγράφου ονομάζεται Volume Snapshot και το αντίγραφο Shadow Volume. Όλα τα αρχεία αποθηκεύονται στο φάκελο System Volume Information στο root volume. Τρεις διεργασίες εκτελούνται από το VSS για τη δημιουργία ενός snapshot [8]:

- Freeze: ο σκληρός δίσκος μαρκάρεται ως μόνο-για-ανάγνωση ώστε να μη μπορεί να γραφεί τίποτα σε αυτόν
- Snap: λαμβάνεται το αντίγραφο του δίσκου με κατάλληλες παραμέτρους ώστε να ανακτηθεί σε μεταγενέστερο χρόνο
- Unfreeze: ο σκληρός δίσκος μαρκάρεται ως read/write για να μπορούν να γραφούν δεδομένα

3.2 Σημασία

Τα Volume Shadow Copies περιέχουν πολύτιμα δεδομένα τα οποία μπορεί να έχουν διαγραφεί. Είναι μόνο-για-ανάγνωση οπότε δεν είναι εφικτή η διαγραφή αρχείων από αυτά. Για παράδειγμα, ας υποθέσουμε ένα αρχείο το οποίο βρίσκεται σε ένα δίσκο χωρίς ενεργοποιημένο το VSS. Τα μόνα δεδομένα τα οποία μπορούμε να δούμε είναι τι υπάρχει αυτή τη στιγμή μέσα στο αρχείο. Η λειτουργία των προσωρινών αρχείων μπορεί να μας δείξει κάποια δεδομένα που μπορεί να προϋπήρχαν σε αυτό αλλά δε μπορεί να μας δείξει ακριβώς πως αυτό το αρχείο έχει αλλάξει στην πάροδο του χρόνου. Το ίδιο αρχείο σε δίσκο με ενεργοποιημένο το VSS μπορεί να ανακτηθεί από κάθε Shadow Copy οπότε ο αναλυτής μπορεί να δει ακριβώς τις αλλαγές σε αυτό με όλες τις λεπτομέρειες. Επιπρόσθετα μπορεί να δει και όλες τις αλλαγές που αφορούν το σύστημα στο οποίο βρίσκεται ή βρισκόταν το αρχείο.

Κατά τη διαγραφή ενός αρχείου από το δίσκο, το σύστημα αρχείων απλώς διαγράφει την αντίστοιχη εγγραφή στο MFT, παρόλα αυτά τα δεδομένα του αρχείου παραμένουν στο δίσκο ανέπαφα. Επειδή το VSC λειτουργεί σε επίπεδο block, μπορεί να διατηρεί αυτά τα δεδομένα στα ίδια blocks έως ότου αντικατασταθούν από άλλο καινούριο αρχείο. Κατά την αντικατάσταση ενός αρχείου το VSC λειτουργεί ως εξής:

Έστω ότι έχουμε ένα αρχείο 5MB. Το VSC δεν αντιγράφει τα 5MB δεδομένων, αλλά μόνο τα blocks τα οποία υπάρχουν στο MFT για αυτό. Όταν ένα καινούριο αρχείο αντικαθιστά τα blocks που χρησιμοποιούσε το αρχείο των 5MB, το VSC θα αντιγράψει αυτά τα blocks ενώσω αντικαθίστανται.

3.3 Λειτουργία

Το Volume Shadow Copy Service χρησιμοποιεί μηχανισμό copy-on-write για τη λειτουργία του Previous Versions στα λειτουργικά συστήματα Windows, ως προκαθορισμένο μηχανισμό:

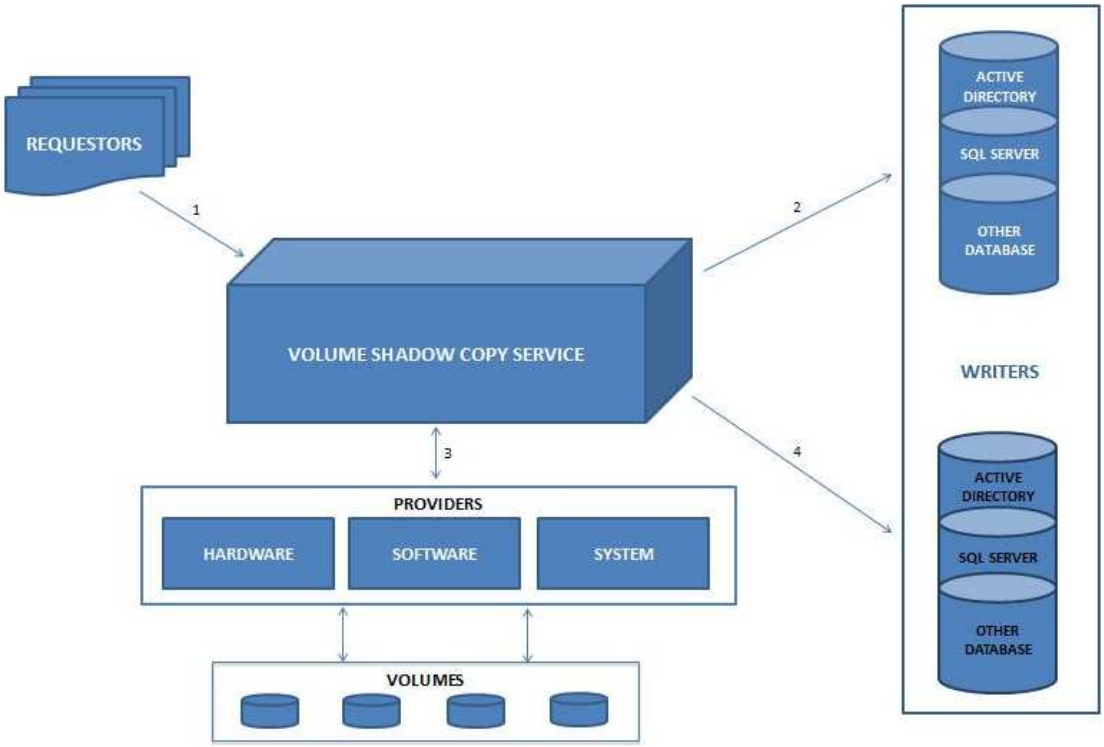
- Clone Shadow Copy: είναι ένα ακριβές αντίγραφο των αρχικών δεδομένων στο δίσκο, το οποίο δημιουργείται είτε μέσω λογισμικού είτε μέσω συσκευής (hardware)
- Copy-on-write: το ενσωματωμένο service στα Windows υποστηρίζει μόνο τέτοια αντίγραφα. Είναι γνωστό και ως Differential Copy αφού δημιουργεί αντίγραφο μόνο των αλλαγών κι όχι πλήρη εικόνα των δεδομένων. Όταν παρατηρείται μια αλλαγή, το block των δεδομένων που άλλαξαν γράφεται σε μια «περιοχή αλλαγών» η οποία σχετίζεται με το shadow copy πριν γίνει αυτή η αλλαγή. Κατά τη διαγραφή ενός αρχείου το μόνο το οποίο αντιγράφεται από το VSS είναι η εγγραφή στο MFT. Τα δεδομένα δεν θα αντιγραφούν έως ότου τα blocks αντικατασταθούν από άλλα δεδομένα.
- Redirect-on-copy: παρόμοια με το copy-on-write με τη διαφορά ότι τα αρχικά δεδομένα αντιγράφονται σε ξεχωριστή τοποθεσία όπως δικτυακός δίσκος ή εξωτερική μονάδα αποθήκευσης [9].

3.4 Αρχιτεκτονική VSS

Το Volume Shadow Copy Service βρίσκεται στο %SystemRoot%\System32\Vssvc.exe και χρησιμοποιεί τα VSS writers, VSS providers και VSS requestors για να δημιουργήσει ένα αντίγραφο [10].

- VSS requestor: είναι η εφαρμογή η οποία στέλνει αιτήματα για δημιουργία αντιγράφων ασφαλείας στο VSS. Παραδείγματα: System Restore, Windows backup
- VSS writer: είναι το λογισμικό το οποίο επιτρέπει σε εφαρμογές όπως το Active Directory, Exchange Server, SQL Server να λάβουν μηνύματα freeze, ώστε τα δεδομένα που αντιγράφονται από αυτές να είναι πλήρη. Γενικά συντονίζει λειτουργίες I/O με τα αντίγραφα ή την επαναφορά αυτών.

- VSS provider: επιτρέπει σε κάποιο κατασκευαστή υλικού ή λογισμικού συμβατότητα με το shadow copy service. Επίσης περιλαμβάνει το μηχανισμό με τον οποίο λαμβάνονται τα αντίγραφα ασφαλείας.



Σχήμα 20. Αρχιτεκτονική του VSS.

Η διαδικασία λήψης αντιγράφων είναι η εξής:

- ✓ Ο requestor επικοινωνεί με το Volume Shadow Copy Service και ζητά τη δημιουργία ενός αντιγράφου. Το VSS επικυρώνει το αίτημα.
- ✓ Το Volume Shadow Copy Service φτιάχνει μια λίστα από writers, οι οποίοι παρέχουν μια λίστα από τα δεδομένα τα οποία πρέπει να αντιγραφούν. Αυτή η λίστα παραδίδεται στον requestor ο οποίος επιλέγει, μέσα από αυτή, τα δεδομένα που θα αντιγραφούν. Ακολούθως ο writer επικοινωνεί με το VSS. Το VSS κατευθύνει τον writer για να κάνει freeze τις λειτουργίες I/O.
- ✓ Το VSS αδειάζει όλους του buffers του συστήματος αρχείων και δίνει εντολή στον provider να δημιουργήσει το αντίγραφο. Ο provider παίρνει τα επιλεγμένα δεδομένα και δημιουργεί snapshot σε συμφωνία με τα αρχικά δεδομένα.
- ✓ Αφού δημιουργηθεί το αντίγραφο, ο provider επικοινωνεί με το Volume Shadow Copy Service για να ενημερώσει τους writers ότι το αντίγραφο δημιουργήθηκε, άρα με τη σειρά τους θα πρέπει να απελευθερώσουν τις I/O λειτουργίες. Το freeze δεν επιτρέπεται να διαρκέσει περισσότερα από 60 δευτερόλεπτα.

3.5 VSS Registry Keys

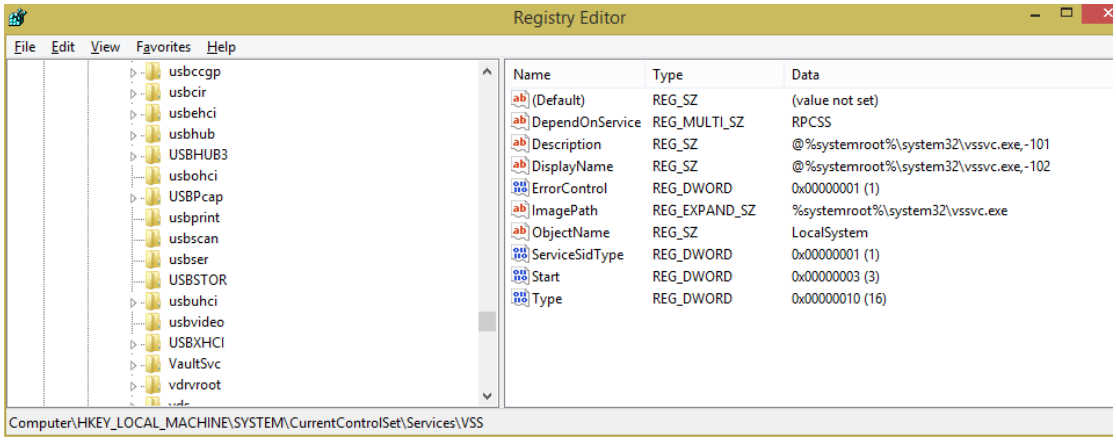
Τα πιο σημαντικά κλειδιά στη registry είναι τα εξής:

α) HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VSS

β) HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\BackupRestore

γ) HKEY_LOCAL_MACHINE \Software\Microsoft\Windows NT\CurrentVersion\SPP\Clients

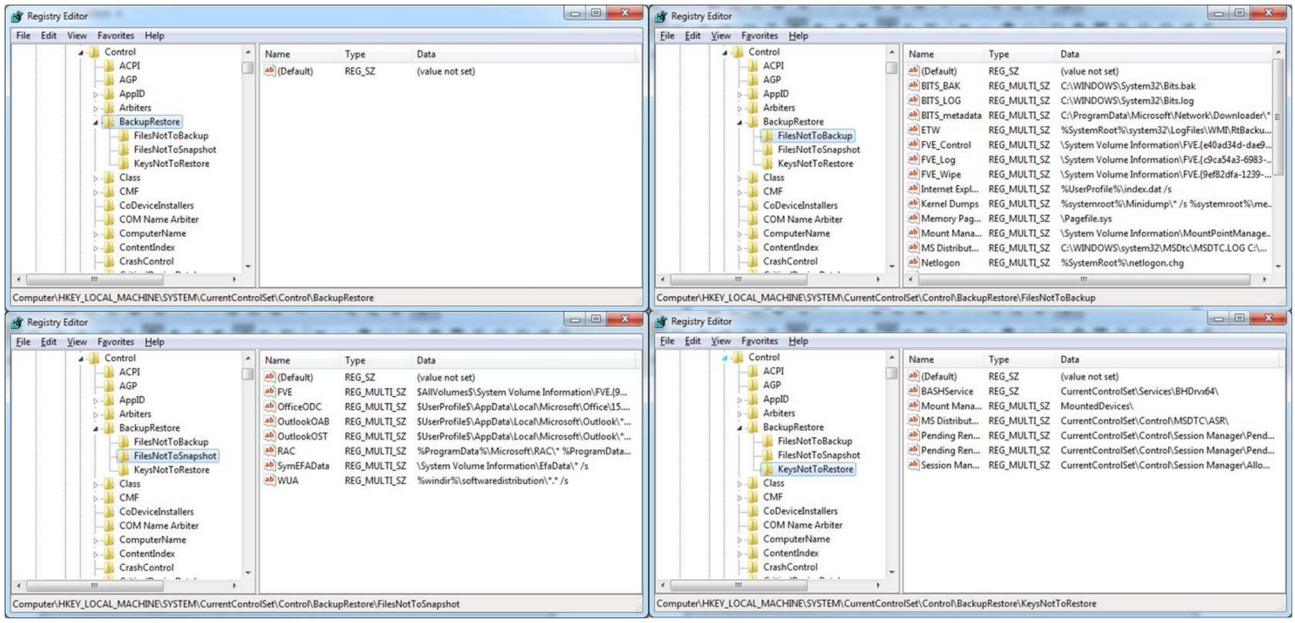
α) το HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VSS είναι το πιο σημαντικό κλειδί στη registry το οποίο καθορίζει τη λειτουργία του VSS. Μπορούμε να αλλάξουμε την τιμή Start στις τιμές: (2) για αυτόματη εκκίνηση, (3) για χειροκίνητη εκκίνηση και (4) για απενεργοποίηση.



Σχήμα 21. Εικόνα από τον registry editor στα Windows 8.1.

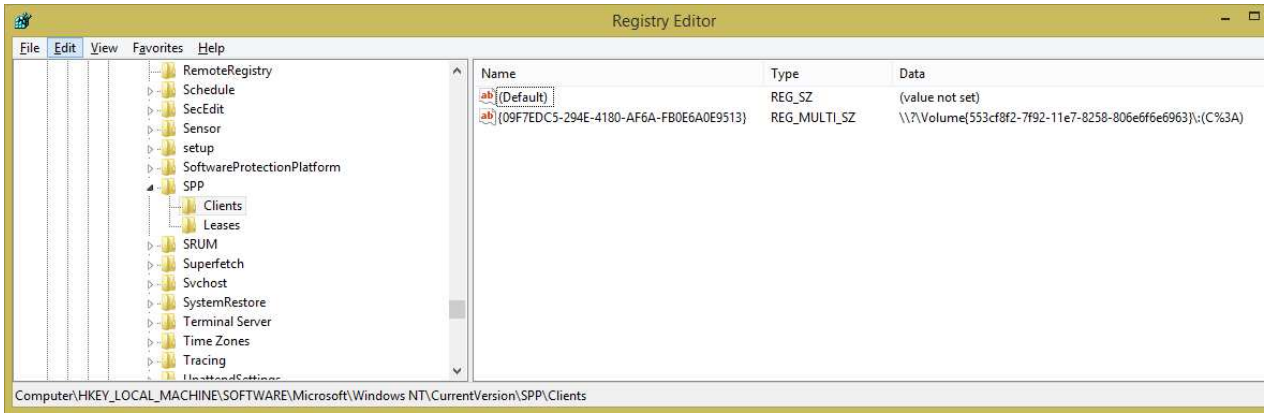
β) το HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\BackupRestore είναι ένα άλλο κλειδί το οποίο επηρεάζει τη λειτουργία του VSC. Κάτω από αυτό υπάρχουν τρία subkeys:

- ✓ FilesNotToBackup: περιέχει μια λίστα από αρχεία και φακέλους που δεν πρέπει να συμπεριληφθούν στο αντίγραφο ασφαλείας. Ανάμεσα σε αυτά περιλαμβάνονται αρχεία του φακέλου Temp, pagefile.sys, hiberfil.sys κ.ά.
- ✓ FilesNotToSnapshot: περιέχει μια λίστα από αρχεία τα οποία θα πρέπει να διαγραφούν από τα καινούρια αντίγραφα ασφαλείας.
- ✓ KeysNotToRestore: περιέχει λίστα από κλειδιά και τιμές που δε θα πρέπει να ανακτηθούν.



Σχήμα 22. Εικόνα από τον registry editor στα Windows 7.

γ) το HKEY_LOCAL_MACHINE \Software\Microsoft\Windows NT\CurrentVersion\SPP\Clients περιέχει μια τιμή, στον υπολογιστή του γράφοντος αυτή είναι {09F7EDC5-294E-4180-AF6A-FB0E6A0E9513} κι αυτή η τιμή είναι η ίδια σε όλους τους δίσκους. Εδώ φαίνονται ποιοι δίσκοι επιβλέπονται από το VSS. Μπορεί να περιέχει διάφορες συμβολοσειρές, κάθεμιά από τις οποίες αναφέρει το GUID και το γράμμα του δίσκου. Αυτή η τιμή θα αντιγράψει οτιδήποτε περιέχεται στην καρτέλα Protection Settings του παραθύρου System Properties.

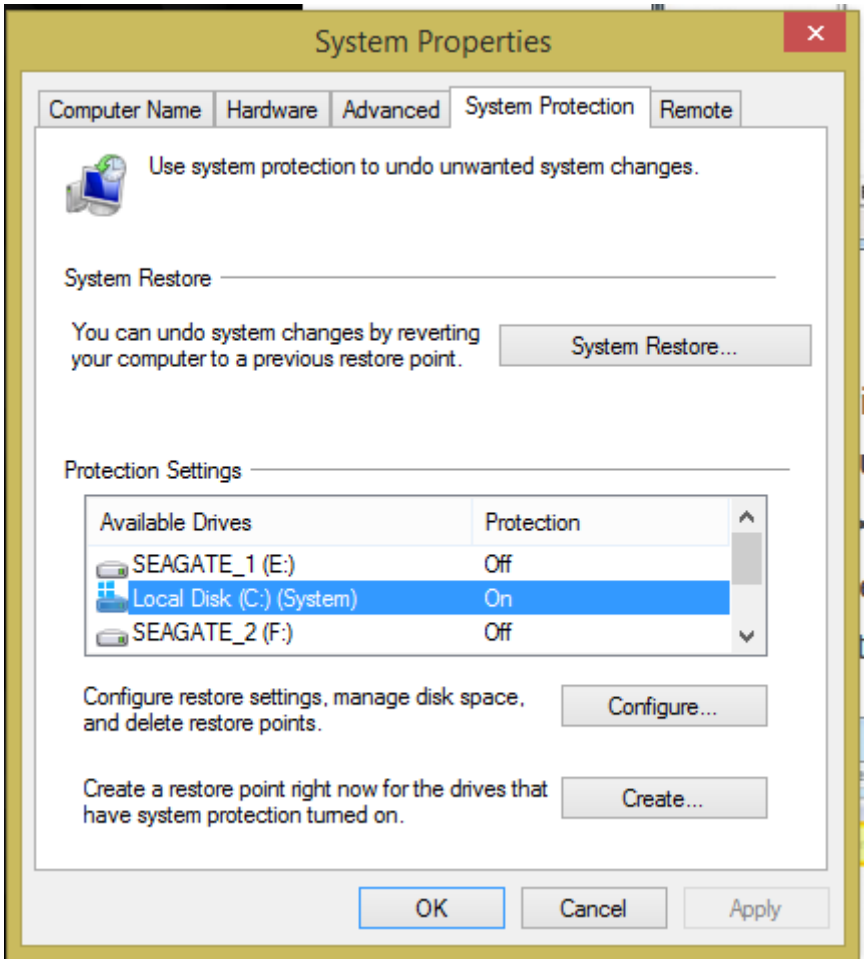


Σχήμα 23. Εικόνα από τον registry editor στα Windows 8.1.

3.6 Παραμετροποίηση VSS, VSC & Previous Versions

Εκτός από την αλλαγή της τιμής του κλειδιού το HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VSS που περιγράφηκε

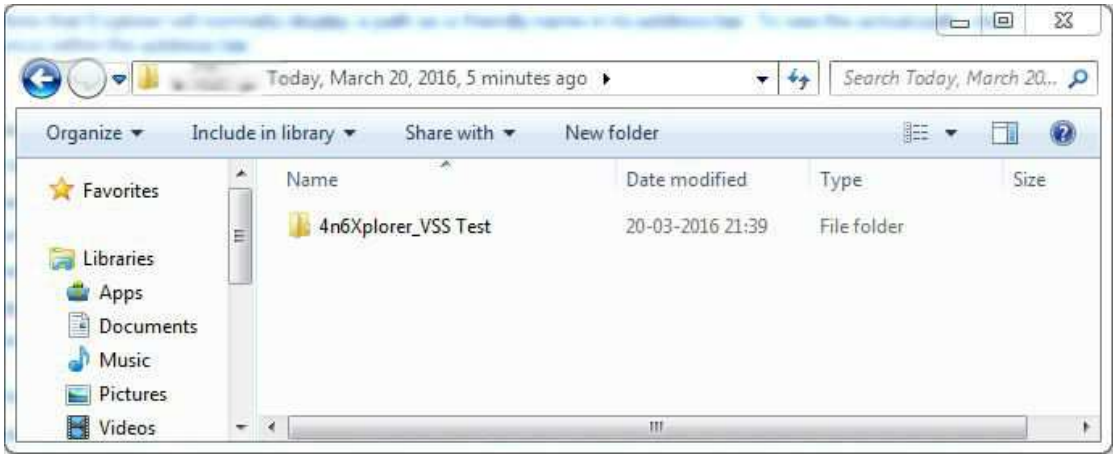
στην ενότητα 3.6, το ίδιο μπορεί να γίνει και από την Windows Services Management Console εκτελώντας την εντολή services.msc. Οι ρυθμίσεις του VSC μπορούν να τροποποιηθούν από τα System Properties, από όπου επιλέγοντας το δίσκο μπορεί να δημιουργηθεί επιτόπου ένα restore point.



Σχήμα 24. Ρυθμίσεις του System Protection στα Windows 8.1.

Μπορεί να χρησιμοποιηθεί η καρτέλα Previous Versions για να φανούν όλες οι εκδόσεις του συγκεκριμένου partition κάθε φορά που λαμβάνεται αντίγραφο ασφαλείας. Αυτό ανοίγει ένα παράθυρο το οποίο εμφανίζει το partition τη στιγμή που λήφθηκε αντίγραφο ασφαλείας. Η διαδρομή δίσκου αποτελείται από localhost\C$\<volume label>(<drive>:)(<date,time>) και είναι ο τρόπος με τον οποίο ο windows explorer οπτικοποιεί το shadow copy (σχήμα 25).

Τα Windows 8 έχουν ένα διαφορετικό μηχανισμό. Το Previous Versions έχει αντικατασταθεί από το File History. Αντίθετα με το Shadow Copy, το οποίο επιτηρεί τα blocks, το File History χρησιμοποιεί το USN Journal για να παρακολουθεί τις αλλαγές και απλά αντιγράφει προηγούμενες εκδόσεις των αρχείων στην backup τοποθεσία που έχει καθοριστεί από το χρήστη. Επιπλέον υπάρχουν δύο επιλογές στο System Protection: ON και OFF. Τέλος τα Windows 7 δε μπορούν να διαβάσουν Shadow Copies από Windows 8.



Σχήμα 25. Previous Versions στα Windows 7.

Στα Windows 8 το Previous Versions έχει αφαιρεθεί σαν δυνατότητα για τοπικούς δίσκους.

3.7 Περιορισμοί σε έρευνες ψηφιακών πειστηρίων

Τα Shadow Copies μπορούν να παρέχουν πληροφορίες για αρχεία τα οποία έχουν διαγραφεί χρονικά ανάμεσα στη λήψη αντιγράφου και στη χρονική στιγμή της έρευνας, αλλά παρέχουν μόνο μια έκδοση των αρχείων. Αν έχουν γίνει αλλαγές πριν δημιουργηθεί το αντίγραφο αυτές θα χαθούν. Επειδή τα Shadow Copies αντιγράφουν σε επίπεδο block κι όχι σε επίπεδο αρχείου, αλλαγές σε μεμονωμένα αρχεία μπορεί να μην είναι αρκετές ώστε να αναγκάσουν το λειτουργικό σύστημα να δημιουργήσει το αντίστοιχο αντίγραφο.

Επιπρόσθετα ο χρήστης μπορεί να απενεργοποιήσει το Shadow Copy service. Άλλο εμπόδιο στις έρευνες μπορεί να είναι το γεγονός ότι οι ρυθμίσεις απαιτούμενου αποθηκευτικού χώρου για τη λήψη αντιγράφου να προσδιορίζουν πιο μικρή χωρητικότητα από αυτή που απαιτείται, επομένως να μην υπάρχουν στο σύστημα πολλές εκδόσεις από αντίγραφα.

4. Windows Shellbags

Τα shellbags είναι ίσως το πιο σημαντικό artifact στη registry από πλευράς άντλησης πληροφοριών. Οι πληροφορίες που μπορούν να αντληθούν από αυτά περιλαμβάνουν: bag number, χρονική στιγμή τελευταίας εγγραφής στη registry, όνομα φακέλου, διαδρομή στο δίσκο, ημερομηνία και ώρα δημιουργίας, τροποποίησης και προσπέλασης [11]. Πιο συγκεκριμένα μας δίνουν πληροφορίες για:

- ✓ Ποια αρχεία προσπελάστηκαν από ένα συγκεκριμένο χρήστη χρησιμοποιώντας τον windows explorer είτε από το τερματικό τοπικά είτε μέσω δικτύου ή ακόμα και από αφαιρούμενο αποθηκευτικό μέσο, π.χ. USB drive
- ✓ Πληθώρα στοιχείων για προϋπάρχοντες φακέλους και αρχεία είτε μετά από αντικατάσταση είτε μετά από διαγραφή
- ✓ Πόσοι και ποιοι διαφορετικοί χρήστες έχουν πρόσβαση για συγκεκριμένους φακέλους μέσα στο σύστημα
- ✓ Τα μέσα με τα οποία προσπελάστηκε ένας συγκεκριμένος φάκελος είτε μέσω συντόμευσης είτε μέσω πλοήγησης στον windows explorer
- ✓ Πότε προσπελάστηκαν συγκεκριμένοι φάκελοι του συστήματος
- ✓ Ποιοι είναι οι χρόνοι MAC (Modified, Accessed, Created) των φακέλων σε αντιδιαστολή με το χρόνο που υποτίθεται ότι υπήρχαν στο σύστημα

4.1 Ιστορική αναδρομή

Από τα Windows XP τα shellbags υπάρχουν στη registry. Υπάρχουν σε όλες τις μετέπειτα εκδόσεις των Windows. Χρησιμοποιούνται για να κρατούν στοιχεία σχετικά με τη δραστηριότητα του χρήστη στο λειτουργικό σύστημα, για να καθορίσουν τις ενέργειες ενός επιτιθέμενου που έχει προσβάλλει ένα υπολογιστικό σύστημα καθώς και τη χρήση (ή μη) αφαιρούμενων μέσων αποθήκευσης [12], [14].

4.2 Δομή των Shellbags

Το σύνολο της πληροφορίας κείται σε δύο στοιχεία, το BagMRU και το Bags. Τα κλειδιά BagMRU αντιπροσωπεύουν το desktop, τα child keys δεν έχουν όμως κάποια αναφορά

σε συγκεκριμένους φακέλους. Φυλάσσουν τα ονόματα και τις τοποθεσίες στο δίσκο, ενώ το Bags φυλάσσουν τις προτιμήσεις θέασης και το μέγεθος του παραθύρου [17]. Τα κλειδιά που βρίσκονται μέσα στο BagMRU έχουν μια τιμή στη registry, την MRUListEx, η οποία είναι binary και κρατούν τη σειρά των πρόσφατων φακέλων/υποφακέλων που προσπελάστηκαν. Αυτά που έχουν την NodeSlot, η οποία είναι DWORD, κρατούν τις επιλογές θέασης για τους διαφορετικούς χρήστες και δείχνουν σε κλειδί στο Bags [14], [17].

Η δομή των shellbags διαφέρει ανάμεσα στις εκδόσεις των Windows, όσον αφορά την τοποθεσία των κλειδιών και τις τιμές αυτών. Για παράδειγμα, στα Windows 7 δεν υπάρχει το κλειδί ShellNoRoam, το οποίο υπήρχε στα Windows XP και η αντίστοιχη πληροφορία φυλάσσεται στα κλειδιά Shell. Τα κλειδιά αυτά βρίσκονται κάτω από το BagMRU με την ίδια σειρά με την όποια έγινε η πρόσβαση/θέαση μέσα από τον windows explorer [14], [17].

4.3 Πληροφορίες για τα Shellbags

Τα shellbags αποθηκεύονται στα κλειδιά BagMRU με τον ίδιο τρόπο και σειρά όπως απεικονίζονται στον windows explorer οι γονικοί φάκελοι και οι φάκελοι παιδιά. Όλοι αυτοί οι φάκελοι περιέχουν τα κλειδιά MRUListEx, NodeSlot, NodeSlots με τιμές :[14], [18].

- ✓ Το MRUListEx έχει τιμή 4 bytes και δείχνει τη σειρά με την οποία προσπελάστηκε κάθε φάκελος παιδί κάτω από τη λίστα BagMRU. Για παράδειγμα, έστω ότι έχουμε έναν φάκελο ο οποίος έχει τρεις υποφακέλους (0,1 και 2) και έστω ότι ο φάκελος 2 προσπελάστηκε τελευταία. Το MRUListEx θα περιέχει από κάτω το φάκελο 2 πρώτο και μετά με τη σειρά προσπέλασης τους φακέλους 0 και 1.
- ✓ Το NodeSlot είναι μια τιμή που αντιστοιχεί στο κλειδί Bags και στη συγκεκριμένη ρύθμιση θέασης για το δεδομένο φάκελο. Συνδυάζοντας τις πληροφορίες των παραπάνω ο αναλυτής μπορεί να συμπεράνει τι είδε ο χρήστης και με ποιο τρόπο.
- ✓ Το NodeSlots βρίσκεται στο BagMRUSubKeys και ενημερώνεται μόνο όταν ένα καινούριο Shellbag δημιουργείται στο σύστημα.

4.4 Ανάλυση των ShellBags

Τα δεδομένα στα ShellBags είναι σε μορφή hex οπότε θα πρέπει να μετασχηματιστούν για να μπορεί η διαδρομή στο δίσκο και οι υπόλοιπες πληροφορίες να είναι αναγνώσιμες από τον αναλυτή. Αυτός απαιτείται να συνδυάσει όλα τα δεδομένα από κάθε στοιχείο στη σωστή σειρά για να μπορεί να καταλήξει σε κάποιο συμπέρασμα. Αυτό στη συνέχεια θα τον οδηγήσει στα κλειδιά Bags ώστε να αντλήσει πληροφορίες για τα εικονίδια, τη θέση, και το χρόνο και ακόμα για το ποιοι φάκελοι έχουν διαγραφεί και αν βρίσκονταν ή όχι σε αφαιρούμενο μέσο αποθήκευσης [12], [18].

Η πληροφορία που αντλεί ο ερευνητής από το ShellBags hive αναλύεται στα εξής [13]:

- ✓ Το Bag number το οποίο προσδιορίζει το Bags SubKeys και περιέχει τις ρυθμίσεις του χρήστη (αναφέρεται και ως NodeSlot)
- ✓ Τη χρονική στιγμή τελευταίας εγγραφής κλειδιού της registry καθορίζει τη χρονική στιγμή πρώτης προσπέλασης ή την τελευταία αλλαγή σε φάκελο
- ✓ Το όνομα του φακέλου όπως αυτό βρίσκεται στο σύστημα
- ✓ Η πλήρης διαδρομή του φακέλου στο δίσκο
- ✓ Η ημερομηνία δημιουργίας και ώρα όπως αυτή είναι αποθηκευμένη τη στιγμή που δημιουργήθηκε το κλειδί BagMRU
- ✓ Η ημερομηνία τροποποίησης και η ώρα όπως αυτή είναι αποθηκευμένη τη στιγμή που δημιουργήθηκε το κλειδί BagMRU
- ✓ Η ημερομηνία προσπέλασης και η ώρα όπως αυτή είναι αποθηκευμένη τη στιγμή που δημιουργήθηκε το κλειδί BagMRU

4.5 Χρήση των ShellBags

Η ανάδραση σε ένα περιστατικό δείχνει πως χρησιμοποιούνται τα ShellBags από έναν ερευνητή. Η πληροφορία για τη δραστηριότητα του χρήστη παραμένει ακόμα κι όταν οι πόροι τους οποίους χρησιμοποίησε δεν υπάρχουν ή δεν είναι πλέον διαθέσιμοι. Τα ShellBags είναι πολύ χρήσιμα σε υποθέσεις εισβολής σε ένα υπολογιστικό σύστημα γιατί μπορούν να απαντήσουν αναφορικά με τη δραστηριότητα του εισβολέα και το τι δεδομένα προσπέλασε. Επίσης μπορούμε να δούμε τα περιεχόμενα ενός USB drive στην περίπτωση που αυτό χρησιμοποιήθηκε κατά την εισβολή, ακόμα κι αν αυτό ήταν κρυπτογραφημένο ή όλα τα περιεχόμενά του διαγράφησαν. Επιπλέον, αποκαλύπτεται ο τρόπος με τον οποίο οι χρήστες χρησιμοποιούν τα αρχεία. Η πληροφορία στο σύνολό της

μπορεί να βοηθήσει συνδυαστικά με την ανάλυση άλλων artifacts όπως οι συντομεύσεις αρχείων (LNK files) [19].

4.6 Τοποθεσία των ShellBags

Τα ShellBags μπορούν να αναλυθούν με χρήση του registry editor που είναι προεγκατεστημένος στο λειτουργικό σύστημα. Θα πρέπει όμως να χρησιμοποιηθεί κι ένας parser έτσι ώστε να οπτικοποιηθούν τα δεδομένα που περιέχουν.

Στα Windows 7, 8, (8.1) και 10 βρίσκονται στο NTUSER.DAT και UsrClass.dat hives:

HKEY_CURRENT_USER\Software\Classes\LocalSettings\Software\Microsoft\Windows \Shell\BagMRU

HKEY_CURRENT_USER\Software\Classes\LocalSettings\Software\Microsoft\Windows \Shell\Bags

4.7 Δημιουργία ShellBags σε τοπικό σύστημα & USB drive

Ας δούμε τώρα πως δημιουργούνται και υπό ποιες συνθήκες τα ShellBags σε ένα σύστημα και όταν σε αυτό συνδεθεί ένα USB drive.

Ενέργεια	**Επίπτωση στα ShellBags**
Δημιουργία φακέλου	Δεν δημιουργούνται εγγραφές
Εξερεύνηση φακέλου	Δημιουργούνται εγγραφές ShellBags
Δημιουργία φακέλων σε υπάρχων φάκελο	Δημιουργούνται εγγραφές ShellBags για κάθε νέο φάκελο
Κλείσιμο παραθύρων	Δεν δημιουργούνται εγγραφές
Εξερεύνηση εκ νέου των καινούριων φακέλων	Δημιουργούνται εγγραφές ShellBags αντίστοιχα για φακέλους που εξερευνόνται για πρώτη φορά/γίνονται αλλαγές για αυτούς που εξερευνώνται ξανά
Διαγραφή φακέλου	Δεν δημιουργούνται εγγραφές
Μετακίνηση φακέλου στην επιφάνεια εργασίας	Ενημέρωση των Bags Subkeys

Ενέργεια	Επίπτωση στα ShellBags
Σύνδεση USB drive	Δεν δημιουργούνται εγγραφές
Αντιγραφή φακέλου από τον υπολογιστή στο USB drive	Ενημερώνεται το MRU time
Εξερεύνηση του φακέλου στο USB drive	Ενημερώνεται το MRU time δεν δημιουργούνται εγγραφές ShellBags
Εξερεύνηση του USB drive	Δημιουργούνται εγγραφές ShellBags
Κλείσιμο του φακέλου και διαγραφή μέσω γραμμής εντολών	Δεν δημιουργούνται εγγραφές
Κλείσιμο παραθύρου και εξαγωγή του USB drive	Δεν δημιουργούνται εγγραφές

Τέλος, αξιοσημείωτο εύρημα είναι το παρακάτω γεγονός. Ως γνωστό, οι εγγραφές shellbags δεν λαμβάνουν χώρα εκτός κι αν γίνει πλοήγηση εντός φακέλου. Στα Windows 7 και 10 παρατηρήθηκε διαφορετική συμπεριφορά. Δημιουργήθηκε ένα directory από τη γραμμή εντολών και δεν δημιουργήθηκαν εγγραφές ShellBags. Με μόνο ένα αριστερό κλικ στον φάκελο, απλώς επιλέγοντάς τον αμέσως δημιούργησε εγγραφές. Επιπρόσθετα, από τη γραμμή εντολών δημιουργήθηκαν νέοι φάκελοι, επιλέχθηκε ένας και με τη βοήθεια του πληκτρολογίου επιλέχθηκαν και οι υπόλοιποι ένας κάθε φορά, χωρίς να πλοηγηθούμε εντός αυτών. Αυτό επίσης δημιούργησε νέες εγγραφές [20].

5. Jump Lists

Οι Jump Lists πρωτοεμφανίστηκαν στα Windows 7 και παρέχουν στο χρήστη ένα γραφικό περιβάλλον, το οποίο σχετίζεται με αρχεία κάθε εγκατεστημένης εφαρμογής τα οποία ο χρήστης έχει προηγουμένως προσπελάσει. Όταν ο χρήστης εκτελεί κάποιες συγκεκριμένες ενέργειες, δημιουργούνται δύο τύποι από Jump Lists, automatic και custom. Οι Jump Lists δημιουργούνται για καθένα χρήστη στις εξής τοποθεσίες:

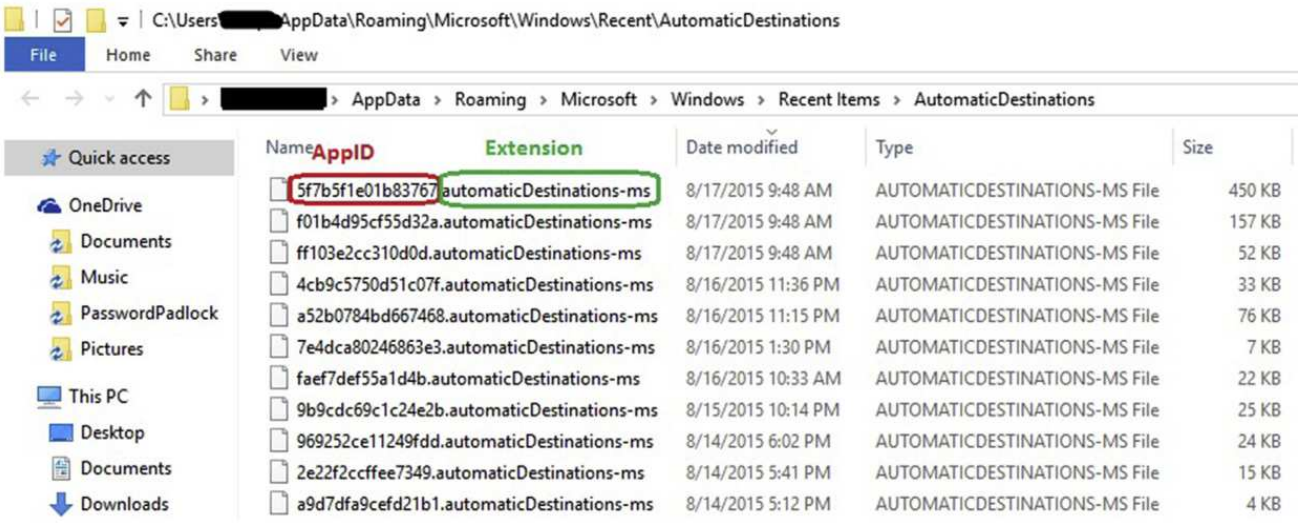
AutoDest:

%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

CustDest:

%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations

Ο πρώτος τύπος σχετίζεται με ενέργειες του χρήστη οι οποίες είναι τυχαίες, π.χ. άνοιγμα αρχείων ή συνδέσεις απομακρυσμένης επιφάνειας εργασίας. Ο δεύτερος τύπος δημιουργείται όταν ο χρήστης, μέσω της taskbar, κάνει pin ένα αρχείο σε μια εφαρμογή [21].



Σχήμα 26. Τα αρχεία Jump Lists στα Windows 10 για τα automaticDestinations.

5.1 Δομή Αρχείων

Η ονομασία του κάθε αρχείου αποτελείται από ένα δεκαεξαδικό αριθμό 16 ψηφίων το οποίο είναι το AppID (Application Identifier) ακολουθούμενο από την επέκταση automaticDestinations-ms ή customDestinations-ms. Τα AppIDs υπολογίζονται από το

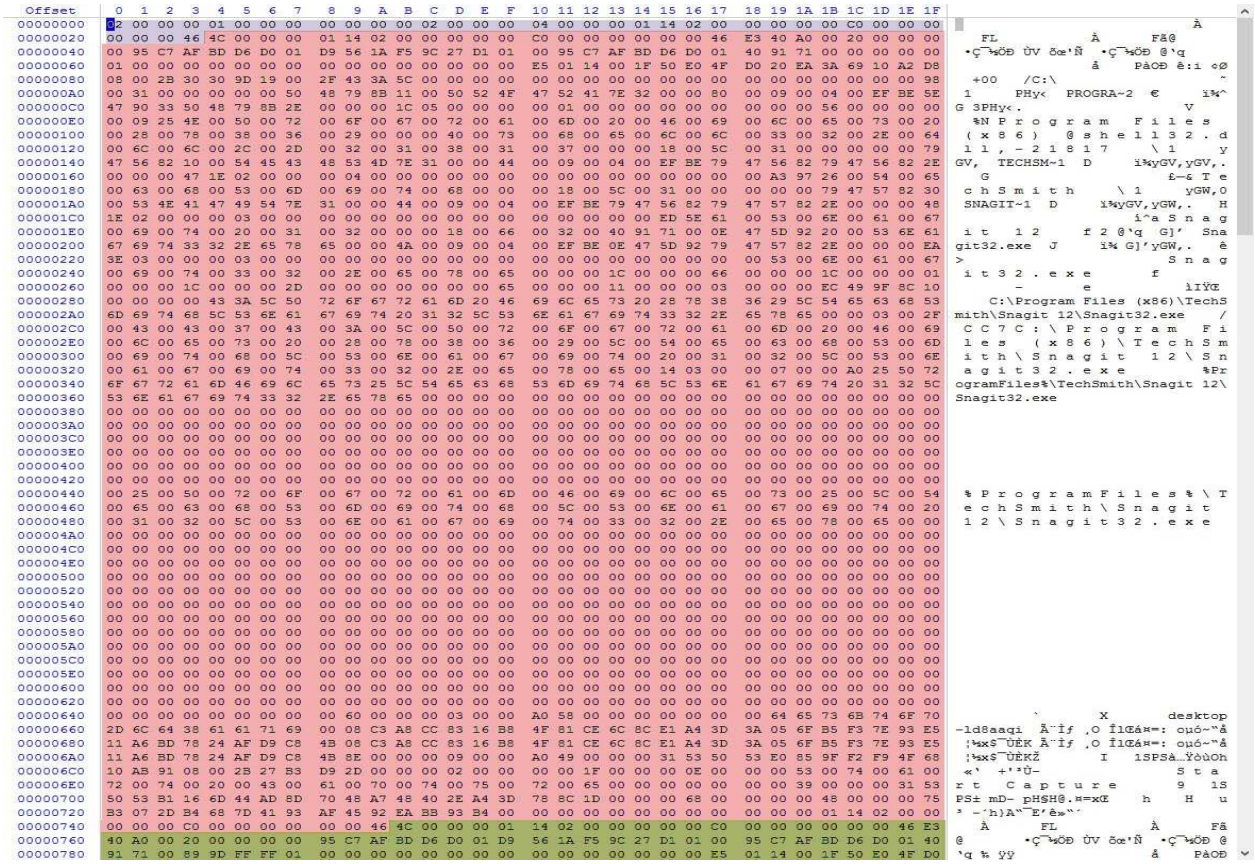
λειτουργικό σύστημα για κάθεμιά εφαρμογή βασιζόμενο στη διαδρομή του δίσκου. Έτσι μέσω του AppID μπορούμε να προσδιορίσουμε ποια εφαρμογή χρησιμοποίησε ο χρήστης. Λίστα με πολλά AppIDs μπορούμε να δούμε στις αναφορές [22] και [23]. Επίσης έχει γραφεί κι ένας calculator για AppIDs από το Hexacorn [24]. Ο αλγόριθμος παραγωγής AppIDs είναι ένα CRC-64 άθροισμα ελέγχου (checksum) που υπολογίζεται από τη διαδρομή στο δίσκο της εφαρμογής. Είναι προφανές ότι αλλαγή του path σε ένα μη προκαθορισμένο οδηγεί σε διαφορετικό AppID.

5.2 Custom Destinations

Ένα αρχείο custom Destination Jump List εσωτερικά φαίνεται ως εξής:

- ✓ Header
- ✓ Σειρά από αρχεία lnk
- ✓ Διάφορες δομές δεδομένων
- ✓ Footer (ένα signature 0xbabffbab)

Σε ένα hex editor μπορούμε να δούμε την παρόμοια εικόνα με την παρακάτω:



Σχήμα 27. Το header, το πρώτο και το δεύτερο αρχείο lnk (μπλε, ροζ, πράσινο αντίστοιχα).

Στο τέλος υπάρχει το footer:

Σχήμα 28. Footer

Από τα παραπάνω συμπεραίνουμε ότι το σημείο που θέλει προσοχή είναι ότι ο parser θα πρέπει να εξάγει σωστά τα αρχεία lnk. Με άλλα λόγια θα πρέπει να γνωρίζουμε σε ποιο σημείο σταματούν και αρχίζουν τα αρχεία lnk.

Τα αρχεία lnk έχουν τα εξής χαρακτηριστικά:

Μήκος header=0x4C

Lnk class identifier GUID=00021401-0000-0000-c000-000000000046

Σύμφωνα με τα παραπάνω, θα πρέπει να ψάξουμε μέσα στο αρχείο μια σειρά, η οποία είναι όπως αυτή: 4C 00 00 00 01 14 02 00 00 00 00 00 C0 00 00 00 00 00 00 46, όπου τα πρώτα 4 bytes είναι το header και τα υπόλοιπα το GUID. Ξέροντας το offset για καθένα από αυτά, ξέρουμε ταυτόχρονα σε ποια θέση αρχίζει κάθε αρχείο lnk. Έτσι μπορούμε να αναγνωρίσουμε όλα τα lnk αρχεία εκτός από το τελευταίο. Για αυτό θα πρέπει να βρούμε το offset του footer και να το χρησιμοποιήσουμε συνδυαστικά με το offset στο οποίο αρχίζει το προηγούμενο lnk αρχείο [27].

Έτσι λοιπόν αναγνωρίζεται κάθε αρχείο lnk, διαχωρίζεται από τα υπόλοιπα και αφού το επεξεργαστούμε με κατάλληλο εργαλείο [28], παρατηρούμε ότι κάθε αρχείο lnk έχει ένα Property block το οποίο περιέχει ένα Title. Τα Titles είναι αυτές οι εγγραφές που έχουν γίνει pinned στην Jump List.

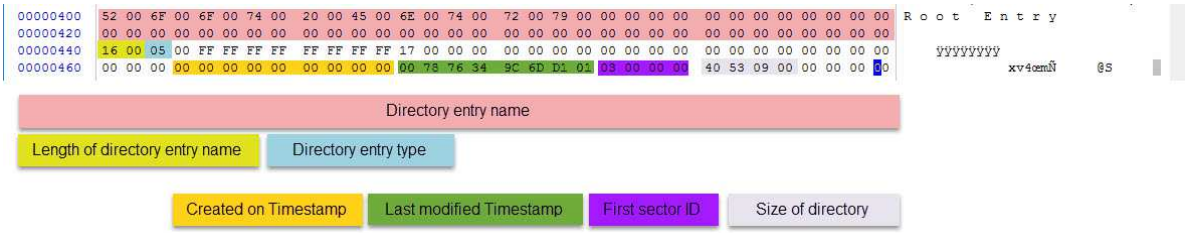
5.3 Automatic Destinations

Τα automatic Destinations Jump Lists αποθηκεύονται σε μορφή OLE (Object Linking & Embedding) CF (Compound Format). Αυτά τα αρχεία έχουν τα εξής πεδία:

- ✓ Header
- ✓ Sector Allocation Tables
- ✓ Directory

Το header έχει μήκος 512bytes και περιέχει κρίσιμη πληροφορία για το parsing του υπόλοιπου αρχείου. Περισσότερες λεπτομέρειες υπάρχουν στην αναφορά [27]. Αρκετά σημαντικό πεδίο είναι το Directory. Αυτό περιέχει εγγραφές μήκους 128bytes. Περιέχει

πληροφορίες όπως ονόματα, χρονική στιγμή δημιουργίας και τροποποίησης, ένα sector ID του πρώτου sector που περιέχει τα δεδομένα. Ενδεικτική εικόνα της δομής του είναι το σχήμα 29.

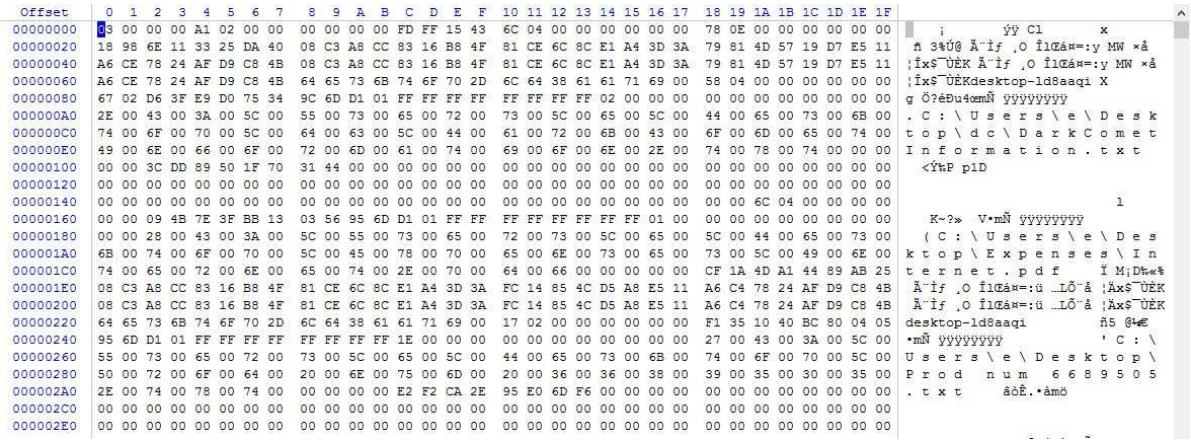


Σχήμα 29. Μια εγγραφή στο Directory μήκους 128bytes.

Ξεχωρίζουμε τα εξής πεδία:

- ✓ Όνομα: Root Entry
- ✓ Μήκος ονόματος: 22
- ✓ Τύπος: 05
- ✓ Ημερομηνία δημιουργίας: δεν έχει αποθηκευθεί
- ✓ Ημερομηνία τροποποίησης: 02/22/2016 18:09:43
- ✓ First sector ID: 3
- ✓ Μέγεθος: 611136bytes

Μια πολύ σημαντική εγγραφή Directory είναι το DestList, του οποίου η μορφή φαίνεται όπως παρακάτω:



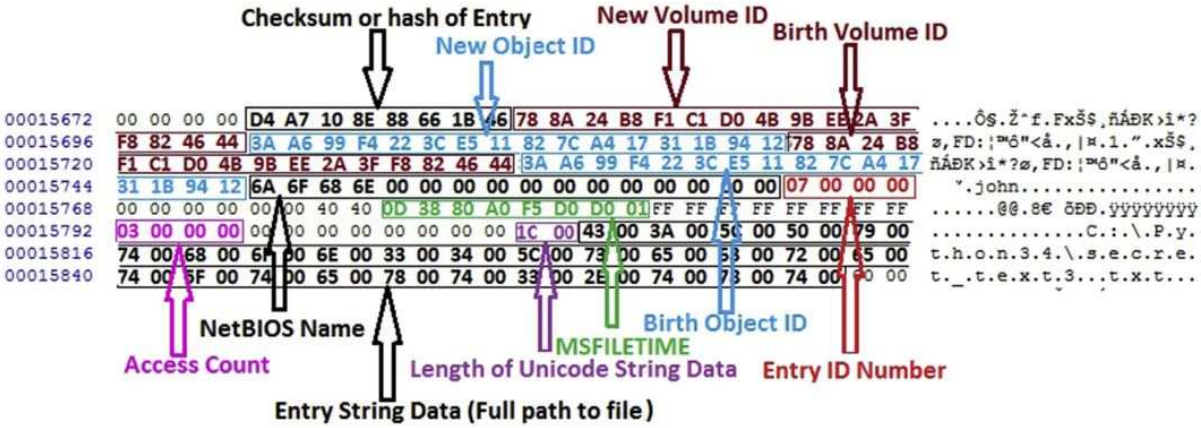
Σχήμα 30. Απεικόνιση του DestList στο Directory ενός OLE CF αρχείου.

Το DestList περιέχει ένα header των 32bytes. Αυτό έχει τα εξής πεδία:

- ✓ Version number
- ✓ Πλήθος DestList εγγραφών
- ✓ Πλήθος pinned DestList εγγραφών
- ✓ Τελευταίος αριθμός εγγραφής που χρησιμοποιήθηκε

Κάθε εγγραφή DestList περιέχει τα εξής:

- ✓ Volume Droid ID
- ✓ File Droid ID
- ✓ Birth volume Droid ID
- ✓ Birth file Droid ID
- ✓ Hostname
- ✓ Entry number
- ✓ Last modified timestamp
- ✓ Pin status
- ✓ Path size
- ✓ Path



Σχήμα 31. Η δομή εγγραφής DestList στα Windows 10.

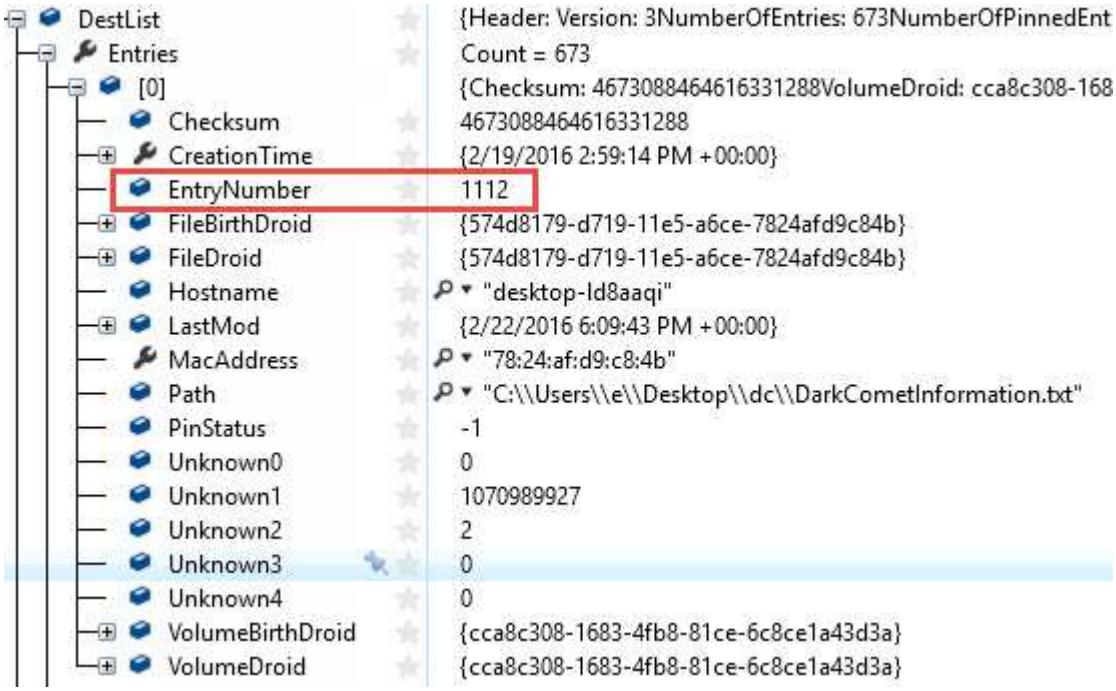
Οι Jump Lists πριν από τα Windows 10 χρησιμοποιούν version number 1, ενώ στα Windows 10 είναι 3. Επίσης άλλη μια διαφορά είναι ότι στο version number 1 το path ανιχνεύεται στο offset 114, ενώ στο version number 3 ανιχνεύεται στο offset 130.

Κάθε αρχείο lnk, στο οποίο αναφέρεται μια εγγραφή DestList, βρίσκεται ως αντικείμενο στο Directory και έχει τη δομή που είδαμε προηγουμένως για το Directory. Για να πάρουμε τα δεδομένα από τα οποία αποτελείται το lnk αρχείο, ελέγχουμε το μέγεθός του και χρησιμοποιούμε το SAT ή το SSAT για να συλλέξουμε τα bytes [27]. Η διαδικασία είναι:

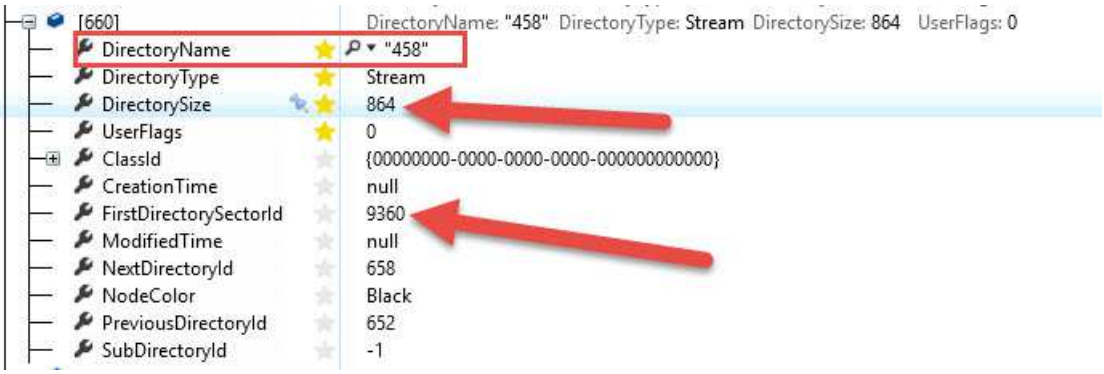
1. Επεξεργασία όλων των εγγραφών Directory
2. Εντοπισμός του DestList
3. Επεξεργασία των εγγραφών DestList
4. Για καθεμιά από αυτές, εντοπισμός της εγγραφής Directory
5. Αφού αποκτήσουμε την εγγραφή Directory για το αρχείο lnk βρίσκουμε τα δεδομένα του lnk

6. Εμφανίζουμε την πληροφορία του DestList και του lnk

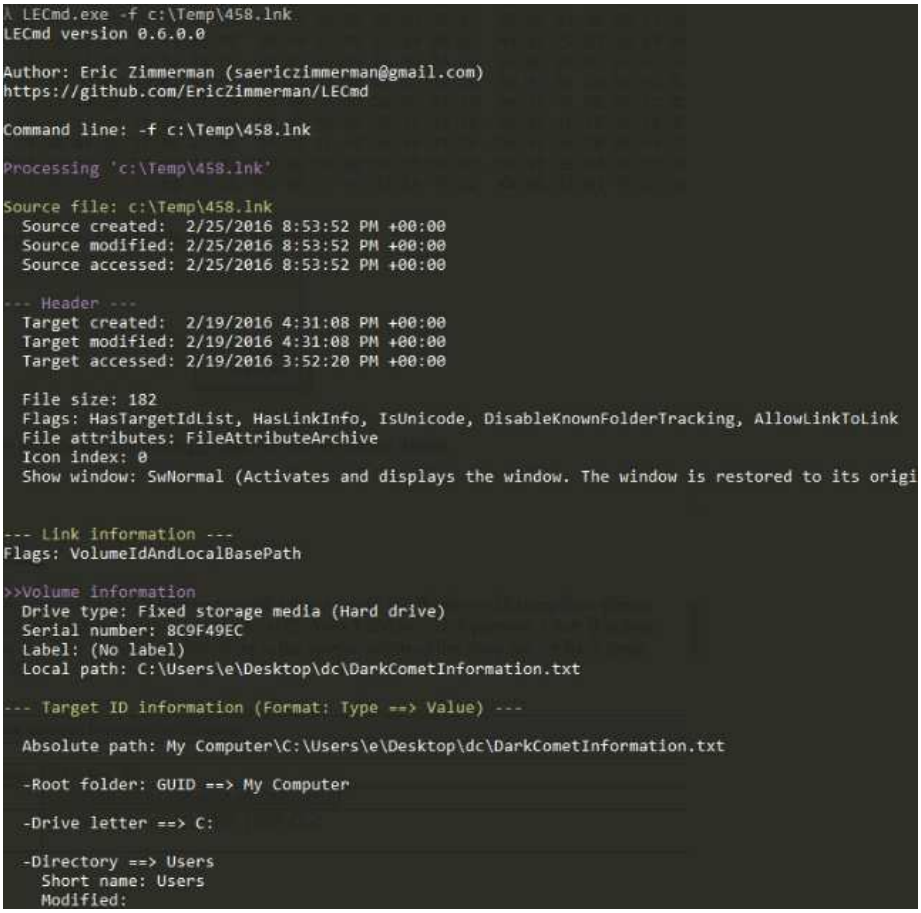
Το σημαντικό είναι το βήμα 5. Έστω ότι έχουμε μια εγγραφή DestList με αριθμό 1112 (ή 0x458). Θα πρέπει να βρούμε μια εγγραφή Directory με Directory Name 458 (σχήματα 32 και 33). Το μέγεθος της εγγραφής Directory είναι 864bytes και με χρήση του SSAT παίρνουμε τα δεδομένα του lnk και τέλος με κάποιο εργαλείο όπως στο [28] οπτικοποιούμε την πληροφορία από το αρχείο lnk.



Σχήμα 32. DestList entry number 1112



Σχήμα 33. Directory Name 458.



Σχήμα 34. Πληροφορίες από το αρχείο lnk.

6. Prefetch (Superfetch) files

Για πρώτη φορά εισήχθησαν στα Windows XP με σκοπό να επιταχύνουν την εκκίνηση του συστήματος και των εφαρμογών. Ο Windows Prefetcher δημιουργεί ένα prefetch αρχείο τη στιγμή που η εφαρμογή εκκινείται για πρώτη φορά. Η διαδικασία του prefetch, τα αρχεία, η δομή τους και η χρησιμότητά τους στην ανάλυση ψηφιακών πειστηρίων είναι σημαντική κι έχει επισημανθεί από διάφορους αναλυτές στο παρελθόν [30], [31], [32]. Τα prefetch files περιέχουν τις παρακάτω πολύτιμες πληροφορίες Ενδεικτικά αναφέρουμε ότι μπορούμε να προσδιορίσουμε ποια προγράμματα εκτελέστηκαν, το όνομα και τη διαδρομή στο δίσκο του εκτελέσιμου, καθώς και τα συσχετιζόμενα DLLs [33]. Τα prefetch files φυλάσσουν:

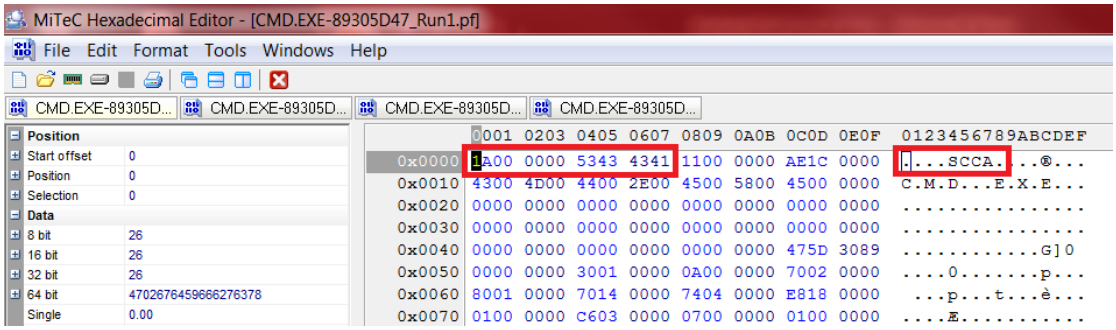
- ✓ Τα file paths όλων των αρχείων στα οποία η εφαρμογή είχε πρόσβαση στα πρώτα 10 δευτερόλεπτα της εκτέλεσής της
- ✓ Το πλήθος των εκτελέσεων της εφαρμογής
- ✓ Την τελευταία χρονική στιγμή εκτέλεσης της εφαρμογής
- ✓ Τους δίσκους και τους φακέλους που προσπελάστηκαν

6.1 Δομή των Prefetch files

Ένα prefetch αρχείο ονομάζεται με το όνομα της εφαρμογής, μια παύλα και ένα hash 8 χαρακτήρων ακολουθούμενο από την επέκταση αρχείου .pf. Το hash υπολογίζεται από το λειτουργικό σύστημα με βάση τη διαδρομή στο δίσκο από την οποία εκτελέστηκε η εφαρμογή. Επομένως μια εφαρμογή η οποία εκτελέστηκε από δύο διαφορετικές τοποθεσίες στο δίσκο θα δημιουργήσει δύο διαφορετικά αρχεία .pf. Από τα Windows 7 μέχρι και τα 8.1, τα αρχεία prefetch έχουν μια σταθερή υπογραφή 4bytes (offset 0x04) 0x41434353 (SCCA), η οποία προηγείται του format version identifier μεγέθους 4bytes. Η Microsoft αλλάζει το format version σε κάθε καινούρια έκδοση των Windows αλλά η δομή των prefetch files παραμένει η ίδια. Η μόνη αλλαγή στα Windows 8, ήταν η προσθήκη επτά νέων τιμών timestamps. Επίσης στα Windows 10 τα αρχεία αυτά είναι συμπιεσμένα.

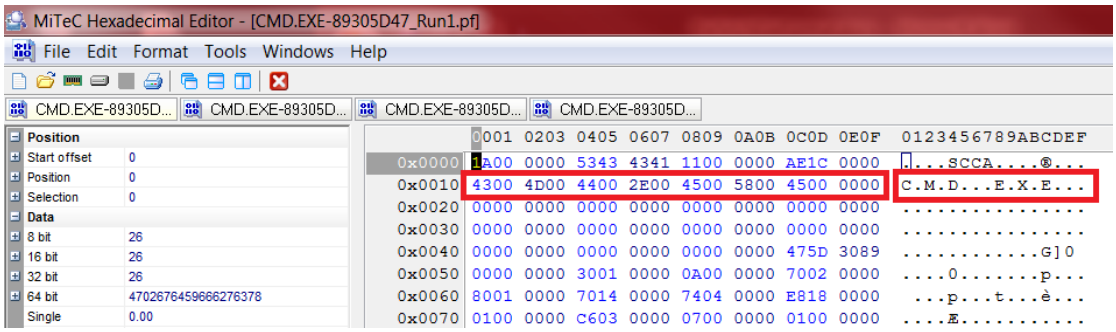
Για τα Windows 8 έχουμε ένα prefetch file header με offset 0x00 μήκους 8bytes. Η πρώτη τιμή είναι η file signature, η οποία περιέχει την τιμή (0x1A, 0x00, 0x00, 0x00, 0x53, 0x43, 0x43, 0x41). Τα πρώτα υποδεικνύουν την έκδοση του λειτουργικού συστήματος: 0x11:

Win XP, 0x17: Win 7, 0x1A: Win 8, 0x1E: Win 10. Η ASCII αναπαράσταση των τελευταίων 4bytes είναι SCCA και αυτή είναι η file signature (σχήμα 35).



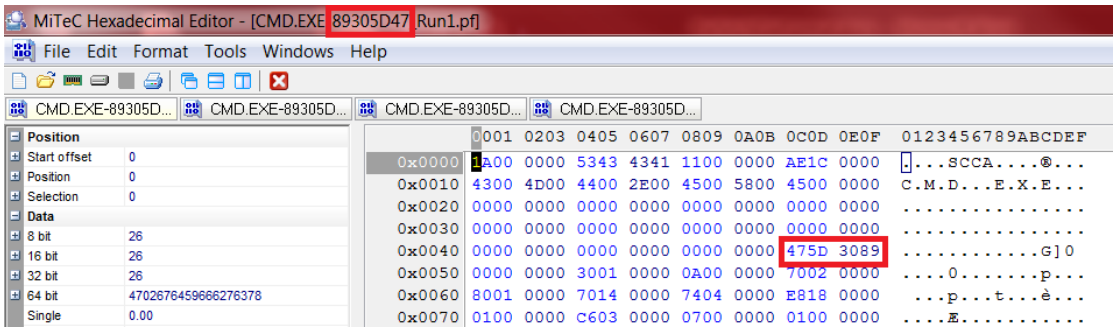
Σχήμα 35. Prefetch file header.

Ακολούθως, στο offset 0x10 βλέπουμε μια Unicode συμβολοσειρά CMD.EXE. Αυτή η τιμή αντιπροσωπεύει το όνομα της εφαρμογής για την οποία δημιουργήθηκε το συγκεκριμένο prefetch file. Εδώ θα πρέπει να γίνει αντιπαραβολή αυτού του ονόματος με αυτό που αναγράφεται στον όνομα του αρχείου .pf ώστε να εξασφαλιστεί ότι είναι τα ίδια και κάποιος κακόβουλος χρήστης δεν έχει εσκεμμένα αλλάξει το όνομα. Η τιμή έχει μέγεθος 60bytes ώστε να επιτρέπονται μεταβλητά μήκη στα ονόματα των εφαρμογών. Το όνομα της εφαρμογής τελειώνει με τιμή null (0x0000).



Σχήμα 36. Όνομα της εφαρμογής.

Επιπρόσθετα έχουμε το file path hash με offset 0x4C και μέγεθος 4bytes. Από αυτό μπορεί να ανιχνευθεί ένα κρυμμένο malware στο σύστημα. Επειδή οι επιτιθέμενοι δίνουν κοινά ονόματα στο malware π.χ. svchost.exe, αν αυτό δεν τρέχει από τον προκαθορισμένο κατάλογο που θα έπρεπε C:\Windows\System32 και εκτελείται από το C:\Windows τότε το αντίστοιχο .pf αρχείο δε θα έχει το hash που θα έπρεπε στο όνομά του.



Σχήμα 37. File path hash.

Ξέχωρα από τα παραπάνω έχουμε το application run count με offset 0x0D μεγέθους 4bytes, στο οποίο φυλάσσεται πόσες φορές εκτελέστηκε μια εφαρμογή και το last access timestamp με offset 0x80 μεγέθους 8bytes. Το τελευταίο δίνει πληροφορία στον αναλυτή για το πότε ήταν η τελευταία φορά που εκτελέστηκε μια εφαρμογή. Η τιμή αποθηκεύεται ως FILETIME object και σύμφωνα με τη Microsoft είναι το πλήθος των διαλειμμάτων διάρκειας 100nanoseconds που έχουν παρέλθει από την 12:00 A.M. January 1, 1601 Coordinated Universal Time (UTC). Αντίθετα με τα Windows 7, στα Windows 8 και 10 έχουμε 8 last file access timestamps.

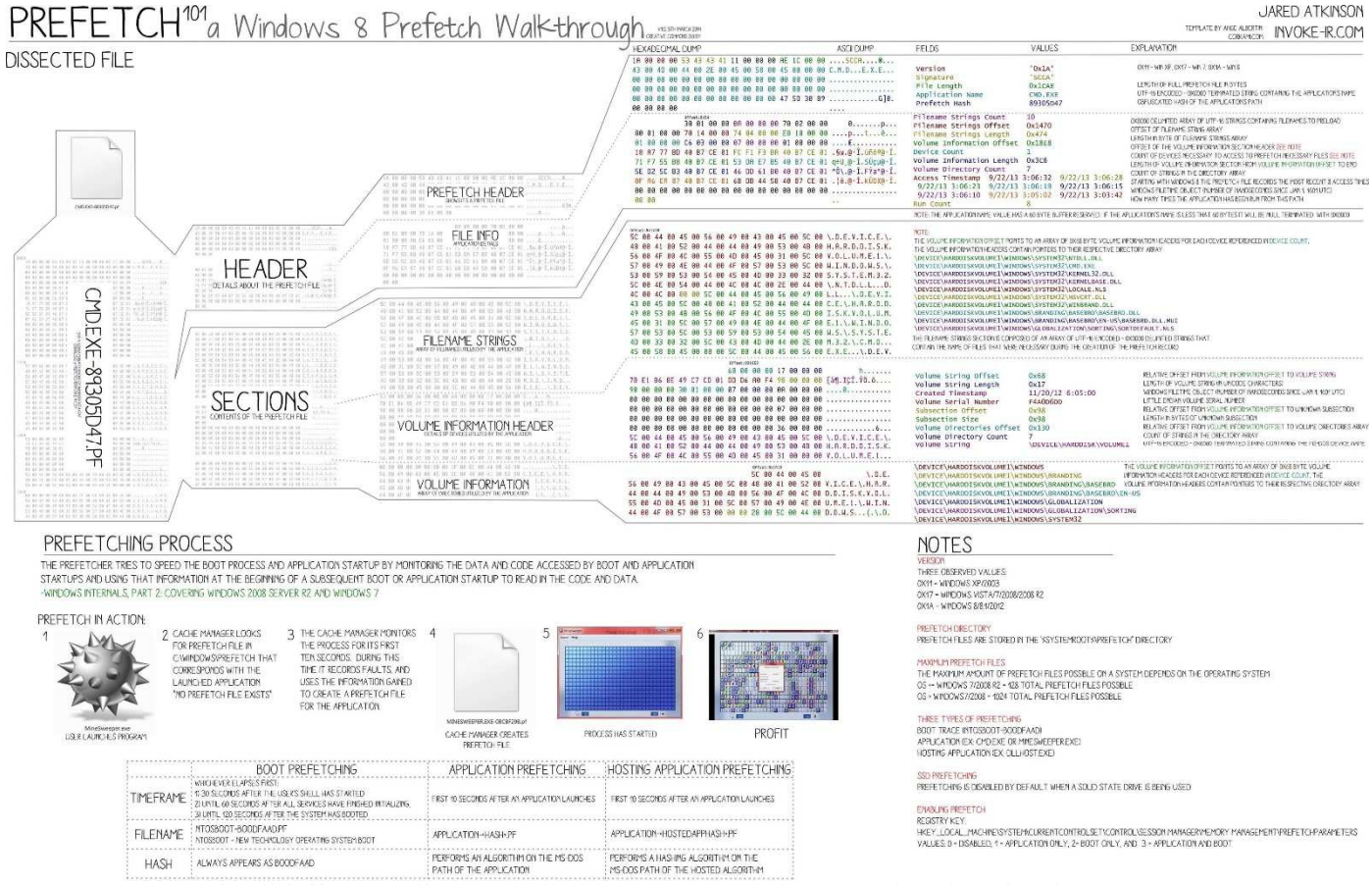
Αξιοσημείωτο είναι το γεγονός ότι στα Windows 10 ο αναλυτής θα πρέπει να αποσυμπιέσει το prefetch αρχείο προτού το κάνει parse. Η συμπίεση γίνεται με τον αλγόριθμο Xpress Huffman, γνωστό και ως MAM format. Τα πρώτα 4bytes αποτελούν το signature, ενώ το τέταρτο byte υποδεικνύει τον αλγόριθμο συμπίεσης που χρησιμοποιήθηκε. Τα επόμενα 4bytes με offset 0x04 παρέχουν το μέγεθος μετά την αποσυμπίεση. Το Windows API το οποίο είναι υπεύθυνο για την αποσυμπίεση είναι το *RtlDecompressBufferEx*, το οποίο είναι διαθέσιμο από τα Windows 8.1. Υπάρχει ένα διαθέσιμο script ελεύθερα διαθέσιμο για την αποσυμπίεση των prefetch files των Windows 10 [30].

Τα κοινά χαρακτηριστικά μεταξύ των Windows αποτυπώνονται στον παρακάτω πίνακα.

Χαρακτηριστικά	**Περιγραφή**
Byte order	Little endian
χρονοσημάνσεις	UTC
Συμβολοσειρά	UTF-16
Τοποθεσία	C:\Windows\Prefetch
Όνομα αρχείου	<όνομα εκτελέσιμου>-<hash>.pf
Header	Offset 04, μήκος 4bytes (SCCA (0x53, 0x43, 0x43, 0x41))

Όνομα αρχείου Unicode	Offset 16, μήκος 30bytes
Ώρα τελευταίας εκτέλεσης	Offset 128, 8bytes
Volume ID	Offset 108, μήκος 4bytes δείχνει στο offset του section D του prefetch file. Το Volume ID βρίσκεται στο Offset του section D + 16bytes, με μήκος 4bytes

Τέλος παραθέτουμε το παρακάτω ποστερ!



http://3.bp.blogspot.com/-hUnqwqJwWcI/UxgCUSFt7RI/AAAAAAAAAQo/iNwKio4RazU/s1600/Prefetch8101low.png

7. Bitlocker

Το Bitlocker πρωτοεμφανίστηκε στα Windows Vista και παρέχει πλήρη κρυπτογράφηση του σκληρού δίσκου. Χρησιμοποιεί τον αλγόριθμο AES για την κρυπτογράφηση σε CBC ή σε XTS mode με κλειδί μεγέθους 128 ή 256bits. Όταν χρησιμοποιείται συνδυαστικά με ένα TPM chip μπορεί να διασφαλίσει την ακεραιότητα της εκκίνησης και των αρχείων του συστήματος πριν αποκρυπτογραφήσει το δίσκο [35]. Ανεπιτυχής επαλήθευση σημαίνει ότι το σύστημα δεν εκκινεί. Επίσης υπάρχει η δυνατότητα κρυπτογράφησης αφαιρούμενων μέσων αποθήκευσης (Bitlocker To Go) με χρήση συνθηματικού ή smart card. Προαιρετικά υπάρχει η δυνατότητα χρήσης ενός diffuser αλγορίθμου, ο οποίος ονομάζεται Elephant [36].

Το κλειδί που εκτελεί την παραπάνω διαδικασία είναι το Full Volume Encryption Key (FVEK) ή/και το TWEAK Key και φυλάσσεται στα metadata του bitlocker. Το FVEK ή/και το TWEAK κρυπτογραφούνται χρησιμοποιώντας ένα άλλο κλειδί, το Volume Master Key (VMK). Αρκετά αντίγραφα του VMK φυλάσσονται επίσης στα metadata. Καθένα από αυτά τα αντίγραφα κρυπτογραφείται χρησιμοποιώντας ένα άλλο κλειδί, γνωστό ως key-protector. Οι key protectors μπορεί να είναι:

- ✓ Trusted Platform Module (TPM)
- ✓ Smart card
- ✓ Recovery password
- ✓ Start-up key
- ✓ User password

7.1 Ανίχνευση κρυπτογραφημένου δίσκου

Οι δίσκοι που έχουν κρυπτογραφηθεί με το Bitlocker έχουν διαφορετικό signature από το τυπικό NTFS header. Ο κρυπτογραφημένος δίσκος αρχίζει με το signature "-FVE-FS-". Τυπικά μπορεί να δείχνει ως:

```
00000000  eb 58 90 2d 46 56 45 2d  46 53 2d 00 02 08 00 00  |.X.-FVE-FS-.....|
00000010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 00 00 00  |........?.......|
00000020  00 00 00 00 e0 1f 00 00  00 00 00 00 00 00 00 00  |................|
00000030  01 00 06 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  80 00 29 00 00 00 00 4e  4f 20 4e 41 4d 45 20 20  |..)....NO NAME  |
00000050  20 20 46 41 54 33 32 20  20 20 33 c9 8e d1 bc f4  |  FAT32   3.....|
```

Σχήμα 38. Header κρυπτογραφημένου δίσκου

Αυτοί οι δίσκοι μπορούν επίσης να αναγνωριστούν από το GUID.

- ✓ GUID για το Bitlocker: 4967d63b-2e29-4ad8-8399-f6a339e3d00
- ✓ GUID για το Bitlocker To Go: 4967d63b-2e29-4ad8-8399-f6a339e3d01

Αντίστοιχα θα βλέπαμε το εξής:

```
000000a0  3b d6 67 49 29 2e d8 4a  83 99 f6 a3 39 e3 d0 01  |;.gI)..J....9...|
```

Σχήμα 39. Αναγνώριση κρυπτογραφημένου δίσκου από το GUID.

7.2 Ανάκτηση κλειδιών

Η ανάκτηση κλειδιών στα Windows 7 μπορεί να γίνει παίρνοντας από τη μνήμη το image και από το FVE βρίσκουμε ακριβώς που είναι το FVEK σε αυτή. Στα Windows 8 όμως οι μηχανισμοί κρυπτογράφησης γίνονται από το module CNG (Cryptography Next Generation), το οποίο αντικαθιστά το CryptoAPI που τους εκτελούσε εσωτερικά στο FVEvol driver. Επομένως η ίδια προσέγγιση δε θα έχει κανένα αποτέλεσμα στα Windows 8.

Χρησιμοποιώντας το debugger WinDbg της Microsoft μπορούμε να δούμε τις λειτουργίες κρυπτογράφησης/αποκρυπτογράφησης σε Windows 7 και 8.1. Στην πρώτη περίπτωση αυτές λαμβάνουν χώρα εντός του FVEvol, ενώ στη δεύτερη περίπτωση αναλαμβάνει το CNG.

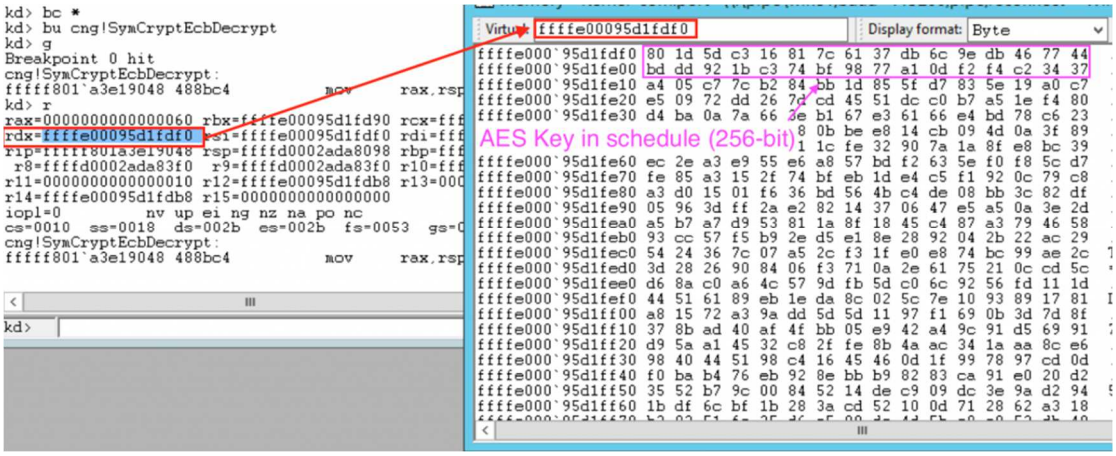
```
Child-SP          RetAddr           Call Site
fffff880`035a2ae8 fffff880`019aa153 fvevol!AesEncrypt
fffff880`035a2af0 fffff880`019aa3d1 fvevol!FveAesCcmDecrypt+0x7fb
fffff880`035a2b40 fffff880`019a68b0 fvevol!FveCryptElephantDecrypt+0x55
fffff880`035a2b90 fffff880`019a6c99 fvevol!FveFilteredRead+0x58
fffff880`035a2bc0 fffff880`019a3d39 fvevol!FvepParseBlockAndFilter+0x109
fffff880`035a2c00 fffff880`019a451d fvevol!ReadDecrypt+0x1c1
fffff880`035a2d00 fffff800`02926cce fvevol!FveWorkerStart+0x71
fffff880`035a2d40 fffff800`0267afe6 nt!PspSystemThreadStartup+0x5a
fffff880`035a2d80 00000000`00000000 nt!KiStartSystemThread+0x16
```

Σχήμα 40. Κρυπτογραφικές συναρτήσεις στα Windows 7.

```
Child-SP          RetAddr           Call Site
ffffd001`dd3cd048 fffff800`cae2b43a cng!SymCryptAesDecrypt
ffffd001`dd3cd050 fffff800`cae23d00 cng!SymCryptEcbDecrypt+0x62
ffffd001`dd3cd0a0 fffff800`cae23e07 cng!MSBlockDecrypt+0x38e
ffffd001`dd3cd210 fffff800`cae2391a cng!MSCryptDecrypt+0xd7
ffffd001`dd3cd270 fffff800`cbb1ede5 cng!BCryptDecrypt+0xea
ffffd001`dd3cd320 fffff800`cbb1de71 fvevol!FveAesEcbDecrypt+0x71
ffffd001`dd3cd380 fffff800`cbb1ed6f fvevol!FveAesCbcDecryptSectorsPageIsolation+0x131
ffffd001`dd3cd8b0 fffff800`cbb1ed22 fvevol!FveAesCbcDecryptSectors+0x2f
ffffd001`dd3cd900 fffff800`cbb1ecb5 fvevol!FveDecryptRegion+0x3e
ffffd001`dd3cd950 fffff800`cbb1e762 fvevol!FveFilteredRead+0x4d
ffffd001`dd3cd990 fffff800`cbb7c668 fvevol!FvepParseBlockAndFilter+0x1a2
ffffd001`dd3cda20 fffff800`cbb7af6d fvevol!ReadDecrypt+0x20c
ffffd001`dd3cdb70 fffff800`cbb7aa39 fvevol!FveCryptoWorker+0x51
ffffd001`dd3cdbe0 fffff801`93f67b2f fvevol!FveWorkerKmTP+0x89
ffffd001`dd3cdc20 fffff801`93f2888f nt!IopProcessWorkItem+0xfb
ffffd001`dd3cdc90 fffff801`93f7e280 nt!ExpWorkerThread+0x69f
ffffd001`dd3cdd40 fffff801`93fd79c6 nt!PspSystemThreadStartup+0x58
ffffd001`dd3cdda0 00000000`00000000 nt!KiStartSystemThread+0x16
```

Σχήμα 41. Κρυπτογραφικές συναρτήσεις στα Windows 8.1.

Τώρα μπορούμε να ανακτήσουμε το FVEK από τη δεύτερη παράμετρο στο SymCryptEcbDecrypt, το οποίο περιέχει ένα δείκτη στη θέση του κλειδιού στη μνήμη. Ο καταχωρητής rdx έχει το δείκτη στη θέση μνήμης που βρίσκεται το FVEK, επομένως θέτουμε ένα breakpoint στο cng!SymCryptEcbDecrypt για να πάρουμε την τιμή του rdx.



Σχήμα 42. Ανάκτηση του FVEK από τη μνήμη στα Windows 8.1.

Όπως αναφέραμε σε προηγούμενη παράγραφο, τα Windows 10 χρησιμοποιούν τον αλγόριθμο XTS AES. Θέτοντας ένα breakpoint στο cng!SymCryptXtsAesDecrypt ελέγχουμε τον καταχωρητή rcx.

Επιπλέον, το μέγεθος των Bitlocker allocations στη μνήμη φαίνεται ότι είναι αρκετά συμπαγές υποδεικνύοντας έτσι πιθανές περιοχές κλειδιών. Το μέγεθος του Cngb φαίνεται ότι είναι 672, αδιακρίτως του τύπου κρυπτογράφησης. Το Volatility με χρήση του plugin poolpeek μπορεί να εντοπίσει τις παραπάνω περιοχές.

```
tribalchicken@instance-1:~$ python volatility/vol.py -f Win81_256.dmp --profile=Win8SP1x64 poolpeek -t Cngb -m 672 -M 672
Volatility Foundation Volatility Framework 2.5
Pool Header: 0xe00095d0b630, Size: 672
0xe00095d0b630  09 00 2a 02 43 6e 67 62 00 00 00 00 00 00 00 00   ..*.Cngb........
0xe00095d0b640  20 00 00 00 52 55 55 55 20 11 d2 95 00 e0 ff ff   ....RUUU........
0xe00095d0b650  60 b6 d0 95 00 e0 ff ff 40 b6 d0 95 00 e0 ff ff   `.......@.......
0xe00095d0b660  6e 02 00 00 4b 53 53 4d 02 00 01 00 01 00 00 00   n...KSSM........
0xe00095d0b670  10 00 00 00 10 00 00 00 00 00 01 00 00 00 00 00   ................
0xe00095d0b680  c0 23 d1 95 00 e0 ff ff c8 28 db c1 98 fb ee 8c   .#.......(......
0xe00095d0b690  6a 34 9d 09 2b 48 4f 32 20 00 00 00 80 1d 5d c3   j4..+HO2......].
0xe00095d0b6a0  16 81 7c 61 37 db 6c 9e db 46 77 44 bd dd 92 1b   ..|a7.l..FwD....
0xe00095d0b6b0  c3 74 bf 98 77 a1 0d f2 f4 c2 34 37 00 00 00 00   .t..w.....47....
0xe00095d0b6c0  80 1d 5d c3 16 81 7c 61 37 db 6c 9e db 46 77 44   ..]...|a7.l..FwD
0xe00095d0b6d0  bd dd 92 1b c3 74 bf 98 77 a1 0d f2 f4 c2 34 37   .....t..w.....47
0xe00095d0b6e0  a4 05 c7 7c b2 84 bb 1d 85 5f d7 83 5e 19 a0 c7   ...|....._..^...
0xe00095d0b6f0  e5 09 72 dd 26 7d cd 45 51 dc c0 b7 a5 1e f4 80   ..r.&}.EQ.......
```

Σχήμα 43. Περιοχές μνήμης που βρίσκονται τα κλειδιά.

Το offset 0x68 φαίνεται ότι αναγνωρίζει κλειδί μεγέθους είτε 128bit είτε 256bit. Η τιμή 10 υποδεικνύει ότι ακολουθεί κλειδί μήκους 128bits, ενώ η τιμή 20 κλειδί μήκους 256bits. Το ενδιαφέρον είναι ότι στην ίδια περιοχή υπάρχουν κι άλλα κλειδιά εκτός του FVEK, ίσως το VMK [34].

8. USB Activity

Η χρονική στιγμή της εισαγωγής μιας συσκευής σε ένα σύστημα και της αποσύνδεσής της μπορεί να δώσει σημαντική πληροφορία για τη δραστηριότητα του κακόβολου (κι όχι μόνο) χρήστη. Για παράδειγμα, ο αναλυτής μπορεί να υπολογίσει πόση ώρα το USB ήταν συνδεδεμένο από τις χρονικές στιγμές εισαγωγής και εξαγωγής του. Αυτή η πληροφορία μπορεί να συνδυαστεί με τη χρονική στιγμή μεταφοράς αρχείων μεταξύ του USB και του δίσκου. Αν ταυτιστούν μπορεί να εξαχθεί το συμπέρασμα ότι το συγκεκριμένο USB χρησιμοποιήθηκε για αυτή τη μεταφορά.

Όταν ένα USB συνδέεται σε ένα σύστημα, τα Windows ενημερώνουν τη registry και τα αρχεία event logs [38]. Στη συνέχεια θα δούμε τη συμπεριφορά των διαφόρων εκδόσεων των Windows όταν συνδέεται στο σύστημα ένα USB, ποια artifacts μπορούν να συγκεντρωθούν από τα Windows 10 σε σύγκριση με τα Windows 7 και 8. Τα πρωτόκολλα που θα αναλυθούν είναι το Mass Storage Class (MSC), το Picture Transport Protocol (PTP) και το Media Transport Protocol (MTP).

8.1 MSC, PTP & MTP

Οι συσκευές USB είναι προσβάσιμες από το λειτουργικό σύστημα ως αφαιρούμενα μέσα λόγω αυτών των πρωτοκόλλων. Το MSC καθορίζει το πρότυπο για την επικοινωνία μεταξύ του λειτουργικού συστήματος και του USB. Χρησιμοποιείται για τη μεταφορά αρχείων από αφαιρούμενες συσκευές όπως: εξωτερικοί δίσκοι, οπτικά μέσα, flash memory cards, κινητά τηλέφωνα, media players. Το PTP χρησιμοποιείται από εκτυπωτές, σαρωτές και άλλες ψηφιακές φωτογραφικές συσκευές για τη μεταφορά αρχείων εικόνας. Το MTP είναι μια επέκταση του PTP και επιτρέπει τη μεταφορά αρχείων ήχου και πολυμέσων από media players στον υπολογιστή. Το MTP και PTP μοιράζονται το ίδιο class code.

8.2 Πληροφορία στη registry

Η registry φυλάσσει πολύτιμη πληροφορία για τις USB συσκευές. Μεταξύ των άλλων artifacts συμπεριλαμβάνονται ο κατασκευαστής, η χρονική στιγμή εγκατάστασης,

τελευταία εισαγωγή και αφαίρεση προς και από το λειτουργικό σύστημα. Τα πιο σημαντικά registry hives είναι τα παρακάτω:

- ✓ HKEY_LOCAL_MACHINE\SYSTEM
- ✓ HKEY_LOCAL_MACHINE\SAM
- ✓ HKEY_LOCAL_MACHINE\Software
- ✓ HKEY_LOCAL_MACHINE\Security
- ✓ HKEY_CURRENT_CONFIG
- ✓ HKEY_CURRENT_USER
- ✓ HKEY_USERS\DEFAULT

Ιδιαιτέρως στο HKEY_LOCAL_MACHINE\SYSTEM τα USBSTOR, USB και Device Classes αφορούν τη δική μας μελέτη.

8.3 Setupapi.dev log

Ένα από τα πιο σημαντικά αρχεία log είναι το setupapi.dev το οποίο φυλάσσει τη χρονική στιγμή κατά την οποία συνδέθηκε μια συσκευή στο σύστημα. Τα Windows 10 έχουν το SetupAPI το οποίο καταγράφει την εγκατάσταση συσκευής. Το Plug and Play (PnP) Manager κρατάει πληροφορίες για τις καινούριες συσκευές που συνδέονται και εγκαθίστανται στο αρχείο setupapi.dev. Εδώ μπορούμε να βρούμε πληροφορίες για τη χρονική στιγμή της πρώτης εγκατάστασης του driver και πληθώρα πληροφοριών για τη συσκευή, όπως το σειριακό αριθμό, το product ID και το Vendor ID. Σύμφωνα με τις προκαθορισμένες ρυθμίσεις το setupapi.dev βρίσκεται στο φακελο INF των Windows. Σημαντικά αρχεία και φάκελοι είναι τα παρακάτω:

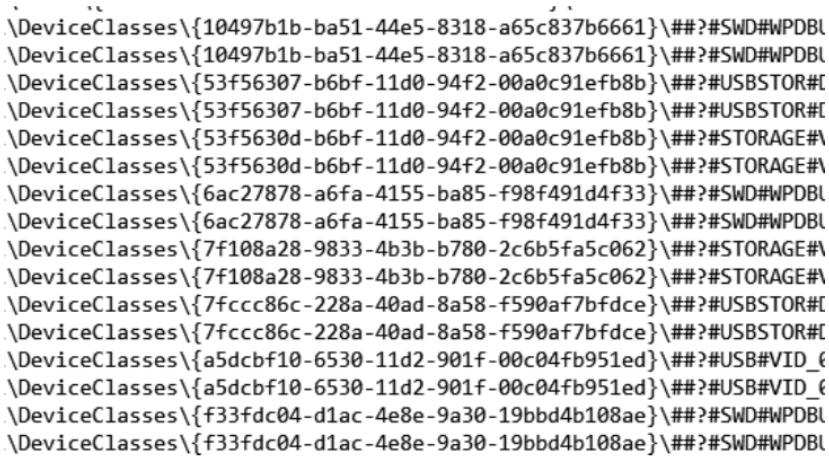
- ✓ Event Logs: C:\Windows\System32\winevt\Logs\
- ✓ Registry hives: C:\Windows\System32\config\
- ✓ PnP Manager logs: C:\Windows\inf\setupapi.dev.log

8.4 Συλλογή artifacts

Η συλλογή των artifacts για ανάλυση γίνεται σε τρεις φάσεις: πριν από τη σύνδεση USB, κατά τη διάρκεια που αυτό είναι συνδεδεμένο, μετά από την αφαίρεση του USB.

Device Classes

Όταν ένας driver PnP «φορτώνεται» το DeviceClasses subkey προστίθεται στη registry. Κάτω από αυτό δημιουργούνται νέα κλειδιά όταν ένα USB συνδέεται στο σύστημα για πρώτη φορά (σχήμα 44).

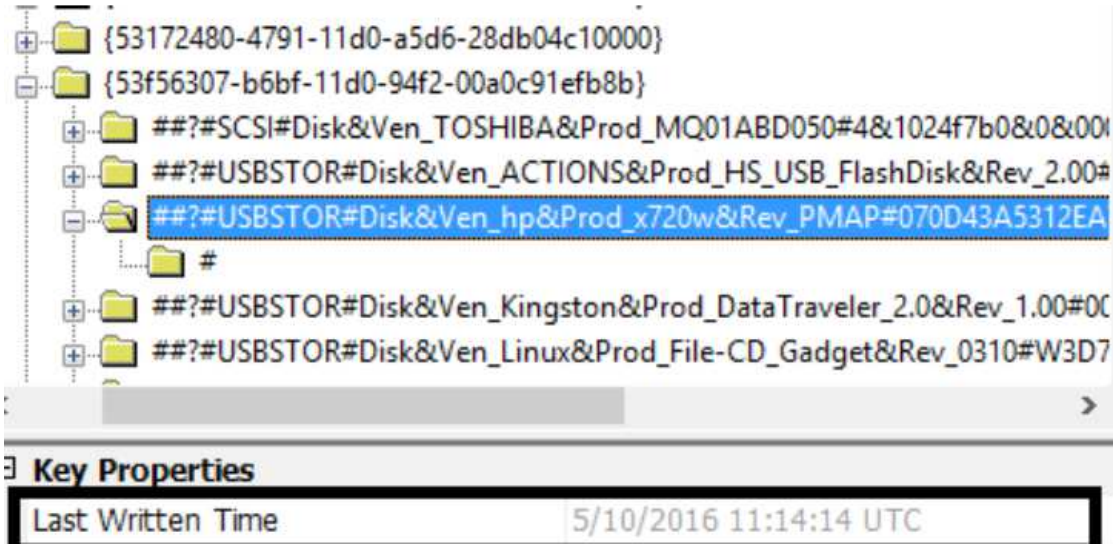


Σχήμα 44. DeviceClasses

Τα DeviceClasses των συσκευών MSC βρίσκονται κάτω από το HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses\ και είναι:

- ✓ 10497b1b-ba51-44e5-8318-a65c837b6661
- ✓ 53f56307-b6bf-11d0-94f2-00a0c91efb8b
- ✓ 53f5630d-b6bf-11d0-94f2-00a0c91efb8b
- ✓ 6ac27878-a6fa-4155-baS5-f9Sf491d4f33
- ✓ 7f108a28-9833-4b3b-b780-2c6b5fa5c062
- ✓ 7fccc86c-228a-40ad-8a58-f590af7bfdce
- ✓ a5dcbf10-6530-11d2-901f-00c04fb951ed
- ✓ f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae

Η μορφή του κλειδιού το οποίο παράγεται στο DeviceClasses περιλαμβάνει:
#USBSTOR#Disk&Ven_[VendorName]&Prod_[ProductName]&Rev_PMAP#[SerialNo.]#{[DeviceClassID]}
Ο σειριακός αριθμός είναι μοναδικός για κάθε USB. Το DeviceClasses που δημιουργούνται κατά την πρώτη σύνδεση του USB είναι ίδια για τα Windows 8 και 10. Αυτά χρησιμοποιούνται για να βρεθεί η χρονική στιγμή της πρώτης σύνδεσης στο σύστημα και ο αναλυτής μπορεί να τα δει με το Access Data Registry Viewer (σχήμα 45). Η τελευταία ώρα που υπάρχει σε κάθε DeviceClasses συσχετίζεται με την πρώτη χρονική στιγμή που συνδέθηκε το USB. Η ώρα φαίνεται σε μορφή 64bit FILETIME.

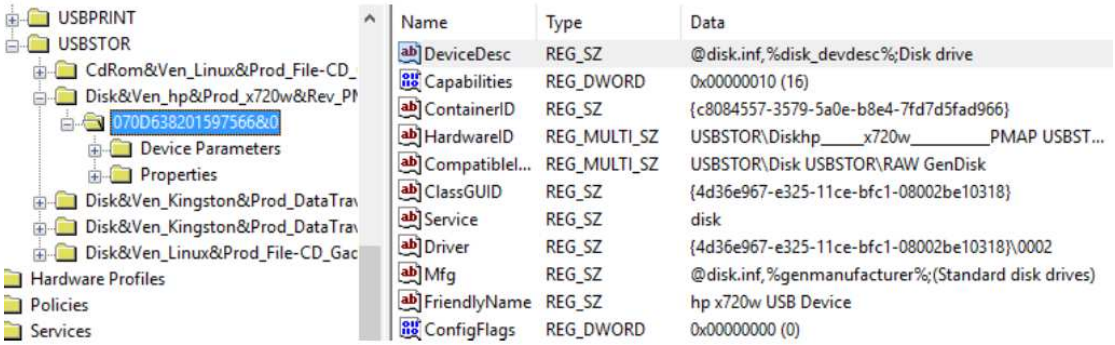


Σχήμα 45. Πρώτη σύνδεση για το DeviceClass 53f56307-b6bf-11d0-94f2-00a0c91efb8b.

Device Specification

USBSTOR key [37]:

Στο registry viewer επιλέγουμε το κλειδί HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_[VendorName]&Prod_[ProductName]&Rev_PMAP\[SerialNo] για να προσδιορίσουμε τον κατασκευαστή, το σειριακό αριθμό, το product ID, το container ID, κλπ. (σχήμα 46)



Σχήμα 46. Χαρακτηριστικά συσκευής από το USBSTOR key.

USB key:

Η διαδρομή στο κλειδί αυτό είναι

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB\VID_[VendorID]&PID_[ProductID]\[SerialNo]

Στα Windows 10 προστίθεται ένα νέο κλειδί USB\VID_[VendorID]&PID_[ProductID]\[SerialNo]\ Device Parameters\e5b3b5ac9725-4f78-963f-03dfb1d828c7

Setupoapi.dev.log:

Όταν μια συσκευή σύνδέεται για πρώτη φορά στο σύστημα, ο PnP Manager εγκαθιστά τον driver και δημιουργεί μια εγγραφή στο αρχείο setupapi.dev. το σχήμα 47 δείχνει τη χρονική στιγμή εγκατάστασης. Με παρόμοιο τρόπο το artifact λαμβάνεται και από τα Windows 8.

```
>>> [Device Install (Hardware initiated) - SWD\WPDBUSENUM\_??_USBSTOR#Disk&Ven_hp&Prod_x720w&F
>>> Section start 2016/05/10 16:14:14.784
     dvi: {Build Driver List} 16:14:15.421
     dvi:      Searching for compatible ID(s):
     dvi:           wpdbusenum\fs
     dvi:           swd\generic
     dvi:      Created Driver Node:
     dvi:           HardwareID   - wpdbusenum\fs
     dvi:           InfName      - C:\WINDOWS\System32\DriverStore\FileRepository\wpdfs.inf_amc
     dvi:           DevDesc      - WPD FileSystem Volume Driver
     dvi:           Section      - Basic_Install
     dvi:           Rank         - 0x00ff2000
     dvi:           Signer Score - INBOX
     dvi:           DrvDate      - 06/21/2006
     dvi:           Version      - 10.0.10586.0
     dvi: {Build Driver List - exit(0x00000000)} 16:14:15.904
```

Σχήμα 47. Χρονική στιγμή πρώτης σύνδεσης στο αρχείο setupapi.dev.

Αντίστοιχα αποτελέσματα παίρνουμε και για σύνδεση συσκευών μέσω MTP και PTP.

ΣΥΝΟΨΗ-ΣΥΜΠΕΡΑΣΜΑΤΑ

Recycle Bin

Οι βασικές διαφορές μεταξύ των Windows 7, 8, 10 για τον κάδο ανακύκλωσης συνοψίζονται στον παρακάτω πίνακα:

Windows 7	Windows 8	Windows 10
Αρχικό offset 01	Αρχικό offset 01	Αρχικό offset 02
Offset διαδρομής δίσκου 24	Offset διαδρομής δίσκου 24	Offset διαδρομής δίσκου 28
Μέγεθος διαδρομής δίσκου εξαρτάται από το μέγεθος του διαγραμμένου αρχείου	Μέγεθος διαδρομής δίσκου είναι πάντα 520bytes	Μέγεθος διαδρομής δίσκου εξαρτάται από το μέγεθος του διαγραμμένου αρχείου

Volume Shadow Copies

Οι βασικές διαφορές μεταξύ των Windows 7, 8, 10 για τα Volume Shadow Copies συνοψίζονται στον παρακάτω πίνακα:

Windows 7	Windows 8	Windows 10
File-based backup/sector-by-sector backup	File-based backup/USN Journal	File-based backup/USN Journal
Previous Versions tab για πλοήγηση και επαναφορά μεμονωμένων αρχείων ή ολόκληρων φακέλων	Δεν υποστηρίζεται GUI/Το Previous Versions έχει αφαιρεθεί	Επανενσωμάτωση του Previous Versions χρησιμοποιώντας το File History αντί το VSC

Windows Jump Lists

Μέχρι στιγμής δεν υπάρχει κάποιο εργαλείο για να κάνει σωστό parsing των Windows 10 Jump Lists. Έχει ξεκινήσει μια προσπάθεια από τον Zimmerman [28]. Τα ήδη υπάρχοντα λειτουργούν σωστά για τις εκδόσεις 7 και 8 των Windows. Αυτό γιατί υπάρχει διαφορά στη δομή του DestList των Windows 10 με τη δομή αυτού στις εκδόσεις 7 και 8 [25], [26]. Ανάμεσα στα Windows 7 και 8 η μόνη διαφορά είναι η δημιουργία των Jump Lists του Windows Photo Viewer και των διαφορετικών internet browsers (Chrome, Firefox, Internet Explorer). Επιπρόσθετα, είναι προφανές ότι τα AppIDs είναι διαφορετικά αφού οι εφαρμογές εκτελούνται από διαφορετικό path [29].

Οι διαφορές αναφορικά με τη δομή του DestList header και των εγγραφών DestList συνοψίζονται στους παρακάτω δύο πίνακες [21], [27]:

ΔΟΜΗ DestList header		
Offset	**Windows 7, 8**	**Windows 10**
0-3	Version number: value 1	Version number: value 3
4-7	Πλήθος εγγραφών στη λίστα	Πλήθος εγγραφών στη λίστα
8-11	Πλήθος pinned εγγραφών	Πλήθος pinned εγγραφών
12-15	Floating point value. Πιθανόν κάποιος counter. Αρχική τιμή 0x00 0x00 0x80 0x3F (=1) (Για τον Windows Explorer 0x66 -x66 0x76 0x41 (δηλ.15,4)). Αυξάνει όσο προστίθενται νέες εγγραφές. Μειώνεται όταν διαγράφεται εγγραφή από τη Jump List.	Πιθανόν κάποιος counter. Σε σειρά από δοκιμές με Notepad είχε πάντα την τιμή 0x00. Στον File explorer αμέσως μετά από εγκατάσταση 0xAE 0xC7 0x96 0x42 (δηλ. 75,39)
16-23	Τελευταίος αριθμός Entry ID	Τελευταίος αριθμός Entry ID
24-31	Πλήθος προστιθέμενων ή αφαιρούμενων ενεργειών	Σε δοκιμή με Notepad ανοίχτηκαν 3 αρχεία και η τιμή ήταν 0x06. Μετά από 6 αρχεία η τιμή ήταν 0C (δηλ. 12). Φαίνεται να είναι 2πλάσιος των αρχείων στη λίστα. Αυξάνεται όταν εγγραφές γίνονται pinned, διαγράφονται ή ανοίγονται

ΔΟΜΗ εγγραφών DestList			
Offset	**Windows 7, 8**	**Offset**	**Windows 10**
0-7	Checksum/hash εγγραφής	**0-7**	Checksum/hash εγγραφής
8-23	New Volume ID	**8-23**	New Volume ID
24-39	Object ID	**24-39**	Object ID
40-55	Birth Volume ID	**40-55**	Birth Volume ID
56-71	Object ID	**56-71**	Object ID
72-87	Όνομα NetBIOS συμπληρωμένο με μηδενικά με μέγιστο μήκος 16bytes	**72-87**	Όνομα NetBIOS συμπληρωμένο με μηδενικά με μέγιστο μήκος 16bytes
88-95	Entry ID number	**88-91**	Entry ID number
96-99	Floating point counter για καταγραφή προσπέλασης αρχείου	**92-99**	Άγνωστο. Σε δοκιμή με Notepad είναι 0x00
100-107	MSFILETIME της τελευταίας καταγεγραμμένης προσπέλασης	**100-107**	MSFILETIME της τελευταίας καταγεγραμμένης προσπέλασης
108-111	Εγγραφή κατάστασης pin '0xFF 0xFF 0xFF 0xFF' = Unpinned, διαφορετικά ο counter αρχίζει από '0x00 0x00 0x00 0x00'.	**108-111**	Εγγραφή κατάστασης pin '0xFF 0xFF 0xFF 0xFF' = Unpinned, διαφορετικά ο counter αρχίζει από '0x00 0x00 0x00 0x00'.
112-113	Μήκος δεδομένων συμβολοσειράς Unicode	**112-115**	Σε δοκιμή με Notepad και file explorer είναι '0xFF 0xFF 0xFF 0xFF'
114-	Δεδομένα συμβολοσειράς	**116-119**	Σε δοκιμή με Notepad και file explorer είναι '0x01 0x00 0x00 0x00'. Αλλαγή σε 0x02 όταν ξαναανοίχθηκαν αρχεία.
		120-127	Σε δοκιμή με Notepad και file explorer είναι 0x00
		128-129	Μήκος δεδομένων συμβολοσειράς Unicode
		130-	Δεδομένα συμβολοσειράς ακολουθούμενα από ‘0x00 0x00 0x00 0x00’

Prefetch Files

Η βασική διαφορά είναι ότι τα prefetch files στα Windows 10 έιναι συμπιεσμένα σε MAM format με τον αλγόριθμο Xpress Huffman. Στα Windows 8 ήταν συμπιεσμένο σε clear text και απλώς χρειαζόμασταν κατάλληλο parser (Τα Windows 8.1 χρησιμοποιούν τον αλγόριθμο αυτό για τη συμπίεση των Superfetch files [30]). Από τη στιγμή που θα γίνει η αποσυμπίεση η επακόλουθη ανάλυση παραμένει η ίδια. Από τα Windows 8 μέχρι και τα 10 μπορούμε να έχουμε 8 τιμές τελευταίας εκτέλεσης αντί μιας στα Windows 7. Στον παρακάτω πίνακα συνοψίζονται οι διαφορές.

Χαρακτηριστικό	**Windows 7**	**Windows 8**	**Windows 10**
Format version identifier	0x17	0x1A	0x1E
Πλήθος εκτελέσεων	Offset 152, μήκος 4bytes	Offset 208, μήκος 4bytes	Offset 208, μήκος 4bytes
Χρόνος τελευταίας προσπέλασης	Μια προσπέλαση	Οκτώ προσπελάσεις	Οκτώ προσπελάσεις
Ενεργοποίηση prefetching	Ενεργοποιημένη. Αν ανιχνευθεί δίσκος SSD απενεργοποιείται	Αναλύονται τα χαρακτηριστικά απόδοσης του συστήματος και ενεργοποιείται/απενεργοποιείται αυτόματα με βάση τη βαθμολογία	Αναλύονται τα χαρακτηριστικά απόδοσης του συστήματος και ενεργοποιείται/απενεργοποιείται αυτόματα με βάση τη βαθμολογία
Πλήθος αρχείων .pf	128	1024	1024

USB activity

Με βάση τη μελέτη των Arshad et al. οι διαφορές και οι ομοιότητες συνοψίζονται στους παρακάτω πίνακες:

MSC συσκευές				
Ενέργεια	Key/subkey	Windows 7	Windows 8	Windows 10
Πρώτη σύνδεση (DeviceClasses hive)	10497b1b-ba51-44e5-8318-a65c837b6661	x	x	x
	53f56307-b6bf-11d0-94f2-00a0c91efb8b	x	x	x
	53f5630d-b6bf-11d0-94f2-00a0c91efb8b	x	x	x
	65a9a6cf-64 cd-480b-843e-32c86e1ba19f	x		
	6ac27878-a6fa-4155-baS5-f9Sf491d4f33		x	x
	7f108a28-9833-4b3b-b780-2c6b5fa5c062		x	x
	7fccc86c-228a-40ad-8a58-f590af7bfdce		x	x
	a5dcbf10-6530-11d2-901f-00c04fb951ed	x	x	x
	EEC5AD98-8080-425f-922A-DABF3DE3F69A	x		
	f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae	x	x	x
Πρώτη σύνδεση (USBSTOR key)	0003	**X**	**X**	**X**
	000A		**X**	**X**
	0064	**X**	**X**	**X**
	0065	**X**	**X**	**X**
Τελευταία σύνδεση (USBSTOR key)	0066		**X**	**X**
Τελευταία εξαγωγή (USBSTOR key)	0067		**X**	**X**
Πρώτη σύνδεση (USB key)	VID_[VendorID]&PID_[ProductID]\[SerialNo]\DeviceParameters \e5b3b5ac-9725-4f78-963f-03dfb1d828c7			**X**
τελευταία σύνδεση (USB key)	VID_[VendorID]&PID_[ProductID]\[SerialNo]\	**X**	**X**	**X**
Πρώτη σύνδεση (software hive)	Microsoft\Windows Portable Devices\Devices \SWD#WPDBUSENUM#_??_ USBSTOR#DISK&VEN_[VenderName] &PROD_[ProductName]&REV_PMAP#[SerialNo]#{53F56307-B6BF11D0-94F2-00A0C91EFB8B}	**X**	**X**	**X**

MTP & PTP συσκευές				
Ενέργεια	Key/subkey	Windows 7	Windows 8	Windows 10
Πρώτη σύνδεση (DeviceClasses hive)	10497b1b-ba51-44e5-8318-a65c837b6661 6ac27878-a6fa-4155-baS5-f9Sf491d4f33 6bdd1fc6-810f-11d0-bec7-08002be2092f a5dcbf10-6530-11d2-901f-00c04fb951ed EEC5AD98-8080-425f-922A-DABF3DE3F69A f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae	x x x x x x	x x x x x	x x x x x x
Πρώτη σύνδεση (USB Property key)	0003 0007 0008 0009 000A 0064 0065	**X** **X** **X**	**X** **X** **X** **X** **X** **X** **X**	**X** **X** **X** **X** **X** **X** **X**
Τελευταία σύνδεση (USB Property key)	0066		x	x
Τελευταία εξαγωγή (USB Property key)	0067		x	x
Πρώτη σύνδεση (USB key)	VID_[VendorID]&PID_[ProductID]\[SerialNo]\DeviceParameters \e5b3b5ac-9725-4f78-963f-03dfb1d828c7			x
τελευταία σύνδεση (USB key)	VID_[VendorID]&PID_[ProductID]\[SerialNo]\	x	x	x

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] http://4n6xplorer.com/forensics/once-upon-a-time-in-recycle-bin/

[2] https://lcdiblog.champlain.edu/2015/03/28/windows-10-recycle-bin-activity-introduction/

[3] https://df-stream.com/2016/04/fun-with-recycle-bin-i-files-windows-10/

[4] https://www.blackbagtech.com/blog/2017/01/19/examining-the-windows-10-recycle-bin/`

[5] Timothy R. Leschke, Cyber Dumpster-Diving: $Recycle.Bin Forensics for Windows 7 and Windows Vista

[6] http://4n6xplorer.com/forensics/the-return-of-recycle-bin/

[7] https://en.wikipedia.org/wiki/Shadow_Copy

[8] https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785914(v=ws.10)

[9] https://msdn.microsoft.com/en-us/library/windows/desktop/aa384649%28v=vs.85%29.aspx

[10] http://www.thewindowsclub.com/vss-volume-shadow-copy-service

[11] Zhu, Gladyshev & James, "Using Shellbag information to reconstruct user activities", Digital Investigation Vol. 6, pp.69-77, 2009

[12] Pullega, http://www.4n6k.com/2013/12/shellbags-forensics-addressing.html

[13] Tilbury, "Computer Forensics Artifacts: Windows 7 shellbags", https://digital-forensics.sans.org/blog/2011/07/05/shellbags

[14] McQuaid, "Forensic Analysis of Windows Shellbags", https://www.magnetforensics.com/computer-forensics/forensic-analysis-of-windows-shellbags/

[15] Carvey, "Shell Item Artifacts, Reloaded", Windows Incident Response, 2013 http://windowsir.blogspot.gr/2013/10/shell-item-artifacts-reloaded.html

[16] Ballenthin, "Windows Shellbag Forensics", http://www.williballenthin.com/forensics/shellbags/

[17] Lo, “Windows Shellbag Forensics in Depth”, SANS Institute, 2014 https://www.sans.org/reading-room/whitepapers/forensics/windows-shellbagsforensics-in-depth-34545

[18] Key, “Parsing Windows Shellbags using the Shellbags parser EnScript”, 2015, https://www.guidancesoftware.com/blog/digital-forensics/2015/03/30/parsing-windows-shellbags-using

[19] Carvey, “ShellBags Analysis”, Windows Incident Response, 2012 http://windowsir.blogspot.gr/2012/08/shellbag-analysis.html

[20] Pullega, https://twitter.com/4n6k/status/898707696643219460

[21] Singh, Singh, “Aforensic insight into Windows 10 Jump Lists”, 2016

[22] http://forensicswiki.org/wiki/List_of_Jump_List_IDs

[23] https://github.com/4n6k/Jump_List_AppIDs/blob/master/4n6k_AppID_Master_List.md

[24] http://www.hexacorn.com/blog/2013/04/30/jumplists-file-names-and-appid-calculator/

[25] Lyness, “Forensic analysis of Windows 7 Jump Lists”, 2012 https://articles.forensicfocus.com/2012/10/30/forensic-analysis-of-windows-7-jump-lists/

[26] Lallie, Bains, “An overview of the jumplist configuration file in Windows 7”, Journal of Digital Forensics, Security & Law, Vol. 7, No. 1, Article 2, 2012

[27] Zimmerman, https://binaryforay.blogspot.gr/2016/02/jump-lists-in-depth-understand-format.html

[28] Zimmerman, Lnk Explorer, https://github.com/EricZimmerman/LECmd

[29] Antonovich, “Jump List Forensics”, Champlain College, Leahy Center for Digital Investigation (LCDI), 2014

[30] Picasso, “A first look at Windows 10 prefetch files”, 2015, http://blog.digital-forensics.it/2015/06/a-first-look-at-windows-10-prefetch.html

[31] Michael, “DFSP#42 – Windows 10 prefetch”, 2016, http://digitalforensicsurvivalpodcast.com/2016/12/06/dfsp-042-windows-10-prefetch/

[32] Wade, “Decoding Prefetch files for forensic purposes: Part 1, 2010, https://www.forensicmag.com/article/2010/12/decoding-prefetch-files-forensic-purposes-part-1

[33] Shashidhar, Novak, “Digital Forensic Analysis on Prefetch Files”, International Journal of Information Security Science, Vol. 4, No. 2, pp. 39-49, 2015

[34] Thomas (Tribal Chicken), “Recovering Bitlocker Keys on Windows 8.1 & 10”, 2016

[35] Microsoft online, “Windows Bitlocker Drive Encryption FAQ”, 2012 https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc766200(v=ws.10)

[36] https://www.forensicswiki.org/wiki/BitLocker_Disk_Encryption

[37] Champlain College, Leahy Center for Digital Investigation (LCDI), “Windows 10 Forensics – Part 1”, 2015

[38] https://www.forensicswiki.org/wiki/USB_History_Viewing

[39] Arshad, Iqbal, Abbas, “USB Storage Device Forensics for Windows 10”, Journal of Forensic Sciences, Vol. 63, Issue 3, 2017