Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή

Ανάλυση και Σύγκριση Κακόβουλου Λογισμικού σε Λειτουργικά Συστήματα Windows 10, Windows 8.1 και Windows 7 για Εξαγωγή Forensic Artifacts

Δημήτριος Κοντογεώργης

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Μάιος 2016

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή

Ανάλυση και Σύγκριση Κακόβουλου Λογισμικού σε Λειτουργικά Συστήματα Windows 10, Windows 8.1 και Windows 7 για Εξαγωγή Forensic Artifacts

Δημήτριος Κοντογεώργης

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Πληροφοριακά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Μάιος 2016

ΛΕΥΚΗ ΣΕΛΙΔΑ

Περίληψη

Οι κυβερνοεγκληματίες σήμερα είναι σε θέση να σχεδιάσουν και να εκτελέσουν μαζικές ή περισσότερο στοχευμένες επιθέσεις με τη χρήση κακόβουλου λογισμικού, για να εισβάλουν και να μολύνουν το πληροφοριακό σύστημα του θύματος και να το χρησιμοποιούν μετά για δικό τους όφελος χωρίς να το γνωρίζει το θύμα. Πολλές φορές οι Εξεταστές Τεκμηρίων έρχονται αντιμέτωποι με υποθέσεις που φαινομενικά δείχνουν να ευθύνεται το άτομο στο οποίο ανήκει ο υπολογιστής αλλά μετά από αρκετές ώρες έρευνας να ευθύνεται το κακόβουλο λογισμικό που ενεργούσε για αυτόν. Στόχος την παρούσας μεταπτυχιακής διατριβής είναι η μελέτη και η καταγραφή της συμπεριφοράς αντιπροσωπευτικών δειγμάτων κακόβουλου λογισμικού από τις οικογένειες των Trojan, των Worm και των Bot στις τρείς τελευταίες και αρκετά διαδεδομένες εκδόσεις Windows (Windows 7, 8.1, 10) και να εξαχθούν κάποιοι κανόνες ώστε να διευκολύνουν τη δουλεία ενός Forensic Examiner και να μειώσουν αυτό το χάσιμο χρόνου. Συγκεκριμένα η έκδοση των Windows 10, λόγο της πρόσφατης διάθεσης της στην αγορά δεν υπάρχει σχετική επαρκής έρευνα για την ανεύρεση ψηφιακών τεκμηρίων, καθώς και για την συμπεριφορά κακόβουλου λογισμικού. Η μεθοδολογία που ακολουθήθηκε στηρίχτηκε στην δυναμική ανάλυση κακόβουλου λογισμικού με την χρήση του Cuckoo Sandboxing. Τα εξήντα δείγματα (είκοσι από κάθε κατηγορία), δοκιμάστηκαν στα τρία λειτουργικά συστήματα, ώστε να καταγραφεί η συμπεριφορά τους. Επιπλέον, καταγράφηκαν για κάθε λειτουργικό και για κάθε κατηγορία κακόβουλου λογισμικού, οι σημαντικότερες θέσεις ανεύρεσης ψηφιακών τεκμηρίων. Από την ανάλυση προέκυψε ότι το είδος του λειτουργικού συστήματος αλλά και η κατηγορία του κακόβουλου λογισμικού, καθορίζουν σημαντικά τις θέσεις ανεύρεσης ψηφιακών τεκμηρίων. Συνεπώς η μοντελοποίηση των παραπάνω ευρημάτων και η ενσωμάτωση τους σε μία εφαρμογή θα μπορούσε να αποτελέσει ένα χρήσιμο εργαλείο για τους ερευνητές ψηφιακών τεκμηρίων, καθώς με μία σύντομη ανάλυση στις σημαντικότερες θέσεις ψηφιακών τεκμηρίων, θα μπορούσαν να βγάλουν ένα άμεσο αποτέλεσμα για την ύπαρξη και το είδος του κακόβουλου λογισμικού.

Summary

Cybercriminals today are able to design and perform mass or more targeted attacks using malicious software to invade and infect the victim's computer system and use it after for their own benefit without the knowledge of the victim. Very often Digitals Examiners get involved with cases which seemingly pointing the responsibility to the person which the computer belongs, but after several hours of research proving to be liable malware which acting for him. The aim of this master thesis is to study and record the behavior of representative malware samples from the families of the Trojan, the Worm and Bot in the last three and fairly widespread Windows versions (Windows 7, 8.1, 10) and extract some rules to facilitate the work of a Forensic Examiner and reduce this waste of time. Specifically, the version of Windows 10, because of the recent disposal of the market there is not enough relevant research finding digital evidence as well as for the malware conduct. The methodology used was based on the dynamic analysis of malware using the Cuckoo Sandboxing. Sixty samples (twenty of each category), tested on three operating systems in order to record their behavior. In addition, the most important positions of digital forensics findings were recorded for each operating system and category. The analysis showed that the type of operating system and category of malware, significantly determine the positions of finding digital forensics. Therefore, the modeling of these findings and their integration in one application could be a useful tool for digital forensics researchers, as with a brief analysis of the most important positions of digital evidence, they could have a direct result on the existence and category of malware.

Ευχαριστίες

Σε σχέση με τη Διπλωματική Εργασία, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα Καθηγητή μου Δρ. Σταύρο Σιαηλή, προς τον οποίο αισθάνομαι ευγνώμων, τόσο για την ουσιαστική βοήθεια και συμμετοχή του στην υλοποίηση της Διπλωματικής Εργασίας, όσο και για το γεγονός ότι δε μού μετέδωσε απλώς τεχνικές και επιστημονικές γνώσεις, αλλά μού εμφύσησε έναν νέο τρόπο σκέψης και προσέγγισης των πραγμάτων.

Επιπλέον θα ήθελα να ευχαριστήσω την σύζυγο μου Ελίνα Γκάνα και τον γιό μου Κωνσταντίνο, από τους οποίους δανείστηκα πολύτιμο χρόνο και που πάντα με στηρίζουν.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Καθώς το κακόβουλο λογισμικό εξελίσσεται και γίνεται όλο και πιο πολυσύνθετο, οι κακόβουλοι εισβολείς έχουν τη δυνατότητα να προσαρμόζουν τη συμπεριφορά τους ανάλογα με το σύστημα που επιθυμούν να μολύνουν και το περιβάλλον. Το κακόβουλο λογισμικό μπορεί να αποκαλυφθεί μόνο μετά από την αναγνώριση συγκεκριμένων παραγόντων του συστήματος και τον συνδυασμό πολλών παραμέτρων και συνθηκών. Για παράδειγμα, ένα συγκεκριμένο κακόβουλο λογισμικό θα μπορούσε να αποκαλύψει τη συμπεριφορά του μόνο όταν εγκατασταθεί σε μια πλατφόρμα των Windows 7 ή όταν ένα συγκεκριμένο λογισμικό είναι εγκατεστημένο στον υπολογιστή του θύματος (όπως PDF Reader) και να παραμείνει εντελώς αδρανές σε οποιαδήποτε άλλη κατάσταση. Ομοίως, μπορεί να αποκαλύψει ένα μέρος της συμπεριφοράς του, ενώ τμήματα της λειτουργικότητας της να παραμείνουν κρυφά μέχρι να πληρούνται ορισμένες προϋποθέσεις που θα προκαλέσουν πρόσθετη δραστηριότητα. Έχουν γίνει προσπάθειες για να αποκαλυφθούν δείγματα που ενεργοποιούνται με την συμπεριφορά (Moser, Kruegel & Kirda, 2007a; Brumley *et al.*, 2008), αλλά έχει επίσης αποδειχθεί ότι είναι εφικτό να εξαπατηθούν τέτοιοι αναλυτές (Sharif *et al.*, 2008). Από τα παραπάνω τεκμηριώνεται η δύσκολη αποστολή την οποία έχει να φέρει εις πέρας ο αναλυτής ψηφιακών τεκμηρίων. Κατά την διάρκεια εξέτασης μίας υπόθεσης, ελλοχεύει πάντα ο κίνδυνος τα ψηφιακά τεκμήρια να είναι αποτέλεσμα δράσης κακόβουλου λογισμικού. Θα μπορούσε δηλαδή να κατηγορηθεί ο κάτοχος ενός υπολογιστικού συστήματος άδικα καθώς ένα κακόβουλου λογισμικό θα μπορούσε να τοποθετήσει στοιχεία ή να δώσει πρόσβαση σε κάποιο τρίτο να το κάνει αυτό. Συνεπώς σε κάθε περίπτωση πριν από την καταγραφή των αποδεικτικών στοιχείων θα πρέπει να γίνεται ενδελεχή έρευνα για την ύπαρξη ή για την προηγούμενη δράση κακόβουλου λογισμικού ώστε να διασφαλίζεται η δίκαια αντιμετώπιση της υπόθεσης.

Το προσωπικό το οποίο εμπλέκεται στην ανάλυση κακόβουλου λογισμικού είτε σαν αναλυτής της λειτουργικότητας του δείγματος, είτε με τον εντοπισμό των αλλαγών που επιφέρει στο σύστημα αντιμετωπίζουν εξίσου μία πρόκληση, θα πρέπει να αναλύσουν πάνω από ένα εκατομμύριο δείγματα την ημέρα (Cnn, 2016). Καταδεικνύεται ότι αυτό είναι πιθανότατα ένα ακατόρθωτο εγχείρημα. Η εργασία αυτή εκπονήθηκε ορμώμενη από την επιθυμία να δημιουργηθούν κάποιοι ενδείκτες (indicators) που θα καθοδηγήσουν τους ερευνητές με σκοπό η διαδικασία ανάλυσης να γίνει ταχύτερη και πιο αποτελεσματική. Η εργασία χωρίζεται σε 5 κεφάλαια. Στο κεφάλαιο 2 γίνεται η ανασκόπηση της βιβλιογραφίας εστιάζοντας σε μια σύντομη ιστορία του λειτουργικού Windows καθώς και στις θέσεις ψηφιακών τεκμηρίων στο Μητρώο των Windows και στο σύστημα αρχείων. Η ραγδαία ανάπτυξη και διάδοση του κακόβουλου λογισμικού έχει επηρεάσει σημαντικά τον τρόπο με τον οποίο οι επαγγελματίες της ασφάλειας και οι εμπειρογνώμονες κακόβουλου λογισμικού έπρεπε να ανταποκριθούν σε αυτές τις απειλές. Κατά συνέπεια, καταγράφονται κάποιες από τις σημαντικότερες κατηγορίες και παρουσιάζεται η στατική και δυναμική ανάλυση του κακόβουλου λογισμικού ως μέρος της μεθοδολογίας που χρησιμοποιείται. Στην συνέχεια αναλύεται το λογισμικό Cuckoo Sandbox και απαριθμούνται οι διαδικασίες νομικής κατοχύρωσης ψηφιακών τεκμηρίων.

Ο σχεδιασμός της πειραματικής διαδικασίας παρουσιάζεται στο κεφάλαιο 3. Γίνεται μία εκτενής αναφορά στην συνεισφορά της έρευνας, οι απαιτήσεις σε υλικό και λογισμικό, ο τρόπος με τον οποίο έγινε η συλλογή των δειγμάτων κακόβουλου λογισμικού που αναλύθηκε καθώς και η περιγραφή της πειραματικής διαδικασίας.

Στο κεφάλαιο 4 αναφέρονται στο αποτέλεσμα της πειραματικής διαδικασίας και παρουσιάζονται τα ψηφιακά τεκμήρια που προέκυψαν με βάση την λειτουργικότητα, το λειτουργικό, το είδος του δείγματος και ανά θέση. Στην συνέχεια στο κεφάλαιο 5 γίνεται αναλυτικός σχολιασμός των ευρημάτων και προτείνονται πιθανές μελλοντικές προεκτάσεις της διεξαχθείσας έρευνας.

Κεφάλαιο 2

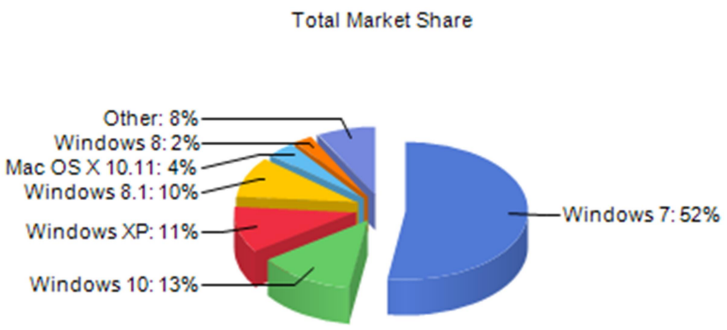
Βιβλιογραφική ανασκόπηση

2.1 Ανασκόπηση Windows

Στόχος του κάθε οργανισμού είναι να οικοδομήσει κατάλληλες άμυνες προκειμένου να προστατεύσει την επιχείρησή του και να αποτρέψει, όσο το δυνατόν περισσότερο, κάθε εισβολέα που καραδοκεί. Επιπλέον οι μεμονωμένοι χρήστες προσπαθούν να διατηρούν τον υπολογιστή τους προστατευμένο από κακόβουλες επιθέσεις, ώστε να προστατέψουν τα δεδομένα τους, την ιδιωτικότητα τους αλλά και να διασφαλίσουν ότι δεν θα ενοχοποιηθούν από στοιχεία που θα ανευρεθούν εξαιτίας μη εξουσιοδοτημένης πρόσβασης ή ενός κακόβουλου προγράμματος. Αναμφισβήτητα όμως, αν οι εγκληματίες αποφασίζουν να επιτεθούν αργά ή γρήγορα θα βρουν τρόπο για να το επιτύχουν. Οι κυβερνοεγκληματίες σήμερα είναι σε θέση να ενορχηστρώσουν και να εκτελέσουν μαζικές ή περισσότερες στοχευμένες επιθέσεις με τη χρήση κακόβουλου λογισμικού ως μέσο για να εισβάλουν και να μολύνουν τα υπολογιστικά συστήματα του θύματος. Η ανίχνευση και η ανάλυση αυτών των επιθέσεων ενδέχεται να μην είναι πάντα εφικτή και μπορεί να μεταβληθεί σε μια δύσκολη και αποθαρρυντική διαδικασία.

2.1.1 Το λειτουργικό σύστημα Windows

Οι διάφορες εκδόσεις των λειτουργικών συστημάτων Windows αντιπροσωπεύουν την πλειοψηφία του συνόλου των εγκατεστημένων λειτουργικών συστημάτων σε όλο κόσμο. Έχει αποδειχθεί σε έρευνα από την Netmarketshare που διεξήχθη τον Φεβρουάριο του 2016 ότι οι διάφορες εκδόσεις των Windows (10, 8.1, 8, 7, Vista, XP, 2000) αθροιστικά αντιπροσωπεύουν το 90,35% των λειτουργικών συστημάτων που καταγράφονται στα αρχεία καταγραφής χρήσης του διαδικτύου (NetMarketShare, 2016)



Εικόνα 1:Το μερίδιο αγοράς για τα διάφορα λειτουργικά συστήματα (NetMarketShare, 2016)

Ως εκ τούτου, λόγω της καθολικής εξάπλωσης της οικογένειας των Windows, κάθε νέα έκδοση του λειτουργικού συστήματος πρέπει να αναλύεται σε βάθος, εξαιτίας των πιθανών οφελών για τους ειδικούς εγκληματολογίας. Οι πληροφορίες αυτές μπορεί να περιλαμβάνουν περισσότερες λεπτομέρειες σχετικά με τη δικτυακή δραστηριότητα του χρήστη, προσδιορίζοντας διαρροές δεδομένων ή δείκτες μόλυνσης των δεδομένων. Η Microsoft έχει ανακοινώσει τις ημερομηνίες μέχρι τις οποίες θα υποστηρίξει τα λειτουργικά της συστήματα (Microsoft, 2016b). Για τα Windows XP, η υποστήριξη τους έχει τελειώσει από τον Απρίλιο του 2014, με αποτέλεσμα εξαιτίας της ανασφάλειας που παρέχει, λόγω των αδυναμιών που πλέον δεν επιδιορθώνονται, ο αριθμός των ενεργών εγκαταστάσεων να μειώνεται συνεχώς. Επιπλέον για τα Windows Vista η υποστήριξη τους τελειώνει σε δέκα περίπου μήνες. Επιπλέον το μερίδιο στις ενεργές εγκαταστάσεις είναι πολύ μικρό. Για τους λόγους αυτούς οι εκδόσεις των Windows που επιλέχτηκαν για να αναλυθούν είναι τα Windows 7, 8.1 και 10.

Από όλες τις σχετικές πηγές ψηφιακών τεκμηρίων που βρίσκονται σε ένα τυπικό λειτουργικό σύστημα των Windows, το μητρώο των Windows είναι ιδιαίτερου ενδιαφέροντος για τον ερευνητή. Το μητρώο είναι δομημένο σαν μια εκτεταμένη ιεραρχική βάση δεδομένων για χρήση από βοηθητικά προγράμματα του συστήματος, καθώς και εφαρμογές τρίτων. Σκοπός της είναι να παρέχει μια κοινή πλατφόρμα για την αποθήκευση των ρυθμίσεων διαμόρφωσης και τις προτιμήσεις. Ως εκ τούτου, η χρησιμότητα του μητρώου εξαρτάται από τις πληροφορίες που αποθηκεύονται σε αυτό. Το μητρώο των Windows μπορεί να χρησιμοποιηθεί για να αποκαλύψει πολλές πληροφορίες στην πορεία μιας έρευνας. Υπάρχουν πολλά σημεία που είναι γνωστά

στους ερευνητές για τα δεδομένα που παρέχουν. Οι διευθύνσεις URL θα αποκαλύψουν ποιες συγκεκριμένες διευθύνσεις έχει πληκτρολογήσει ένα άτομο στη γραμμή διευθύνσεων του Internet Explorer. Οι λίστες με τα πιο πρόσφατα χρησιμοποιημένα στοιχεία είναι πηγές που δείχνουν ποια αρχεία έχουν ανοιχτεί ή ποιες εφαρμογές έχουν τρέξει. Τα ShellBags είναι ένα σύνολο από κλειδιά που αντιπροσωπεύουν τις ρυθμίσεις του χρήστη για τον File Explorer. Μπορούν να αποδείξουν ότι ένας χρήστης έχει ανοίξει συγκεκριμένους καταλόγους σχετικά με το σύστημα ή με τις δικτυακές θέσεις. Το κακόβουλο λογισμικό το οποίο προσπαθεί να επιτύχει μόνιμη παρουσία μετά από μια επανεκκίνηση του συστήματος, μπορεί να έχει παρουσία μέσα στο μητρώο. Αυτό συμβαίνει επειδή υπάρχουν πολλά κλειδιά που σχετίζονται με εφαρμογές ή υπηρεσίες που ξεκινούν κατά την εκκίνηση του συστήματος. Πληροφορίες απεγκατάστασης εφαρμογών μπορούν να δώσουν σε έναν ερευνητή μια εικόνα για τα προγράμματα που έχουν εγκατασταθεί στο σύστημα. Ιδιαίτερα χρήσιμη είναι η ικανότητα να εντοπίζονται και να αναγνωρίζονται αφαιρούμενοι δίσκοι USB που έχουν χρησιμοποιηθεί από τον μοναδικό σειριακό αριθμό τους. Αυτά είναι μόνο μερικά παραδείγματα των χρήσιμων στοιχείων που μπορούν να χρησιμοποιηθούν κατά τη διάρκεια μιας έρευνας (Stormo, 2013).

2.1.2 Θέσεις ψηφιακών τεκμηρίων στο Μητρώο των Windows και στο σύστημα αρχείων

Η καταγραφή των σημείων που αναμένεται ότι βρίσκονται εκτελέσιμα αρχεία, είναι σημαντική για την κατανόηση των αρχείων που είναι ύποπτα. Τα εκτελέσιμα συνήθως χωρίζονται σε δύο μεγάλες κατηγορίες: τα αρχεία του συστήματος και τα αρχεία της εφαρμογής. Τα περισσότερα αρχεία συστήματος βρίσκονται σε υποκαταλόγους κάτω από το κατάλογο C:\Windows. Τα αρχεία εφαρμογών κατά κύριο λόγο εντοπίζονται σε συγκεκριμένους υποκαταλόγους της εφαρμογής στο C:\Program Files και C:\Program Files (x86). Επιπλέον, οι φάκελοι προσωρινών αρχείων των Windows αποθηκεύουν προσωρινά αρχεία που χρησιμοποιούνται κατά την εγκατάσταση ή την αναβάθμιση της εφαρμογής. Αυτοί οι φάκελοι είναι συχνά γκρίζες ζώνες, δεδομένου ότι τόσο νόμιμες εφαρμογές, όσο και κακόβουλα προγράμματα κάνουν χρήση της προσωρινής τοποθεσίες αρχείων (Malicious-streams, 2014) .

Κάθε εκτελέσιμο αρχείο που βρίσκεται έξω από μια γνωστή θέση όπου αναμένεται να βρίσκεται, μπορεί να θεωρηθεί ύποπτο, πράγμα το οποίο αφήνει μια πολύ μεγάλη

περιοχή για έρευνα λαμβάνοντας υπόψη τον αριθμό των φακέλων σε μία τυπική εγκατάσταση των Windows. Ωστόσο, ο προσδιορισμός των θέσεων που συχνά ευνοείται για εγκατάσταση κακόβουλου λογισμικού θα βοηθήσει να περιοριστεί το πεδίο της αναζήτησης. Κακόβουλα εκτελέσιμα εγκαθίστανται σε εγγράψιμες περιοχές στο σύστημα αρχείων ενός θύματος, με πιο συνηθισμένη θέση τον κατάλογο του χρήστη. Μέσα σε αυτόν τον φάκελο οι θέσεις Application Data, Local, Roaming και Temp συνήθως είναι οι πιο δημοφιλείς, ενώ λιγότερο συχνές θέσεις μπορεί να περιλαμβάνουν τον φάκελο προσωρινής αποθήκευσης (cache) του Internet, τις θέσεις λήψης αρχείων, κοινόχρηστες τοποθεσίες των χρηστών, καθώς και το επίπεδο της ρίζας των δίσκων του συστήματος ή καταλόγους αρχείων προγράμματος.

Ένας σημαντικός ερευνητής που δραστηριοποιείται στον τομέα της ανάλυσης του μητρώου των Windows είναι ο Harlan Carvey. Το βιβλίο του Carvey “Windows Registry Forensics” (Carvey, 2011) έχει αποτελέσει χρήσιμο βοήθημα στην κατανόηση της διαδικασίας της ανάλυσης του μητρώου και την ουσιαστική ερμηνεία των αντικειμένων. Πραγματεύεται την ιατροδικαστική αξία του μητρώου των Windows και δίνει μια γενική εικόνα για το πώς γίνεται προσπέλαση και περιήγηση στα δεδομένα που αποθηκεύονται μέσα σε αυτό.

Το μητρώο των Windows αποτελεί μία ιεραρχική βάση δεδομένων με πολλά δυαδικά αρχεία γνωστά ως κυψέλες (hives). Μία από αυτές τις κυψέλες, η κυψέλη του υλικού, υπάρχει μόνο σαν πτητική μνήμη στο σύστημα και αναδημιουργείται κάθε φορά που ξεκινάει το σύστημα. Ο κατάλογος “Windows\System32\config” περιέχει τα αρχεία SAM, Security, Software, System και Default. Για κάθε χρήστη στο σύστημα, υπάρχει μια πρόσθετη κυψέλη "ntuser.dat" που βρίσκεται στο %UserProfile%. Για παράδειγμα, "C:\ Users\Mary\Ntuser.dat".

Οι βασικοί φάκελοι σε μία τυπική εγκατάσταση των Windows είναι πέντε (EC-Council, 2010). Αυτοί οι πέντε φάκελοι περιέχουν υποφακέλους και αρχεία, όπου σχεδόν στο κάθε αρχείο αντιστοιχεί και μια τιμή.

- HKEY_CLASSES_ROOTQ: σε αυτόν τον φάκελο υπάρχουν όλες οι ρυθμίσεις που αφορούν εφαρμογές που υποστηρίζουν πληροφορίες καταγραφής κλάσεων COM όπως ProgIDs, CLSIDs και IIDs (Microsoft, no date).
- HKEY_CURRENT_USER: σε αυτόν τον φάκελο υπάρχουν όλες οι ρυθμίσεις του χρήστη στις διάφορες εφαρμογές που χρησιμοποιεί. Περιέχει τις πληροφορίες ρύθμισης των παραμέτρων για το χρήστη που είναι συνδεδεμένος τη συγκεκριμένη στιγμή. Οι φάκελοι του χρήστη, τα χρώματα της οθόνης κι οι

ρυθμίσεις του πίνακα ελέγχου αποθηκεύονται εδώ. Αυτές οι πληροφορίες σχετίζονται με το προφίλ του χρήστη (Microsoft, 2016b).

- HKEY_LOCAL_MACHINE: σε αυτόν τον φάκελο υπάρχουν πληροφορίες που χρησιμοποιούν οι διάφορες εφαρμογές που αφορούν τον υπολογιστή για όλους τους χρήστες.
- HKEY_USERS: εδώ υπάρχουν οι ρυθμίσεις των χρηστών κι οι ρυθμίσεις του πίνακα ελέγχου. Το HKEY_CURRENT_USER είναι υποφάκελος του HKEY_USERS. HKEY_USERS
- HKEY_CURRENT_CONFIG: Σε αυτό το κλειδί αποθηκεύονται όλες οι ρυθμίσεις που αφορούν το hardware του υπολογιστή, σύμφωνα πάντα με το HKEY_CURRENT_USER.

Κατά την ανάλυση κακόβουλου λογισμικού, κάποιες θέσεις στο μητρώο αλλά και στο σύστημα αρχείων των Windows, έχουν καταγραφεί σαν σημαντικές για την ανεύρεση πιθανών στοιχείων μόλυνσης. Συνοπτικά θα μπορούσαμε να καταγράψουμε τις εξής (Symantec, 2009a; Norton, 2010; Carvey, 2011; Cert-Eu, 2012; RSA, 2013; Bayer *et al.*, 2014; Harrell, 2014; Horsman, Laing & Vickers, 2014; Malicious-streams, 2014; Fnal, 2016):

- Hkey_Local_Machine\System\Controlset001\Control\Nls\Customlocale\En-Us Checkpoint (Checkpoint, 2013). Οι καταχωρήσεις σε αυτό το δευτερεύον κλειδί απαριθμούν τα locales που είναι εγκατεστημένα στο σύστημα.
- Hkey_Local_Machine\System\Controlset001\Control\Session Trendmicro (Trendmicro, 2015). Το δευτερεύον κλειδί Session Manager περιέχει τα δεδομένα που χρησιμοποιούνται από πολλά διαφορετικά συστατικά του Διαχειριστή Συνόδου.
- Hkey_Local_Machine\System\Controlset001\Control\Nls Checkpoint (Checkpoint, 2013). Αυτό το δευτερεύον κλειδί περιέχει πληροφορίες σχετικά με τις υποστηριζόμενες γλώσσες και τις σελίδες κώδικα.
- Hkey_Local_Machine\Software\Wow6432node\Microsoft\Windows\Current version (CIRCL, 2009). Ένα από τα σημεία στα οποία γίνεται αυτόματα φόρτωση λογισμικού κατά την εκκίνηση.
- Hkey_Local_Machine\Software\Microsoft\Rpc (Enigmasoftware, 2014). Περιέχει το όριο για τις εισερχόμενες κλήσεις Microsoft Remote Procedure Call (RPC).

- Hkey_Current_User\Software\Microsoft\Windowsnt\Currentversion\Windows (Malwareremovalguides, 2013). Καταγράφονται οι εφαρμογές που ξεκινούν αυτόματα για τον συγκεκριμένο χρήστη.
- Hkey_Current_User\Software\Microsoft\Windows\Currentversion\ (Symantec, 2014). Καταγράφονται οι εφαρμογές που ξεκινούν αυτόματα για τον συγκεκριμένο χρήστη.
- Hkey_Local_Machine\System\Controlset001\Control (FireEye, 2012). Περιέχει πληροφορίες για τον έλεγχο εκκίνησης του συστήματος κι ορισμένες πτυχές της διαμόρφωσης των συσκευών.
- Hkey_Local_Machine\Software\Microsoft\ (Kleiman & Hunter, 2006). Ελέγχουν τις προεπιλεγμένες ρυθμίσεις εκκίνησης, τις ρυθμίσεις δικαιωμάτων πρόσβασης και τις δυνατότητες ασφαλείας δικαιώματα σε επίπεδο κλήσεων για εφαρμογές COM-based.
- Hkey_Current_User\Software\Microsoft\Windows\Currentversion\Explorer (Symantec, 2009b). Περιέχουν τα ονόματα των πρόσφατα χρησιμοποιημένων εκτελέσιμων αρχείων και τα μονοπάτια του φακέλου τους.
- Documents And Settings\[User Name]\Start Menu\Programs\Startup (Trendmicro, 2015b). Περιέχει μια λίστα με τα προγράμματα που εκτελούνται αυτόματα κάθε φορά που ξεκινά ο υπολογιστής.
- Documents and Settings\[User Name]\Local Settings\Temp (Malwarebytes, 2014). Αποθηκευμένα προσωρινά αρχεία του χρήστη.
- %Systemdrive%\Users\victim_user\AppData\ (Bitdefender, 2010). Περιέχει δεδομένα εγκαταστημένων εφαρμογών.
- %Systemdrive%\Windows\System32. Ο πιο γνωστός φάκελος συστήματος σε μία εγκατάσταση των Windows.
- %Systemdrive%\Windows\INF\. Ο φάκελος που αποθηκεύονται οι οδηγοί για τις συσκευές του υπολογιστή που τρέχει λειτουργικό Windows.
- %Systemdrive%\Windows\Globalization\Sorting\sortdefault.nls (Checkpoint, 2013).
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ (Symantec, 2009a). Περιέχει πληροφορίες σχετικά με μια εφαρμογή που χρειάζεται για να υποστηρίξει τη λειτουργία COM. Οι πληροφορίες αυτές περιλαμβάνουν θέματα όπως υποστηριζόμενες μορφές δεδομένων, πληροφορίες συμβατότητας, DCOM και ελέγχους.

- HKEY_CURRENT_USER\Software\Microsoft (Trendmicro, 2015a). Αποθηκεύει πληροφορίες διαμόρφωσης για τα προγράμματα της Microsoft που εκτελούνται στον υπολογιστή. Κάθε δευτερεύον κλειδί αντιπροσωπεύει ένα διαφορετικό πρόγραμμα. Οι εγγραφές στα δευτερεύοντα κλειδιά του προγράμματος είναι συγκεκριμένες για τον τρέχοντα χρήστη.

2.2 Ανασκόπηση κακόβουλου λογισμικού

Ο όρος Malware είναι ένας γενικός όρος για να περιγράψει όλα τα είδη του ανεπιθύμητου λογισμικού (π.χ. viruses, worms και Trojan horses) και αποτελεί συντόμευση του malicious software, και αναφέρετε σε λογισμικό το οποίο έχει σχεδιαστεί για να καταστρέψει ή να προκαλέσει ζημιά, χωρίς την άδεια του ιδιοκτήτη (M.G. Schultz *et al.*, 2001). Οι ερευνητές προέβλεψαν ότι οι επιχειρήσεις σε όλο τον κόσμο θα δαπανήσουν περίπου $ 500 δισεκατομμύρια δολάρια το 2014 σε διορθώσεις και την ανάκτηση από παραβιάσεις δεδομένων και malware. Αντίστοιχα, οι καταναλωτές σε όλο τον κόσμο ξεπέρασαν τα $25 δισεκατομμύρια, ως αποτέλεσμα των εν λόγω απειλών. Επίσης, σπαταλήθηκαν 1,2 δισεκατομμύρια αντιμετωπίζοντας τις συνέπειες, σύμφωνα με τη «The Link Between Pirated Software and Cybersecurity Breaches», στην οποία ερωτήθηκαν 951 καταναλωτές, καθώς και 450 CIOs και IT διαχειριστές. Η μελέτη κυκλοφόρησε από τη Microsoft ως μέρος της παγκόσμιας «Play It Safe» εκστρατεία της (*Play It Safe*, no date). Τέτοια είδη λογισμικού αποτελούν μία σοβαρή απειλή ασφαλείας για τους χρήστες. Το πρόβλημα με την δημιουργία κακόβουλου κώδικα είναι ότι αποτελεί μία κερδοφόρα και ακμάζουσα αγορά (Symantec, 2015). Οι δημιουργοί του λογισμικού αυτού μπορούν να πουλήσουν τον κώδικα τους σε απατεώνες, οι οποίοι μπορούν να το χρησιμοποιήσουν για να μολύνουν μεγάλο αριθμό μηχανημάτων, έτσι ώστε να αποτελέσουν μέρος μίας πλατφόρμας επίθεσης άρνησης υπηρεσίας ή αποστολής spam. Μία άλλη παράμετρος του προβλήματος είναι ότι ακόμη και άνθρωποι οι οποίοι δεν έχουν κάποιο συγκεκριμένο ενδιαφέρον για τους υπολογιστές, ανησυχούν για λογισμικού τύπου Ransomware. Αυτό συμβαίνει καθώς περιστατικά ασφαλείας επηρεάζουν χιλιάδες χρήστες και συνήθως αποτελούν ειδήσεις υψηλής ακροαματικότητας.

Προσπάθειες για να διερευνηθεί η ενδεχόμενη πρόληψη κρουσμάτων κακόβουλου λογισμικού έχουν παρακινήσει προηγούμενες μελέτες σχετικά με τους κακόβουλους κώδικες (Balthrop *et al.*, 2004; Costa *et al.*, 2005; Zou, Towsley & Gong, 2007). Η

τρέχουσα έρευνα στην ασφάλεια του κυβερνοχώρου επικεντρώνεται στην χαρακτηρισμό και μοντελοποίηση συγκεκριμένων επιθέσεων, με στόχο την κατανόηση των μηχανισμών της διείσδυσης, της ανίχνευσης και της αντιμετώπισης. Καθώς οι κυβερνοαπειλές πολλαπλασιάζονται τόσο σε όγκο όσο και στην πολυπλοκότητά τους, έχει αυξηθεί σημαντικά το ενδιαφέρον στο μεταδοτικό κακόβουλο λογισμικό (Liu *et al.*, 2016). Θεωρητικά, ένα από τα ενδιαφέροντα θέματα είναι η δημιουργία αξιόπιστων μαθηματικών μοντέλων που μπορούν να εφαρμοστούν στην αποτελεσματική περιγραφή και πρόβλεψη της εξέλιξης του κακόβουλου λογισμικού ηλεκτρονικών υπολογιστών. Δεδομένου ότι η εξάπλωση του κακόβουλου κώδικα είναι παρόμοια με τις βιολογικές επιδημίες (Vespignani, 2005), ορισμένα επιδημιολογικά μοντέλα έχουν υιοθετηθεί για να μελετηθεί η συμπεριφορά του κακόβουλου λογισμικού (Cheng *et al.*, 2011; Li *et al.*, 2014; Mishra & Pandey, 2014; Misra, Verma & Sharma, 2014; Shukla *et al.*, 2014). Επιπλέον, νέες στρατηγικές και μεθοδολογίες είναι αναγκαίες για την αποτροπή εισβολών και την αντιμετώπιση των επιπτώσεων τους (Gil, Kott & Barabási, 2014).

Εξαιτίας των οικονομικών επιπτώσεων και των δυσμενών αποτελεσμάτων του κακόβουλου λογισμικού, η ανίχνευση του αποτελεί ένα από τα πλέον ενδιαφέροντα αντικείμενα στην κυβερνοασφάλεια. Για να προστατευθούν οι νόμιμοι χρήστες, η πιο σημαντική γραμμή άμυνας είναι τα αντιικά προγράμματα, τα οποία χρησιμοποιούν μεθόδους που βασίζονται σε μοναδικές υπογραφές (ένα σύνολο από σύντομα και μοναδικά αλφαριθμητικά) που προέρχονται από ήδη γνωστά κακόβουλα αρχεία. (Kephart & Arnold, 1994; Griffin et al., 2009). Συγκεκριμένα, ένα εκτελέσιμο αρχείο αναγνωρίζεται ως κακόβουλο αν η υπογραφή του ταιριάζει με τη λίστα των διαθέσιμων υπογραφών. Μια τέτοια προσέγγιση είναι γρήγορη και απλή για τον εντοπισμό γνωστού κακόβουλου λογισμικού με μικρό ποσοστό λάθους. Ωστόσο, η δημιουργία υπογραφής είναι μια σκληρή δουλειά που απαιτεί πολύ χρόνο, πόρους και το πιο σημαντικό, την τεχνογνωσία. Αυτό είναι το κύριο μειονέκτημα της μεθόδου αυτής. Το δεύτερο ζήτημα είναι ότι η μέθοδος που βασίζεται στις υπογραφές περιορίζεται στο να αναγνωρίζει ήδη γνωστό κακόβουλο λογισμικό κι ως εκ τούτου είναι αναξιόπιστο κι αναποτελεσματικό έναντι νέων, αόρατων κακόβουλων δειγμάτων. Όταν ένα άγνωστο δείγμα αναγνωρίζεται, είναι απολύτως απαραίτητο να ενημερωθεί η βάση υπογραφών, έτσι ώστε το δείγμα να καταγραφεί για να μπορεί να ανιχνευτεί από τον μηχανισμό ανάλυσης (Symantec, 2010; Comodo, 2016). Συνεπώς είναι απόλυτη ανάγκη να είναι εφικτή, η κατά το δυνατό ταχύτερη ανάλυση του άγνωστου δείγματος και η κατανόηση της συμπεριφοράς του και της επήρειας του στο σύστημα. Επιπλέον, η γνώση της

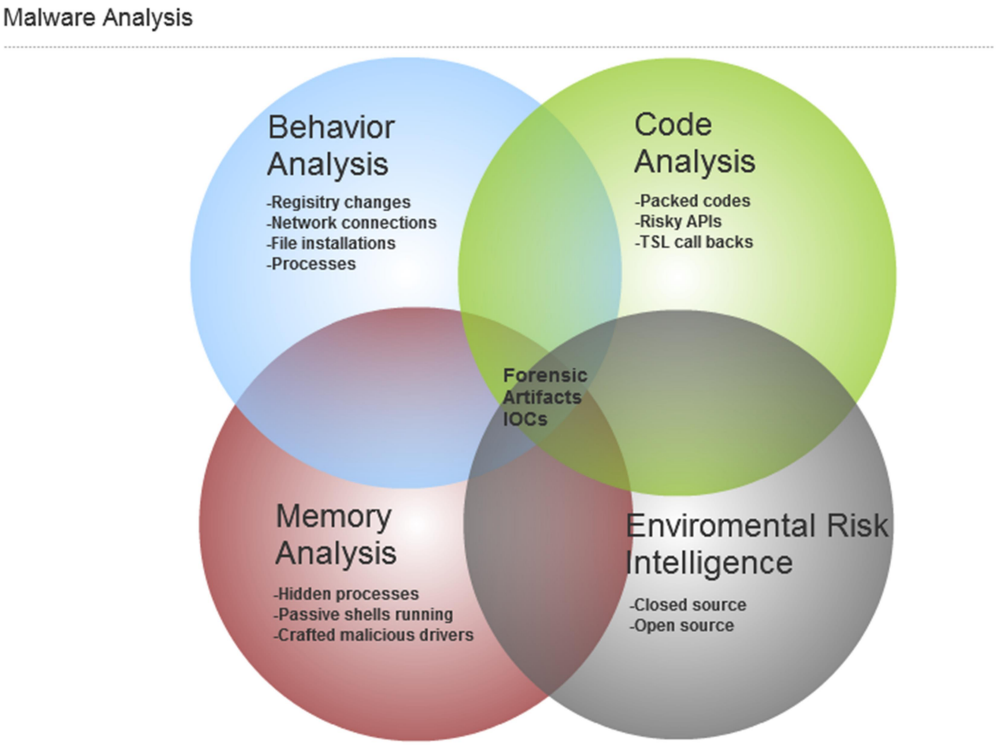
λειτουργίας του λογισμικού είναι απαραίτητη για την αφαίρεση του. Για να αφαιρεθεί εντελώς το λογισμικό, δεν είναι αρκετή συνήθως η διαγραφή του αρχείου. Είναι επίσης αναγκαία η αφαίρεση των κατάλοιπων που αφήνει πίσω, όπως για παράδειγμα εγγραφές στην μνήμη, υπηρεσίες ή διεργασίες, καθώς και η αντιστροφή των αλλαγών που έγιναν στα αρχικά αρχεία. Όλα τα παραπάνω απαιτούν λεπτομερή κατανόηση της συμπεριφοράς και της λειτουργικότητας του. Η παραδοσιακή προσέγγιση για την ανάλυση άγνωστου κακόβουλου λογισμικού, αναφέρεται στην εκτέλεση του αρχείου σε περιορισμένο και ελεγχόμενο περιβάλλον και ταυτόχρονα στην παρατήρηση των ενεργειών του. Το περιβάλλον αυτό είναι συχνά ένας εκσφαλματωτής, ο οποίος χρησιμοποιείται από έναν ερευνητή για την ανάγνωση βήμα με βήμα του κώδικα, ώστε να κατανοηθεί η λειτουργικότητα του. Δυστυχώς όμως οι εταιρίες δημιουργίας αντιικών προγραμμάτων, λαμβάνουν καθημερινά εκατοντάδες νέα δείγματα κάθε μέρα. Είναι πασιφανές ότι η χειροκίνητη ανάλυση της τους, καθίσταται ανεδαφική. Συνεπώς ανακύπτει η ανάγκη για αυτόματη ανάλυση. Μία προσέγγιση για την αυτοματοποιημένη ανάλυση είναι η εκτέλεση του δείγματος σε ένα εικονικό περιβάλλον ή σε ένα περιβάλλον που εξομοιώνει ένα λειτουργικό σύστημα. Κατά την εκτέλεση του δείγματος, η αλληλεπίδραση του με το σύστημα (π.χ. τοπικές κλίσεις ή οι Windows API που μπαίνουν σε λειτουργία), μπορούν να καταγραφούν και να αναλυθούν αργότερα. Αύτη η προσέγγιση απαλλάσσει τον αναλυτή από μία επίπονη εργασία η οποία θα ήταν η χειροκίνητη ανάλυση κάθε δείγματος το οποίο λαμβάνεται. Φυσικά η εμπλοκή του στην εκ των υστέρων ανάλυση, είναι απαραίτητη και επιθυμητή, μετά την αυτόματη ανάλυση. Τότε όμως θα έχει στην διάθεση του πληροφορίες, για τις ενέργειες του δείγματος, ώστε να οδηγήσουν και να βοηθήσουν τον αναλυτή.

Εξαιτίας της ραγδαίας ανάπτυξης της πληροφορικής, το κακόβουλο λογισμικό έχει καταστεί μία σοβαρή απειλή για τα υπολογιστικά συστήματα και τα δίκτυα υπολογιστών. Για παράδειγμα οι απειλές σε υπολογιστικά συστήματα και σε υπηρεσίες αυξάνονται συνεχώς, εκμεταλλευόμενες τις αδυναμίες της μαζικής κατασκευής λογισμικού με την χρήση έτοιμων πακέτων ανάπτυξης λογισμικού (Sun & Chen, 2009), όπως για παράδειγμα το Microsoft Visual Studio (Microsoft, 2016a).Επιπλέον στην εφαρμογή online συναλλαγών, τα trojan horses συχνά κλέβουν πολύτιμες πληροφορίες από τους χρήστες, χρησιμοποιώντας ηλεκτρονικό ψάρεμα (Abdelhamid, Ayesh & Thabtah, 2014).

Στην πραγματικότητα, με απλές τεχνικές σύγχυσης, μπορεί εύκολα να παρακάμψει κάποιος τις τεχνικές που βασίζονται στην ανίχνευση υπογραφών. Εκτός αυτού, παρακινούμενη από τα οικονομικά οφέλη, τα σημερινά δείγματα κακόβουλου λογισμικού δημιουργούνται με υψηλή ταχύτητα (χιλιάδες ανά ημέρα). Για παράδειγμα, η Symantec ανέφερε ότι 21,7 εκατομμύρια νέα κομμάτια του malware δημιουργήθηκαν τον Οκτώβριο του 2015 (Symantec, 2015) σύμφωνα με την έκθεση απειλών της McAfee Labs, υπήρχαν περισσότερα από 450 εκατομμύρια συνολικά δείγματα malware κατά το τέταρτο τρίμηνο του 2015 (McAfee Labs, 2015).

Προκειμένου να επιλυθούν τα προαναφερθέντα προβλήματα, η μέθοδος που βασίζεται σε ανίχνευση με ευρετικούς αλγόριθμους αναζήτησης, η οποία χρησιμοποιεί την εξόρυξη δεδομένων καθώς και τεχνικές μηχανικής μάθησης, έχει αναπτυχθεί για τη αντιμετώπιση ευφυούς κακόβουλου λογισμικού. Η προσέγγιση αυτή έχει ως στόχο να καταγραφούν ειδικά μοτίβα που περιγράφουν τα χαρακτηριστικά του κακόβουλου λογισμικού. Για παράδειγμα οι (Matthew G. Schultz *et al.*, 2001) εντόπισαν τρεις διαφορετικούς τύπους χαρακτηριστικών (πληροφορίες για τους πόρους του συστήματος, για το σύστημα, για τα εκτυπώσιμα σύμβολα και ακολουθίες byte) από τα δείγματα, στην συνέχεια εισάγοντας τα στους αλγόριθμους Ripper, Naive Bayes και Multi-Naive Bayes δημιούργησαν ταξινόμηση των δειγμάτων κακόβουλου κώδικα και των ασφαλών αρχείων. Από τις κλήσεις διεπαφής προγράμματος εφαρμογής (API) μπορούν να αναπαρασταθούν οι δράσεις από ένα εκτελέσιμο κι είναι μία από τις πιο αποτελεσματικές λειτουργίες που χρησιμοποιούνται από τις ευρετικές μεθόδους. Πολλές έρευνες έχουν γίνει με βάση τις κλήσεις API, συμπεριλαμβανομένων των (Hofmeyr, Forrest & Somayaji, 1998; Ye *et al.*, 2008) και ούτω καθεξής. Υπάρχουν άλλοι ερευνητές οι οποίοι δούλεψαν με άλλα σημαντικά χαρακτηριστικά (π.χ., οι οδηγίες του μηχανήματος) για την ανίχνευση κακόβουλου λογισμικού, όπως (Santos *et al.*, 2010; Shabtai *et al.*, 2012) και (Runwal, Low & Stamp, 2012). Παρά το γεγονός ότι οι εργασίες αυτές επιδεικνύουν θετικά αποτελέσματα ανίχνευσης, δεν έλαβαν τη σειρά των χαρακτηριστικών υπόψη κι ως εκ τούτου αδυνατούν να δημιουργήσουν μοτίβα ανίχνευσης με αξιοσημείωτη διαφορά μεταξύ κακόβουλου και φυσιολογικών αρχείων.

2.3 Τεχνικές ανάλυσης κακόβουλου λογισμικού



Εικόνα 2:Οι τέσσερεις περιοχές που εστιάζεται η ανάλυση κακόβουλου κώδικα (Franolich, 2012)

Η ανάλυση κακόβουλου λογισμικού είναι η διαδικασία καθορισμού του σκοπού και της λειτουργικότητας ενός συγκεκριμένου δείγματος (π.χ. ιός, σκουλήκι, ή δούρειος ίππος). Αυτή η διαδικασία είναι ένα απαραίτητο βήμα για να είναι σε θέση να αναπτυχτούν αποτελεσματικές τεχνικές ανίχνευσης για κακόβουλο κώδικα. Επιπλέον, είναι ένα σημαντικό προαπαιτούμενο για την ανάπτυξη των εργαλείων αφαίρεσης που μπορούν να διαγράψουν πλήρως το κακόβουλο λογισμικό από ένα μολυσμένο μηχάνημα. Παραδοσιακά, η ανάλυση του κακόβουλου λογισμικού είναι μια χειροκίνητη διαδικασία που είναι επίπονη και χρονοβόρα. Δυστυχώς, ο αριθμός των δειγμάτων που πρέπει να αναλυθούν από τις εταιρίες ασφαλείας σε καθημερινή βάση αυξάνεται συνεχώς. Αυτό αποκαλύπτει σαφώς την ανάγκη για εργαλεία που αυτοματοποιούν και την απλοποιούν τα μέρη της διαδικασίας ανάλυσης.

Ο πρωταρχικός στόχος της ανάλυσης κακόβουλου λογισμικού είναι η εκ βαθέων κατανόηση των προθέσεων που έχει, καθώς η κατανόηση της συμπεριφοράς και της εσωτερικής διαμόρφωσης του μέσω της ανάλυσης του κώδικα του δείγματος. Με αυτόν τον τρόπο θα καταγραφούν:

- Τα χαρακτηριστικά και οι δυνατότητες των κακόβουλων δείγματος υπό ανάλυση.
- Η δυνητική βλάβη που μπορεί να δημιουργηθεί.
- Οι δείκτες έκθεσης (IOC) οι οποίοι μπορεί να χρησιμοποιηθούν για τον προσδιορισμό μελλοντικών μολύνσεων ή για δραστηριότητα που χρησιμοποιήθηκε εντός του δικτύου ή του συστήματος αρχείων. Παρατηρούμενοι IOCs μπορεί να ενσωματωθούν σε αμυντικά συστήματα και στη διαδικασία αντιμετώπισης περιστατικών.

2.3.1 Στατική ανάλυση

Η ανάλυση άγνωστων εκτελέσιμων δεν είναι ένα καινούργιο πρόβλημα. Συνεπώς, πολλές λύσεις υπάρχουν ήδη. Αυτές οι λύσεις μπορούν να χωριστούν σε δύο κατηγορίες: στατική ανάλυση και δυναμική. Στις επόμενες παραγράφους θα καταγραφεί η στατική ανάλυση κώδικα και θα επισημανθούν οι σημαντικοί περιορισμοί, οι οποίοι κάνουν την δυναμική ανάλυση απαραίτητη.

Στην στατική ανάλυση, μια ομάδα αντιμετώπισης περιστατικών αναλύει τον κώδικα ή τη δομή ενός προγράμματος για να καθορίσει τη λειτουργία του χωρίς να τρέχει το πρόγραμμα (Sikorski & Honig, 2012). Τα πρώτα βήματα περιλαμβάνουν τη χρήση όλων των διαθέσιμων αντιικών προγραμμάτων. Αυτό θα μπορούσε να δώσει στοιχεία για ένα γνωστό κακόβουλο λογισμικό, για το οποίο υπάρχει διαθέσιμη υπογραφή, εξοικονομώντας σημαντικό χρόνο στη διαδικασία. Ένα σημαντικό μειονέκτημα αυτής της τεχνικής είναι η εξάρτηση από την ανίχνευση των αντιικών που στηρίζεται κυρίως σε ένα αρχείο με υπογραφές. Προγραμματιστές κακόβουλου κώδικα μπορούν να αλλάξουν εύκολα τον κώδικα με σκοπό να αποφύγουν τον εντοπισμό (Dalziel, 2014). Στην ιδανική περίπτωση, θα πρέπει να επιχειρήσουν την ανίχνευση χρησιμοποιώντας διαφορετικά αντιικά προγράμματα. Μια καλή πηγή για να επιτευχθεί αυτό χωρίς σημαντικές επενδύσεις είναι το VirusTotal (VirusTotal, no date) διαδικτυακή υπηρεσία που παρέχει τη δυνατότητα σάρωσης εκτελέσιμων των Windows, αρχείων Android APK, PDF, εικόνες και JavaScript ανάμεσα σε άλλες μορφές αρχείων. Το VirusTotal προσφέρει μια δωρεάν υπηρεσία στην κοινότητα και χρησιμοποιεί πάνω από 50 διαφορετικές μηχανές σάρωσης.

Η στατική ανάλυση είναι η διαδικασία της ανάλυσης των κώδικα μιας εφαρμογής, χωρίς να εκτελεστεί πραγματικά η εφαρμογή. Στην ανάλυση αυτή, ο δυαδικός κώδικας αποσυναρμολογείται πρώτα, γεγονός το οποίο αναφέρεται ουσιαστικά στην μετατροπή του δυαδικού κώδικα σε εντολές σε γλώσσα μηχανής. Στην συνέχεια οι τεχνικές ανάλυσης τόσο της ροής ελέγχου όσο και της ροής δεδομένων συνεπικουρούν για την παραγωγή αποτελεσμάτων της λειτουργίας ενός προγράμματος. Μία σειρά τεχνικών ανάλυσης δυαδικού κώδικα (Kruegel, Robertson & Vigna, no date; Christodorescu & Jha, 2003; Christodorescu *et al.*, 2005) έχουν παρουσιαστεί για την ανίχνευση διαφορετικών τύπων κακόβουλου λογισμικού. Το πλεονέκτημα της στατικής ανάλυσης είναι ότι διενεργείται γρήγορα και μπορεί να καλύψει το σύνολο του κώδικα της εφαρμογής. Ένα γενικό πρόβλημα που αντιμετωπίζει η συγκεκριμένη ανάλυση είναι ότι πολλές από τις ερωτήσεις που αφορούν μία εφαρμογή και τις ιδιότητες της, παραμένουν συνήθως αναπάντητες. Φυσικά υπάρχει πλούσια βιβλιογραφία, με τεχνικές στατικής ανάλυσης, πράγμα το οποίο υποδεικνύει ότι πολλά προβλήματα μπορούν να προσεγγιστούν καλά στην πράξη, συχνά επειδή είναι δύσκολο να συμβεί κάτι παράξενο στις πραγματικές εφαρμογές. Δυστυχώς η κατάσταση αντιστρέφεται όταν η ανάλυση αφορά κακόβουλο λογισμικό. Επειδή τέτοια λογισμικά έχουν κατασκευαστεί κατευθείαν από κυβερνοεγκληματίες μπορεί να είναι δημιουργημένα σκόπιμα έτσι, ώστε να είναι δύσκολο να αναλυθούν. Συγκεκριμένα, ο επιτιθέμενος μπορεί να κάνει χρήση τεχνικών δυαδικής σύγχυσης (binary obfuscation) για να εμποδίσει τόσο την αποσυναρμολόγηση του κώδικα, όσο και την ανάλυση του, μεθόδους που χρησιμοποιούν οι τεχνικές στατικής ανάλυσης.

Ο όρος “δυαδική σύγχυση” αναφέρεται σε τεχνικές οι οποίες διατηρούν τον σχεδιασμό και την λειτουργικότητα του κώδικα, αλλά παράλληλα δυσκολεύουν το έργο του αναλυτή, ο οποίος θέλει να εξάγει και να κατανοήσει την δομή του προγράμματος. Στο τομέα της αποσυναρμολόγησης, η σύγχυση αναφέρεται σε μετασχηματισμούς του δυαδικού κώδικα ώστε κατά το parsing των εντολών να είναι δύσκολο. Οι (Linn & Debray, 2003), εισήγαγαν καινοτόμες τεχνικές σύγχυσης οι οποίες εκμεταλλεύονται το γεγονός ότι οι εντολές των επεξεργαστών Intel x86, περιέχουν εντολές μεταβλητού μήκους οι οποίες μπορούν να ξεκινήσουν από μια αυθαίρετη διεύθυνση μνήμης. Εισάγοντας padding bytes σε περιοχές της μνήμης οι οποίες δεν μπορούν να προσπελαστούν κατά την εκτέλεση, οι αποσυναρμολογητές μπορούν να συγχυστούν και να καταλάβουν λάθος τον κώδικα. Αν κι η συγκεκριμένη έρευνα, περιορίζεται σε

επεξεργαστές Intel x86, τα αποτελέσματα της σύγχυσης απέναντι στους πιο σύγχρονους αποσυναρμολογητές είναι εκπληκτικά.

Εκτός από τις τεχνικές σύγχυσης, για να αυξήσουν την δυσκολία της διαδικασίας αποσυναρμολόγησης, ο κώδικας εν γένει μπορεί να είναι δύσκολος στην εξαγωγή της ροής ελέγχου ενός προγράμματος ή στην πραγματοποίηση ανάλυσης της ροής δεδομένων. Η βασική ιδέα πίσω από τέτοιες τεχνικές είναι να εφαρμόζονται αυτόματα, αλλά να είναι δύσκολο να αναιρεθούν, ακόμη κι αν η διαδικασία μετασχηματισμού είναι γνωστή. Αυτή η φιλοσοφία είναι παρόμοια με αυτήν που ονομάζεται "one-way translation process" (Wang, 2001), με την κρυπτογραφία. Στις δύο αυτές περιπτώσεις, η διαδικασία η οποία προτείνεται είναι εύκολο να εκτελεστεί προς μια κατεύθυνση, αλλά δύσκολο να αντιστραφεί. Ένα παράδειγμα για να γίνει κατανοητή μια τέτοια περίπτωση, είναι η χρήση μίας αρχής που ονομάζεται «αδιαφανείς σταθερές». Οι αδιαφανείς σταθερές είναι η επέκταση της ιδέας των αδιαφανών εκφράσεων, όπως ορίζονται στο (Collberg, Thomborson & Low, 1998) σαν εκφράσεις δυαδικών τιμών των οποίων οι τιμές είναι γνωστές στον συναρμολογητή αλλά δύσκολο να προσδιοριστούν σε μία αυτόματη αποσυναρμολόγηση. Η διαφορά μεταξύ αδιαφανών σταθερών και αδιαφανών εκφράσεων είναι ότι αδιαφανείς σταθερές δεν έχουν λογικές αλλά ακέραιες τιμές. Πιο συγκεκριμένα, αδιαφανείς σταθερές είναι μηχανισμοί για να φορτωθεί μια σταθερά σε έναν καταχωρητή, η τιμή των οποίων δεν μπορεί να προσδιοριστεί στατικά. Βασισμένος σε αδιαφανείς σταθερές, μπορεί κανείς να οικοδομήσει μια σειρά από μετασχηματισμούς συσκότισης που είναι δύσκολο να αναλυθούν στατικά. Για παράδειγμα, κάποιος μπορεί να αντικαταστήσει το στόχο των εντολών άμεσης μεταφοράς ελέγχου (όπως άλματα ή κλήσεις), με έμμεσες μεταβλητές που χρησιμοποιούν αδιαφανείς σταθερές ως στόχοι για άλμα. Ένας άλλος τομέας της εφαρμογής των αδιαφανών σταθερών είναι η θέση των δεδομένων και η συσκότιση της χρήσης δεδομένων. Η θέση ενός στοιχείου δεδομένων συχνά προσδιορίζεται παρέχοντας μια σταθερή, απόλυτη διεύθυνση ή μια σταθερή απόκλιση σε σχέση με ένα συγκεκριμένο καταχωρητή. Σε αμφότερες τις περιπτώσεις, το έργο ενός στατικού αναλυτή μπορεί να είναι περίπλοκο εάν το πραγματικό στοιχείο δεδομένων το οποίο προσπελάζεται είναι κρυφό. Με την συσκότιση στη χρήση δεδομένων, η ιχνηλάτηση των τιμών στους καταχωρητές περιπλέκεται από το γεγονός ότι το περιεχόμενο του καταχωρητή συχνά αποθηκεύεται και επαναφόρτωνεται από άγνωστες τοποθεσίες. Τέλος, ο κώδικας που αναλύεται με ένα στατικό αναλυτή δεν είναι απαραίτητα ο κώδικας που λειτουργεί πραγματικά. Ειδικότερα, αυτό είναι πραγματικότητα για τα

αυτοτροποποιούμενα προγράμματα που χρησιμοποιούν πολυμορφικές και μεταμορφικές τεχνικές (Szor, 2005) κι ενωμένα εκτελέσιμα που ξαναεμφανίζονται κατά τη διάρκεια της εκτέλεσης (Oberhumer,M., 2004).

Ο κατακερματισμός αρχείων είναι μια άλλη μέθοδος για τον εντοπισμό κακόβουλου λογισμικού. Μια καλή βάση για αυτό το είδος των πληροφοριών βρίσκεται στο (Malware Hash Registry) (MHR, no date). Αυτή η υπηρεσία είναι δωρεάν μόνο για μη εμπορική χρήση. Το MHR συμπληρώνει το αντιικό πρόγραμμα, βοηθώντας στον εντοπισμό γνωστών δειγμάτων κακόβουλου λογισμικού.

Άλλες μορφές ταυτοποίησης που χρησιμοποιούν στατικές μέθοδοι περιλαμβάνουν ανάλυση των συμβολοσειρών, των αρχείων DLL, τη καταγραφή των συναρτήσεων και την εξέταση συσκευασμένων εκτελέσιμων. Πολλές φορές ένα δείγμα κακόβουλου λογισμικού θα περιέχει κομμάτια που δείχνουν μια διεύθυνση IP ή το όνομα ενός κεντρικού υπολογιστή.

2.3.2 Δυναμική ανάλυση

Οι τεχνικές δυναμικής ανάλυσης της συμπεριφοράς κακόβουλου λογισμικού χαρακτηρίζονται από την ανάλυση των πραγματικών οδηγιών εκτέλεσης ενός προγράμματος ή από τα αποτελέσματα που φέρνει το πρόγραμμα αυτό στο λειτουργικό σύστημα. Σε σύγκριση με την στατική τεχνική, η δυναμική ανάλυση είναι λιγότερο ευάλωτη σε διάφορες τεχνικές συσκότισης κώδικα (Moser, Kruegel & Kirda, 2007b). Οι (Christodorescu, Jha & Kruegel, 2007) εισάγουν τις προδιαγραφές του κακόβουλου λογισμικού στις ροές δεδομένων μεταξύ των κλήσεων συστήματος, οι οποίες καταγράφουν τις πραγματικές σχέσεις μεταξύ των κλήσεων συστήματος και είναι δύσκολο να παρακαμφθούν με τυχαίες κλήσεις συστήματος. Από τότε, οι εν λόγω λεπτομέρειες του κακόβουλου λογισμικού έχουν χρησιμοποιηθεί ευρέως σε εργασίες ανάλυσης κακόβουλου λογισμικού, όπως η εξαγωγή διακριτών λειτουργιών κακόβουλου λογισμικού εξορύσσοντας τη διαφορά μεταξύ της συμπεριφοράς κακόβουλου λογισμικού και την καλοήθη συμπεριφορά του προγράμματος (Fredrikson *et al.*, 2010), τον καθορισμό οικογενειών κακόβουλου λογισμικού στις οποίες τα δείγματα μοιράζονται κοινές λειτουργίες (Bayer *et al.*, 2009; Babić, Reynaud & Song, 2011; Park, Reeves & Stamp, 2013) και την ανίχνευση κακόβουλης συμπεριφοράς (Martignoni *et al.*, 2008; Kolbitsch *et al.*, 2009; Bayer, Kirda & Kruegel, 2010). Μια άλλη

μέθοδος χρησιμοποιεί έναν αντιπροσωπευτικό κοινό γράφημα συμπεριφοράς για όλα τα δείγματα κακόβουλου λογισμικού σε μια οικογένεια, αντί ενός γραφήματος συμπεριφοράς ανά περίπτωση. Η προτεινόμενη προσέγγιση είναι έγκυρη και αποτελεσματική δεδομένου ότι τα περισσότερα νέα κακόβουλα λογισμικά είναι παραλλαγές γνωστών οικογενειών (Gordon, no date; Park *et al.*, 2010; Vlachos, Ilioudis & Papanikolaou, 2012; Park, Reeves & Stamp, 2013). Παρά τις διάφορες μεταμορφικές και πολυμορφικές συσκοτίσεις, δείγματα κακόβουλου λογισμικού μέσα στην ίδια οικογένεια τείνουν να αποκαλύψουν παρόμοια κακόβουλη συμπεριφορά (Lindorfer *et al.*, 2012). Η πορεία των δράσεων που αναλαμβάνονται κατά τη διάρκεια της ανάλυσης κακόβουλου κώδικα απαιτεί την εκτέλεση του υπό διερεύνηση εκτελέσιμου δείγματος σε ένα ελεγχόμενο απομονωμένο και ασφαλές περιβάλλον. Η μεθοδολογία δυναμικής ανάλυσης επιτρέπει να προσδιοριστεί η συμπεριφορά του κακόβουλου λογισμικού και να καταγραφεί ο τρόπος που αλληλεπιδρά με το δίκτυο, το σύστημα αρχείων, το μητρώο και άλλα (Martin, 2016).

Οι υπεύθυνοι για περιστατικά ασφαλείας ανταποκρινόμενοι σε μια εισβολή, χρησιμοποιούν μερικές φορές την ανάλυση κακόβουλου λογισμικού κατά τη διάρκεια της αντιμετώπισης περιστατικών. Η αποτελεσματική αντιμετώπιση απαιτεί την ορθή εφαρμογή και εκτέλεση της αναγνώρισης, της ανάσχεσης, της εξάλειψης και της διαδικασίας ανάκτησης (Jason T. Luttgens, Pepe & Mandia, 2014). Ένα από τα διδάγματα που αντλήθηκαν κατά τη διάρκεια της αντιμετώπισης περιστατικών είναι η δημιουργία υπογραφών για τερματικά ή δικτυακά συστήματα ανίχνευσης και αποτροπής εισβολών. Η σωστή δημιουργία υπογραφής θα βελτιώσει την ανίχνευση σε περίπτωση μελλοντικών κινδύνων.

Ένας από αυτούς τους δείκτες είναι το file hash (Sikorski & Honig, 2012). Από όλους τους δείκτες της έκθεσης (IOCs), αυτός είναι ο πιο αδύναμος. Είναι πολύ εύκολο να αλλαχτεί ένα κομμάτι κώδικα για να περάσει έναν κανόνα ανίχνευσης που βασίζεται σε ένα hash. Η πλήρης κατανόηση του κακόβουλου κώδικα απαιτεί την αποσυναρμολόγηση και την ανάλυση της κάθε λειτουργίας και το σύστημα εκτέλεσης κλήσεων. Αυτή η διαδικασία είναι εξαιρετικά επίπονη, απαιτεί υψηλό βαθμό εξειδίκευσης και μερικές φορές παίρνει χρόνια για να ολοκληρωθεί. Σύμφωνα με το ινστιτούτο AV-TEST (*AV-TEST*, no date) το 2014 ο συνολικός αριθμός των δειγμάτων κακόβουλου λογισμικού που ανιχνεύτηκαν ήταν πάνω από 140.000.000. Η προσπάθεια να αναλυθεί το κάθε κομμάτι του κακόβουλου λογισμικού, με αντίστροφη μηχανική είναι μια αδύνατη αποστολή.

Μία ταχύτερη προσέγγιση είναι να επικεντρωθεί στις συνέπειες που ένα κακόβουλο λογισμικό επιφέρει σε ένα σύστημα που μολύνει. Δείκτες όπως τα του κλειδιά μητρώου, τα εισαγόμενα αρχεία, οι συνδέσεις, τα ερωτήματα DNS και οι συμβολοσειρές που κατευθύνουν σε συγκεκριμένες θέσεις ή πρόσωπα. Οι δείκτες αυτοί βοηθούν στη δημιουργία των υπογραφών. Δεν υπάρχει ανάγκη να αναλυθεί το δείγμα. Ο έλεγχος κακόβουλου λογισμικού με αυτό τον τρόπο απαιτεί ο κώδικας να τρέξει σε ένα σύστημα για να παρατηρηθεί η συμπεριφορά του. Η δράση αυτή θα μολύνει το σύστημα και θα το κάνει ανασφαλές. Αυτός είναι ο λόγος που το κακόβουλο λογισμικό συνήθως δοκιμάζεται χρησιμοποιώντας ένα απομονωμένο σύστημα.

Τι θα συμβεί αν το κακόβουλο λογισμικό προσπαθήσει να δημιουργήσει ένα ερώτημα DNS και αποτύχει; Μια λύση είναι να δημιουργηθεί μια «μαύρη τρύπα Διαδικτύου» (black hole Internet). Η μαύρη τρύπα θα δράσει ως καταβόθρα για όλες τις δραστηριότητες που διενεργεί το μολυσμένο σύστημα. Αν το κακόβουλο λογισμικό κάνει μια αίτηση DNS για το «evil.com», η μαύρη τρύπα θα μπορούσε να απαντήσει με μια διεύθυνση IP υπό τον έλεγχό μας. Η μαύρη τρύπα φιλοξενεί επίσης τις πιο κοινές υπηρεσίες που ζητούνται από κακόβουλο λογισμικό, όπως το HTTP, FTP και SSL. Ένα παράδειγμα ενός τέτοιου συστήματος είναι FakeNet (*FakeNet*, 2012).

Βασική δυναμική ανάλυση είναι η συνέχεια της στατικής ανάλυσης. Σε αυτή τη φάση, το κακόβουλο λογισμικό εκτελείται και παρατηρείται η συμπεριφορά του. Η πιο δημοφιλής μέθοδος ανάλυσης της λειτουργίας κακόβουλου λογισμικού με έναν ασφαλή τρόπο είναι η χρήση της τεχνολογίας sandbox. Το sandbox εκτελείται ως ένα ξεχωριστό σύστημα, περιέχει το μη αξιόπιστο πρόγραμμα και αποτρέπει οποιαδήποτε ενέργεια από την πρόσβαση του πραγματικού δικτύου. Τα Sandboxes συχνά παρέχουν υπηρεσίες δικτύου για το κακόβουλο λογισμικό σε μια μορφή «μαύρης τρύπας». Εάν το μη αξιόπιστο πρόγραμμα κάνει μια αίτηση DNS για παράδειγμα, το sandbox θα απαντήσει την ερώτηση, συνήθως με 127.0.0.1 (loopback).

Μια σειρά από υλοποιήσεις sandbox υπάρχουν ήδη και είναι έτοιμες προς χρήση. Μερικές από αυτές τις επιλογές είναι Anubis, BitBlaze, EUREKA, ViCheck (Zeltser, 2016). Οι περισσότερες από αυτές τις υπηρεσίες είναι διαθέσιμες χωρίς κόστος. Ωστόσο, ορισμένοι οργανισμοί προτιμούν να δημιουργούν τα δικά τους εσωτερικά sandbox λόγω της πολιτικών ή άλλων εταιρικών απαιτήσεων. Η δημιουργία ενός ιδιόκτητου sandbox θα μπορούσε να είναι μια ακριβή επιλογή, λόγω των πολλών ωρών εργασίας που απαιτούνται για την εγκατάσταση και την ρύθμιση των πολλών κομματιών του λογισμικού.

Από τότε που ξεκίνησε η εξάπλωση των μεταμορφικών ιών, η δυναμική ανάλυση κακόβουλου λογισμικού έχει εδραιωθεί σε μια αποτελεσματική προσέγγιση για την κατανόηση και την κατηγοριοποίηση κακόβουλου λογισμικού, παρατηρώντας την εκτέλεση των δειγμάτων κακόβουλου λογισμικού σε περιβάλλον καραντίνας (Willems, Holz & Freiling, 2007; Egele *et al.*, 2012). Η αλληλεπίδραση μεταξύ της εκτέλεσης του δείγματος του κακόβουλου λογισμικού και του λειτουργικού συστήματος επιτρέπει σε δυναμικά συστήματα ανάλυσης κακόβουλου λογισμικού, να συλλέξουν εκείνα τα χαρακτηριστικά συμπεριφοράς που βοηθούν στη διαμόρφωση τεχνικών άμυνας. Έτσι, στο πλαίσιο της ανάλυσης κακόβουλου λογισμικού η κύρια απαίτηση είναι η συλλογή πληροφοριών από ένα κακόβουλο λογισμικό που ο αναλυτής θεωρεί πολύτιμο.

Οι τρέχουσες προσεγγίσεις για την αυτόματη ανάλυση πάσχουν από ορισμένες ελλείψεις. Ένα πρόβλημα είναι ότι ο κακόβουλος κώδικας είναι συχνά εξοπλισμένος με ρουτίνες ανίχνευσης που ελέγχουν για την παρουσία μιας εικονικής μηχανής ή για προσομοιωμένο περιβάλλον λειτουργικού συστήματος. Όταν ένα τέτοιο περιβάλλον έχει εντοπιστεί, το κακόβουλο λογισμικό τροποποιεί την συμπεριφορά του κι η ανάλυση αποδίδει λανθασμένα αποτελέσματα. Το κακόβουλο λογισμικό ελέγχει επίσης για λογισμικό (ακόμα και υλικό) που έχει σημεία διακοπής (breakpoints) για να ανιχνεύσει αν το πρόγραμμα εκτελείται σε ένα πρόγραμμα εντοπισμού σφαλμάτων. Αυτό προϋποθέτει ότι το περιβάλλον ανάλυσης θα είναι αόρατο για τον κακόβουλο κώδικα. Ένα άλλο πρόβλημα είναι όταν το περιβάλλον ανάλυσης δεν παρακολουθεί την πλήρη αλληλεπίδραση με το σύστημα. Όταν συμβαίνει αυτό, ο κακόβουλος κώδικας θα μπορούσε να αποφύγει την ανάλυση. Αυτό θα ήταν δυνατό, επειδή υπάρχουν χιλιάδες κλήσεις API των Windows, συχνά με παραμέτρους που αποτελούνται από σύνθετες δομές δεδομένων. Επιπλέον, ο κακόβουλος κώδικας θα μπορούσε να αλληλεπιδράσει άμεσα με το λειτουργικό σύστημα μέσω native κλήσεις συστήματος. Έτσι, το περιβάλλον ανάλυσης θα πρέπει να είναι ολοκληρωμένο και να καλύπτει όλες τις πτυχές της αλληλεπίδρασης ενός προγράμματος με το περιβάλλον του.

2.3.3 Ανάλυση σε εικονικό περιβάλλον

Σε αντίθεση με την στατική ανάλυση, η δυναμική τεχνική αναλύει τον κώδικα κατά την εκτέλεση του. Ενώ αυτές οι τεχνικές δεν είναι εξαντλητικές, έχουν το σημαντικό πλεονέκτημα ότι μόνο ο κώδικας που εκτελείται στην πραγματικότητα αναλύεται. Έτσι, η δυναμική ανάλυση έχει ανοσία σε προσπάθειες συσκότισης και δεν έχει κανένα πρόβλημα με την αυτό-τροποποίηση των προγραμμάτων. Κατά τη χρήση δυναμικών

τεχνικών ανάλυσης, τίθεται το ερώτημα σε ποιο περιβάλλον θα πρέπει να εκτελεστεί το δείγμα. Φυσικά, η εκτέλεση κακόβουλου λογισμικού απευθείας στον υπολογιστή του αναλυτή, ο οποίος πιθανότατα συνδέεται με το Internet, θα μπορούσε να αποβεί ολέθρια, καθώς ο κακόβουλος κώδικας θα μπορούσε εύκολα να ξεφύγει και να μολύνει άλλα μηχανήματα. Επιπλέον, η χρήση ενός ειδικά απομονωμένου μηχανήματος που επανέρχεται μετά από κάθε δυναμική δοκιμή δεν αποτελεί ικανοποιητική λύση εξαιτίας της επιβάρυνσης σε χρόνο. Τρέχοντας το εκτελέσιμο σε μια εικονική μηχανή (δηλαδή, ένα εικονικό υπολογιστή), όπως αυτό που παρέχεται από την VMware (VMware, 2016), είναι μια δημοφιλής λύση. Στην περίπτωση αυτή, το κακόβουλο λογισμικό μπορεί να επηρεάσει μόνο τον εικονικό υπολογιστή κι όχι το πραγματικό. Μετά την εκτέλεση μιας δυναμικής ανάλυσης, η μολυσμένη εικόνα του σκληρού δίσκου απλά απορρίπτεται κι αντικαθίσταται από ένα καθαρό αντίγραφο (τα λεγόμενα στιγμιότυπα). Οι λύσεις virtualization είναι σχετικά γρήγορες. Δεν υπάρχει σχεδόν καμία διαφορά από το να τρέχει το εκτελέσιμο στον πραγματικό υπολογιστή κι η αποκατάσταση από μια καθαρή εικόνα αποτελεί πολύ πιο γρήγορη διαδικασία από ότι η εγκατάσταση του λειτουργικού συστήματος σε ένα φυσικό μηχάνημα. Δυστυχώς, ένα σημαντικό μειονέκτημα είναι ότι το εκτελέσιμο που θα αναλυθεί, μπορεί να ανιχνεύσει ότι τρέχει σε ένα εικονικό μηχάνημα και, ως εκ τούτου, να τροποποιήσει τη συμπεριφορά του. Στην πραγματικότητα, ένας αριθμός διαφορετικών μηχανισμών έχουν δημοσιευθεί (Robin & Irvine, 2000; Rutkowska, 2004) που εξηγούν τον τρόπο με τον οποίο ένα πρόγραμμα μπορεί να ανιχνεύσει εάν εκτελείται μέσα σε μια εικονική μηχανή. Φυσικά, αυτοί οι μηχανισμοί είναι επίσης διαθέσιμοι για χρήση από τους δημιουργούς κακόβουλου λογισμικού.

Ένας υπολογιστής εξομοιωτής είναι ένα λογισμικό που προσομοιώνει έναν προσωπικό υπολογιστή (PC), συμπεριλαμβανομένου του επεξεργαστή, της κάρτας γραφικών, του σκληρού δίσκου και άλλους πόρους, με σκοπό να τρέχει ένα λειτουργικό σύστημα χωρίς τροποποιήσεις. Είναι σημαντικό να γίνει διάκριση των εξομοιωτών (emulators) από τις εικονικές μηχανές όπως το VMware. Όπως οι εξομοιωτές PC, οι μηχανές εξομοίωσης μπορούν να τρέξουν ένα λειτουργικό σύστημα χωρίς τροποποιήσεις, αλλά εκτελούν στατιστικά ένα υποσύνολο των άμεσων εντολών στην πραγματική CPU. Αυτό έρχεται σε αντίθεση με τους εξομοιωτές PC, οι οποίοι προσομοιώνουν όλες τις οδηγίες στο λογισμικό. Επειδή όλες οι οδηγίες εξομοιώνονται σε λογισμικό, το σύστημα εμφανίζεται ακριβώς όπως ένα πραγματικό μηχάνημα σε ένα εκτελέσιμο πρόγραμμα, διατηρούν όμως τον πλήρη έλεγχο. Έτσι, είναι πιο δύσκολο για ένα πρόγραμμα να ανιχνεύσει ότι

έχει εκτελεστεί μέσα σε ένα εξομοιωτή υπολογιστή από ό, τι σε ένα εικονικό περιβάλλον.

Εκτός από τη διαφοροποίηση του τύπου του περιβάλλοντος που χρησιμοποιείται για τη δυναμική ανάλυση, μπορεί κανείς να διακρίνει και να ταξινομήσει διαφορετικά είδη πληροφοριών που μπορούν να συλλεχτούν κατά τη διαδικασία της ανάλυσης. Πολλά συστήματα εστιάζουν στην αλληλεπίδραση μεταξύ μιας εφαρμογής και του λειτουργικού συστήματος και αναχαιτίζουν τις κλήσεις συστήματος ή πιάνουν τις κλήσεις Windows API. Για παράδειγμα, ένα σύνολο εργαλείων που παρέχονται από το Sysinternals (Russinovich, M., Cogswell, no date) επιτρέπουν ο αναλυτής να καταγράψει όλες τις διεργασίες που εκτελούνται στα Windows (παρόμοιο με το Windows Task Manager) ή να καταγράψει όλα τις τροποποιήσεις του μητρώου των Windows και την δραστηριότητα του συστήματος αρχείων. Τα εργαλεία αυτά υλοποιούνται ως οδηγοί λειτουργικού συστήματος, οι οποίοι παρακολουθούν τις κλήσεις συστήματος των Windows. Αυτό έχει ως αποτέλεσμα να είναι αόρατα σε μία εφαρμογή που αναλύεται. Δεν μπορούν ωστόσο να υποκλέψουν και να αναλύσουν τις κλήσεις Windows API ή άλλες λειτουργίες του χρήστη.

Από την άλλη πλευρά, υπάρχουν εργαλεία (Hunt & Brubacher, 1999) που μπορούν να υποκλέψουν τις δραστηριότητες του χρήστη, συμπεριλαμβανομένων όλων των κλήσεων Windows API. Αυτό συνήθως πραγματοποιείται ξαναγράφοντας κάποιες συναρτήσεις. Η αρχική λειτουργία διατηρείται ως υπορουτίνα και εκτελείται μέσα από ένα «τραμπολίνο». Δυστυχώς, το γεγονός ότι ο κώδικας πρέπει να τροποποιηθεί, τον κάνει ανιχνεύσιμο στον κακόβουλο κώδικα που πραγματοποιεί έλεγχο της ακεραιότητας.

2.4 Το λογισμικό Cuckoo

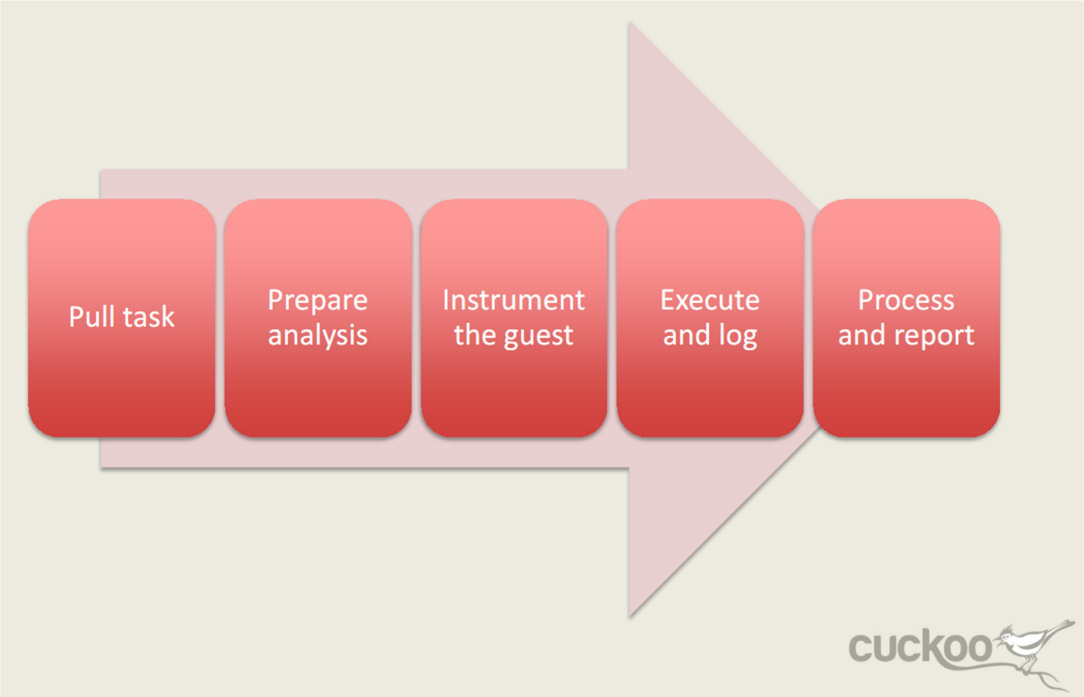
Η αναγνώριση και η ανάλυση του κακόβουλου λογισμικού είναι μία από τις ενέργειες που εκτελούνται από χειριστές περιστατικών ασφαλείας. Μόνο ένας μικρός αριθμός εταιριών παρέχει την απαραίτητη τεχνολογία για την αυτοματοποίηση αυτή. Τις περισσότερες φορές αυτές οι επιλογές είναι πέρα από την δυνατότητα των μικρών οργανισμών, λόγω του υψηλού κόστους που συνδέεται με την αδειοδότηση και τη συντήρηση. Ένα από τα δωρεάν πακέτα λογισμικού που επιτρέπουν σε έναν οργανισμό να δημιουργήσει το δικό του εσωτερικό περιβάλλον δοκιμών κακόβουλου κώδικα, είναι το Cuckoo Sandbox (Oktavianto & Muhardianto, 2013). Πρόκειται για ένα έργο

ελεύθερου λογισμικού σύμφωνα με την άδεια GNU GPLv3. Επιτρέπει στο χρήστη να αναλύσει και να συλλέξει στοιχεία εναντίον υπόπτων δειγμάτων κακόβουλου λογισμικού. Η εγκατάσταση και η παραμετροποίηση απαιτεί προσεκτική διαμόρφωση. Με τη χρήση τεχνικών που δανείστηκε από τις μεθόδους DevOps, μια μικρή ομάδα των ερευνητών ασφαλείας μπορεί να δημιουργήσει ένα περιβάλλον sandbox που δεν είναι μόνο επαναλαμβανόμενο και συνεπές, αλλά και επεκτάσιμο. Ο χρήστης μπορεί να δημιουργήσει πολλαπλά πρότυπα «προφίλ», το οποίο επιτρέπει την ευέλικτη δοκιμή.

Τα script σε Python κι οι βιβλιοθήκες του αποτελούν σημαντικά συστατικά του (Hosmer, 2014). Το σύστημα αποτελείται από έναν κεντρικό υπολογιστή όπου έχει εγκατασταθεί το λογισμικό Cuckoo, ένα πρόγραμμα virtualization, όπως το Oracle VirtualBox (Dash, 2013) και τον πράκτορα του Cuckoo που τρέχει μέσα σε μία εικονική μηχανή. Το φιλοξενούμενο σύστημα θα είναι μια εικονική μηχανή που εκτελεί μία έκδοση των Windows. Αυτή η εικονική μηχανή θα έχει απενεργοποιημένα το User Account Control (UAC), τις αυτόματες ενημερώσεις και το τοίχος προστασίας.

Μία μεθοδολογία που χρησιμοποιείται για την κατανόηση κακόβουλου κώδικα είναι το περιβάλλον δοκιμών sandboxing (Greamo & Ghosh, 2011). Με απλούς όρους, η διαδικασία αυτή περιλαμβάνει την εκτέλεση κακόβουλου κώδικα με ελεγχόμενο τρόπο που να επιτρέπει την άμεση παρατήρηση των αποτελεσμάτων. Με την τεκμηρίωση αποδεικτικών, όπως ανοικτών θυρών, κλειδιά μητρώου, διευθύνσεις IP, αρχείων που ενσωματώθηκαν και τα ονόματα τομέων, μια ομάδα μπορεί να κερδίσει πληροφορίες σχετικά με την τακτική του αντιπάλου.

Το απομονωμένο περιβάλλον επιτρέπει στο δείγμα να τρέξει χωρίς να επηρεάζει δυσμενώς το σύστημα του υπολογιστή φιλοξενίας ή τον επισκέπτη. Μια φιλοξενούμενη εικονική μηχανή των Windows έχει κατασκευαστεί και διαμορφωθεί μέσα σε μία υποστηριζόμενη εικονική μηχανή (VM), όπως το λογισμικό VirtualBox. Μόλις επιτευχθεί η επιθυμητή κατάσταση του συστήματος λαμβάνεται ένα στιγμιότυπο συστήματος. Αυτό το στιγμιότυπο μπορεί να χρησιμοποιηθεί για να επανέλθει το σύστημα σε μια γνωστή, καθαρή κατάσταση μετά την ανάλυση του δείγματος. Μια διεπαφή γραμμής εντολών χρησιμοποιείται για να εκτελούνται όλες οι εντολές στο πλαίσιο Cuckoo Sandbox.



Εικόνα 3: Η διαδικασία ανάλυσης με την χρήση του Cuckoo (Bremer, 2014)

2.5 Νομική κατοχύρωση ψηφιακών τεκμηρίων

Κάθε αποδεικτική πληροφορία που αποθηκεύεται ή μεταδίδεται ψηφιακά, η οποία μπορεί να χρησιμοποιηθεί από ένα μέρος σε μια νομική διαμάχη στο Δικαστήριο, αναφέρεται ως ψηφιακή απόδειξη (U.S. Legal, no date). Τα τελευταία χρόνια η χρήση των ψηφιακών αποδεικτικών στοιχείων έχει αυξηθεί εκθετικά. Κατά συνέπεια, είναι κρίσιμο για τα ψηφιακά αποδεικτικά στοιχεία να υπάρχει η δυνατότητα επικύρωσης και να διατηρείται η αποδεικτική αξία των ψηφιακών αποδεικτικών στοιχείων. Οι ψηφιακές αποδείξεις θα πρέπει να εξεταστούν προσεκτικά από το δικαστήριο πριν να θεωρηθούν αξιόπιστες. Επιπλέον, είναι θεμελιώδους σημασίας να γίνει διάκριση μεταξύ των τοπικών και απομακρυσμένων ψηφιακών αποδεικτικών στοιχείων.

Τα ψηφιακά στοιχεία υπάρχουν παντού, δεδομένου ότι μπορεί να βρίσκονται οπουδήποτε στον κόσμο. Μπορούν εύκολα να μετακινηθούν από τη μία συσκευή σε μία άλλη κάπου στον κόσμο. Μπορεί να διατηρούνται σε κινητό εξοπλισμό (τηλέφωνα, PDA, φορητούς υπολογιστές, GPSes, κλπ) και ιδιαίτερα σε διακομιστές που παρέχουν υπηρεσίες μέσω Διαδικτύου. Η συσκευή υπό έρευνα μπορεί να βρίσκεται σε μια διαφορετική χώρα από αυτή στην οποία το έγκλημα έχει διαπραχθεί, το οποίο μπορεί

να είναι ένα εμπόδιο για την απόκτηση των ψηφιακών τεκμηρίων. Τα ψηφιακά τεκμήρια μπορούν εύκολα να παραποιηθούν από τον ιδιοκτήτη της συσκευής, δεδομένου ότι ο ιδιοκτήτης έχει πλήρη πρόσβαση σε οποιαδήποτε στοιχεία υλικού και λογισμικού. Στην περίπτωση κατά την οποία τα στοιχεία είναι αποθηκευμένα σε απομακρυσμένη τοποθεσία, θα μπορούσαν να αλλοιωθούν ή να χαθούν σε βάθος χρόνου.

Μια ψηφιακή απόδειξη αναφέρεται ως τοπική σε περίπτωση που η πληροφορία αποθηκεύεται σε μια συσκευή που ανήκει στον κατηγορούμενο. Στις περισσότερες περιπτώσεις, το δικαστήριο μπορεί να διατάξει την κατάσχεση της ερευνώμενης συσκευής. Τέτοια στοιχεία μπορεί να εξαχθούν από τα ψηφιακά έγγραφα, από το ιστορικό περιήγησης και ούτω καθεξής. Η αλλοίωση τοπικών στοιχείων είναι εύκολη για έναν κατηγορούμενο που διαθέτει βασικές τεχνικές δεξιότητες.

Η απομακρυσμένη απόδειξη σχετίζεται με πληροφορίες που είναι αποθηκευμένες σε έναν απομακρυσμένο υπολογιστή (Maio, 2014). Σε αυτή την περίπτωση είναι δύσκολο να παραποιηθούν από τον κατηγορούμενο, δεδομένου ότι απαιτείται μη εξουσιοδοτημένη πρόσβαση στο απομακρυσμένο σύστημα ή παρέμβαση ενός συνεργού. Απομακρυσμένα ψηφιακά αποδεικτικά στοιχεία μπορούν να εξαχθούν από το κοινωνικά δίκτυα, τα e-mail που αποθηκεύονται σε ένα διακομιστή και ούτω καθεξής. Μπορούν να θεωρηθούν πιο αξιόπιστα και να επικυρωθούν, κατά μία έννοια, από την εταιρεία που παρέχει την υπηρεσία (Google, Facebook, κλπ). Στην πράξη, η εταιρεία μπορεί να ενεργεί ως αξιόπιστη τρίτη πηγή στη δίκη.

Οι ερευνητές κακόβουλου κώδικα στηρίζονται σε μεγάλο βαθμό στην ιατροδικαστική διατήρηση των μη μόνιμων στοιχείων (Malin, Casey & Aquilina, 2013). Επειδή η λειτουργία ενός ύποπτου υπολογιστή αλλάζει συνήθως το σύστημα, πρέπει να ληφθεί μέριμνα ώστε να ελαχιστοποιηθούν οι αλλαγές που έγιναν στο σύστημα, πρέπει να γίνει συλλογή των μη μόνιμων στοιχείων πρώτα με την διαδικασία την οποία περιγράφεται λεπτομερώς στο (*RFC 3227 - Guidelines for Evidence Collection and Archiving*, no date) .

Στο Ηνωμένο Βασίλειο, ο Association of Chief Police Officers (ACPO) έχει εξετάσει τα ψηφιακά αποδεικτικά στοιχεία από την άποψη της δικανικής υπολογιστών και καταγράφει τα αποτελέσματα στο φυλλάδιο με τίτλο «Good Practice Guide for Computer Based Electronic Evidence» (Williams, 2011; CERT-EU, 2012), από το οποίο

ξεχωρίζουν τέσσερις βασικές αρχές για τον χειρισμό των ψηφιακών αποδεικτικών στοιχείων:

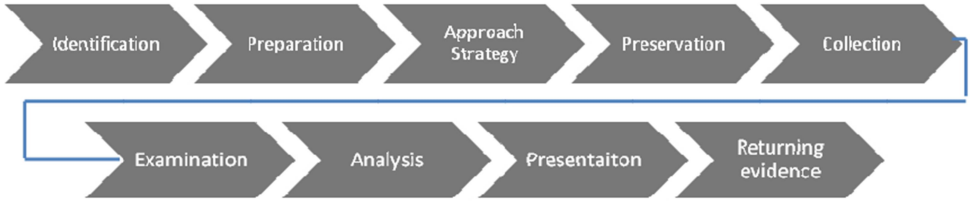
- Αρχή 1: Κανένα μέτρο που λαμβάνεται από τις υπηρεσίες επιβολής του νόμου ή τους αντιπροσώπους τους, δεν θα πρέπει να αλλάξει τα δεδομένα που τηρούνται σε υπολογιστή ή μέσο αποθήκευσης, το οποίο μπορεί στη συνέχεια να γίνει πειστήριο στο δικαστήριο.
- Αρχή 2: Σε περιπτώσεις όπου ένα άτομο θεωρεί ότι είναι απαραίτητο για την προσπέλαση των πρωτότυπων δεδομένων που τηρούνται σε έναν υπολογιστή ή σε μέσο αποθήκευσης, το πρόσωπο αυτό θα πρέπει να είναι εξουσιοδοτημένο να το κάνει, έτσι ώστε είναι σε θέση να δώσει αποδεικτικά στοιχεία που εξηγούν τη συνάφεια και τις συνέπειες των πράξεών τους.
- Αρχή 3: Η ιχνηλάτηση ελέγχου (audit trail) ή άλλες καταγραφές όλων των διεργασιών που εφαρμόζονται σε ηλεκτρονικά αποδεικτικά στοιχεία σε θα πρέπει να δημιουργούνται και να διατηρούνται. Μία ανεξάρτητη τρίτη οντότητα θα πρέπει να είναι σε θέση να εξετάσει αυτές τις διαδικασίες και να επιτύχει το ίδιο αποτέλεσμα.
- Αρχή 4: Ο υπεύθυνος της έρευνας (ο αξιωματικός της υπόθεσης) έχει τη συνολική ευθύνη για τη διασφάλιση ότι το δίκαιο και οι αρχές αυτές τηρούνται.

Οι τρέχουσες μέθοδοι συλλογής ψηφιακών αποδεικτικών στοιχείων βασίζονται σε πρώιμες εμπειρίες μονάδων επιβολής του νόμου στη συλλογή αποδεικτικών στοιχείων από τους σκληρούς δίσκους. Κατά την τελευταία δεκαετία, ένα πρότυπο για αυτή τη διαδικασία δημιουργήθηκε, αναπτύχθηκε και βελτιώθηκε. Στην ανάπτυξη της διαδικασίας για τα ψηφιακά αποδεικτικά στοιχεία, η Τεχνική Ομάδα Εργασίας Ψηφιακών Δεδομένων (SWGDE) του National Institute of Justice (NIJ) συνέταξε το έγγραφο " Electronic Crime Scene Investigation, A Guide for First responders» (National Institute of Justice, 2008), όπου περιέγραψε μια διαδικασία τεσσάρων σταδίων για την αντιμετώπιση των ψηφιακών αποδεικτικών στοιχείων. Αυτές οι τέσσερις φάσεις είναι η συλλογή, εξέταση, ανάλυση και παρουσίαση των ψηφιακών αποδείξεων.

FIGURE 1 DIGITAL FORENSIC FOUR STAGE PROCESS

Εικόνα 4: Διαδικασία συλλογής ψηφιακών τεκμηρίων τεσσάρων σταδίων (Shipley, 2007)

Η διαδικασία αυτή αργότερα ενισχύθηκε με το Abstract Digital Forensics Model, το οποίο αύξησε τα στάδια σε εννέα.



Εικόνα 5: Διαδικασία συλλογής ψηφιακών τεκμηρίων εννέα σταδίων (Shipley, 2007)

Κεφάλαιο 3

Σχεδιασμός

3.1 Συνεισφορά της έρευνας.

Όπως αναφέρθηκε στην προηγούμενη ενότητα, πολλοί ερευνητές έχουν διεξάγει μελέτες για την ανεύρεση ψηφιακών τεκμηρίων στο λειτουργικό σύστημα Windows. Συγκεκριμένα έχουν αναλυθεί προηγούμενες εκδόσεις των Windows όπως Vista (Purcell & Lang, 2008) , 7 (Thomas, Sherly & Dija, 2013) και 8 (Stormo, 2013). Επιπλέον έρευνες έχουν γίνει για την μελέτη συγκεκριμένων περιοχών του λειτουργικού όπως το μητρώο (Mee, Tryfonas & Sutherland, 2006; Carvey, 2009; Microsoft, 2016b), η μνήμη (Schuster, 2006; Dolan-Gavitt, 2008; Thomas, Sherly & Dija, 2013; Shanks, 2014), οι συσκευές USB (Carvey & Altheide, 2005; Collie, 2013) και τα αρχεία (Carrier, 2005; Carvey, 2009; Malicious-streams, 2014). Τα Windows 10 είναι ένα νέο λειτουργικό σύστημα που κυκλοφόρησε το καλοκαίρι του 2015 κι ως εκ τούτου δεν έχει αποτελέσει αντικείμενο μελέτης για την συμπεριφορά του στα ψηφιακά τεκμήρια. Για πρώτη φορά στην μελέτη αυτή αναλύεται το λειτουργικό αυτό κι επίσης για πρώτη φορά γίνεται συγκριτική μελέτη για τρία λειτουργικά (Windows 7, 8.1 και 10), ώστε να μελετηθεί αν προκαλείται διαφορετική συμπεριφορά στην εκτέλεση συγκεκριμένων δειγμάτων κακόβουλου κώδικα. Αυτό θα προσφέρει στους αναλυτές ψηφιακών τεκμηρίων μία πολύτιμη βοήθεια, καθώς θα έχουν ένα οδηγό για τις θέσεις στις οποίες αναμένονται να υπάρχουν ψηφιακά τεκμήρια. Με μία σύντομη έρευνα στις συγκεκριμένες θέσεις θα έχουν ένα πρώτο αποτέλεσμα, αν ένα σύστημα έχει προσβληθεί από κακόβουλο λογισμικό ή όχι.

Η ανάλυση κακόβουλου λογισμικού είναι ένα πεδίο με μεγάλο ερευνητικό ενδιαφέρον. Πολλοί ερευνητές κάνοντας στατική ανάλυση (Christodorescu & Jha, 2003; Linn & Debray, 2003) ή δυναμική ανάλυση (Bayer *et al.*, 2006; Egele *et al.*, 2012; Bristow, 2013; Mishra & Pandey, 2014), προσπαθούν να κατανοήσουν την συμπεριφορά του

κακόβουλου κώδικα. Στην παρούσα μεταπτυχιακή διατριβή εντοπίζονται οι πιο κοινές θέσεις στις οποίες ένας αναλυτής ψηφιακών τεκμηρίων πρέπει να εστιάσει την έρευνα του και καταγράφονται οι σημαντικότερες από αυτές τόσο ως προς την κατηγορία του κακόβουλου προγράμματος, όσο και ως προς το λειτουργικό που εκτελείται. Επιπλέον γίνεται παρόμοια ανάλυση με κριτήριο την λειτουργικότητα του κακόβουλου λογισμικού.

3.2 Απαιτήσεις σε λογισμικό

Για την επιτυχή εκτέλεση της πειραματικής διαδικασίες υπάρχουν συγκεκριμένες ανάγκες σε υλικό και λογισμικό όπως παρουσιάζονται στην συνέχεια.

3.2.1 Λειτουργικό σύστημα Ubuntu

Την βάση του εικονικού εργαστηρίου αποτελεί το λειτουργικό σύστημα Ubuntu 14.04. Η επιλογή της συγκεκριμένης έκδοσης έγινε, καθώς αποτελεί μία πλατφόρμα με πολλά διαθέσιμα πακέτα που χρειάστηκαν κατά την εκτέλεση των πειραμάτων. Επιπλέον η συμβατότητα του με το Cuckoo είναι άριστη και έχει χρησιμοποιηθεί και από άλλους ερευνητές (Shanks, 2014) για τον ίδιο σκοπό. Το λειτουργικό αυτό αποτελεί την βάση για την ανάπτυξη της υποδομής, για την διενέργεια των πειραμάτων.

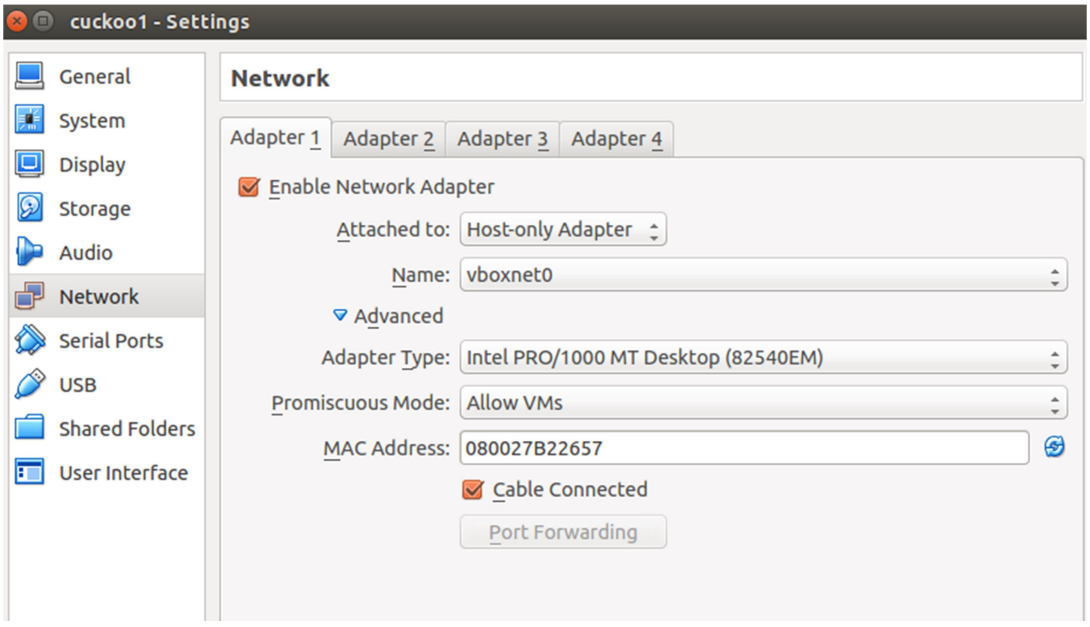
Εγκαταστάθηκε η τελευταία έκδοση του VirtualBox (4.3.34) (Oracle, 2016), το οποίο θα χρησιμοποιηθεί για να τρέξουν οι τρεις εκδόσεις των Windows τις οποίες θα μελετήσουμε. Το VirtualBox είναι ένα εργαλείο virtualization ανοικτού λογισμικού, με δωρεάν διάθεση και υπάρχουν εκδόσεις για Windows, Linux, OS X και Solaris. Δημιουργήθηκε μια νέα εικονική μηχανή πατώντας Νέα κι ακολουθώντας τις οδηγίες. Επιλέχτηκαν ως λειτουργικό τα Microsoft Windows κι η αντίστοιχη έκδοση κάθε φορά (7, 8.1, 10). Στο επόμενο βήμα ορίστηκε το μέγεθος την μνήμης και δημιουργήθηκε ένας νέος εικονικός σκληρός δίσκος τύπου VDI. Μετά από αυτό, η εικονική μηχανή είναι εισηγμένη στα αριστερά.

Για την παραμετροποίηση της μεταβαίνουμε στις ρυθμίσεις, επιλέγουμε δίκτυο και απενεργοποιούμε το "Ενεργοποίηση Δικτύου". Πηγαίνουμε στην καρτέλα System → Accelaration και ενεργοποιούμε το VT-x. Αποθηκεύουμε τις ρυθμίσεις κι εκτελούμε την εικονική μηχανή. Τώρα επιλέγουμε τη μονάδα δίσκου και βάζουμε το δίσκο

εγκατάστασης των Windows. Ακολουθούμε τις οδηγίες και προχωράμε σε μία τυπική εγκατάσταση των Windows.

Όταν εγκατασταθεί και εκκινήσει το λειτουργικό Windows στο εσωτερικό του VirtualBox, κάνουμε κλικ στην επιλογή Devices→Insert Guest Additions CD. Εμείς θα αναφερθούμε στο Windows ενσωματωμένο στο VirtualBox, ως φιλοξενούμενο λειτουργικό (guest). Το σύστημα όπου είναι εγκατεστημένο το VirtualBox, ονομάζεται οικοδεσπότης (host). Οι εικονικές μηχανές όχι μόνο προσφέρουν την ευκαιρία να τρέξει εύκολα ένα λειτουργικό σύστημα μέσα σε ένα άλλο, αλλά επιτρέπουν επίσης να αποθηκεύεται η κατάσταση της μηχανής, η επαναφόρτωση της αλλά και έχουν την δυνατότητα να πηγαίνουμε πίσω σε μια κατάσταση στο παρελθόν. Στην ανάλυση κακόβουλου λογισμικού, αυτό βοηθάει να αποθηκευτεί ένα καθαρό περιβάλλον εργαστηρίου και να μπορεί να πραγματοποιηθεί μετάβαση πίσω σε αυτό, σε περίπτωση που ένα κακόβουλο πρόγραμμα μολύνει το σύστημα.

Τέλος σαν επιλογή για σύνδεση σε δίκτυο επιλέγεται Host-only Adapter και σαν όνομα δικτύου το vboxnet0. Αφού γίνουν αυτές οι επιλογές, ενεργοποιείται η κάρτα δικτύου.



Εικόνα 6: Ρυθμίσεις της κάρτας δικτύου

3.2.2 Cuckoo

Το Cuckoo Sandbox είναι ένα αυτοματοποιημένο περιβάλλον δοκιμών κακόβουλου λογισμικού που χρησιμοποιείται για την εκτέλεση δυναμικής ανάλυσης δειγμάτων. Ένα

απομονωμένο εικονικό μηχάνημα Windows χρησιμοποιείται για την εκτέλεση του δείγματος και την ανάλυση των αποτελεσμάτων. Ως εκ τούτου εκτελέσιμα αρχεία, DLL, διευθύνσεις URL, PDF και αρχεία Microsoft Office, μεταξύ άλλων, μπορούν να αναλυθούν σε αυτό το περιβάλλον. Το Cuckoo έχει πολλά χαρακτηριστικά, όπως η παρακολούθηση οποιονδήποτε αρχείων δημιουργούνται, διαγράφονται ή κατεβάζονται κατά την εκτέλεση, την καταγραφή ενός αντιγράφου της μνήμης από το μηχάνημα και τη σύλληψη οποιαδήποτε δικτυακής κίνησης (Oktavianto & Muhardianto, 2013).

Αρχικά έγινε ενημέρωση του λειτουργικού συστήματος Ubuntu, με τις εντολές sudo apt-get update και sudo apt-get upgrade. Στην συνέχεια εγκαταστάθηκαν τα προαπαιτούμενα που χρειάζεται το Cuckoo καθώς και κάποια χρήσιμα εργαλεία για την ανάλυση του ιομορφικού λογισμικού συγκεκριμένα εγκαταστάθηκαν (Gwallgofi, 2016; Primalsecurity, 2016; Sandbox, 2016; Tech Anarchy, 2016):

- Python 2.7. Η βασική βιβλιοθήκη της Python, με την οποία είναι γραμμένος ο κώδικας του Cuckoo.
- SQLAlchemy. Μία συλλογή με εργαλεία για βάσεις δεδομένων που χρησιμοποιεί η Python.
- Python BSON. Για τον χειρισμό αρχείων Json στην Python.
- Dpkt. Για την εξαγωγή πληροφοριών από αρχεία PCAP.
- Jinja2. Για τον χειρισμό των αναφορών σε μορφή HTML και του web interface.
- Magic. Χρησιμοποιείται για την αυτόματη αναγνώριση του τύπου των αρχείων που είναι υπό ανάλυση.
- Pydeep. Για τον υπολογισμό του ssdeep fuzzy hash των αρχείων. Είναι απαραίτητη η λήψη του από εξωτερική πηγή (Ssdeep, 2016) κι όχι από το αποθετήριο του Ubuntu.
- Pymongo. Για την αποθήκευση των αποτελεσμάτων στην βάση δεδομένων MongoDB.
- YARA και Yara Python. Για το ταίριασμα των υπογραφών των αρχείων, με το αντίστοιχο της Yara (Yara, 2016).
- Bottlepy. Για την χρήση του api.py ή του web.py του Cuckoo.
- Django. Για την εμφάνιση των αποτελεσμάτων σε web interface
- Pefile. Πραγματοποιεί την στατική ανάλυση εκτελέσιμων αρχείων.

- Volatility. Η ενσωμάτωση του Cuckoo με το Volatility (Volatility, no date) είναι προαιρετική. Ωστόσο, η μείωση του χρόνου που απαιτείται για την ανάλυση ενός δείγματος και την αύξηση των πιθανών δεικτών μόλυνσης, αξίζουν τα πρόσθετα βήματα που απαιτούνται για την ενσωμάτωση του.
- Chardet. Για την ανίχνευση κωδικοποίησης συμβολοσειρών.

Στην συνέχεια είναι απαραίτητο να ρυθμιστεί το Cuckoo, ώστε να μπορεί να χρησιμοποιεί το Tcpdump, χωρίς διαχειριστικά δικαιώματα. Αυτό γίνεται με την εντολή sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump. Για την επαλήθευση με την εντολή getcap /usr/sbin/tcpdump περιμένουμε ως αποτέλεσμα να έχουμε /usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip.

Κατόπιν έγινε η εγκατάσταση της τελευταίας έκδοσης του Cuckoo (2.0) κι η αρχική παραμετροποίηση του προγράμματος. Δημιουργήθηκε ένας χρήστης cuckoo για να εκτελεί την ανάλυση, με την εντολή sudo adduser cuckoo. Για να έχει αυτός ο χρήστης δικαιώματα στο Virtualbox πρέπει να ανήκει στην ομάδα vboxusers (sudo usermod -a -G vboxusers cuckoo).

Το Cuckoo χρησιμοποιεί και στηρίζεται σε έξι αρχεία ρυθμίσεων. Παρακάτω φαίνεται καθένα από αυτά, καθώς κι η επιπλέον ρύθμιση η οποία έγινε:

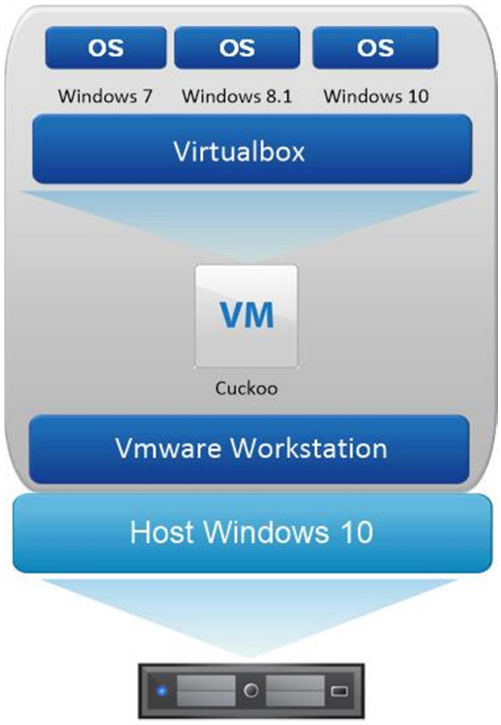
1. Cuckoo.conf. Για τη διαμόρφωση γενικών επιλογών συμπεριφοράς και ανάλυσης. Ορίστηκε η επιλογή memory_dump = on ώστε να γίνεται αντίγραφο της μνήμης, καθώς και machinery = virtualbox αφού χρησιμοποιήθηκε το virtualbox σαν λογισμικό virtualization.
2. Auxiliary.conf. Για την ενεργοποίηση και τη ρύθμιση βοηθητικών μονάδων.
3. Virtualbox.conf. Για τον καθορισμό των επιλογών για το λογισμικό virtualization. Εδώ επιλέχθηκε mode = gui ώστε κατά την διάρκεια εκτέλεσης της ανάλυσης να φαίνεται η επιφάνεια εργασίας του υπό ανάλυση λειτουργικού, label = cuckoo1 το οποίο είναι το όνομα του υπό εξέταση λειτουργικού, ip = 192.168.56.101 είναι η διεύθυνση του και snapshot = Snapshot1 το όνομα της αποθηκευμένης κατάστασης του λειτουργικού.
4. Memory.conf. Το αρχείο ρυθμίσεων memory.conf χρησιμοποιείται για να ρυθμιστεί ποιες μεταβλητές του Volatility πρόκειται να χρησιμοποιηθούν.
5. Processing.conf. Για την ενεργοποίηση και ρύθμιση μονάδων επεξεργασίας.

6. Reporting.conf. Για την ενεργοποίηση ή απενεργοποίηση αναφορών. Εδώ ενεργοποιήθηκε (enabled=yes) η εξαγωγή της αναφοράς σε Html μορφή.

Για κάθε ένα από τα τρία λειτουργικά προστέθηκαν τα VirtualBox Guest Additions. Στην συνέχεια έγινε εγκατάσταση της Python 2.7 (Python, 2016) καθώς και του Pillow Python Imaging Library (Pillow, 2016). Αφού έγινε η κατάλληλη ρύθμιση σε επίπεδο δικτύου για να μπορεί να επικοινωνεί το Cuckoo με το VM που αναλύεται, έγινε απενεργοποίηση των Windows Update, του Windows Firewall και του Windows Defender. Έγινε η εγκατάσταση του agent του Cuckoo έτσι ώστε να ξεκινάει αυτόματα με τα Windows. Η αποθήκευση της τρέχουσας κατάστασης της μηχανής ονομάζεται «Snapshot». Μόλις έχουν τελειώσει οι εγκαταστάσεις των παραπάνω λογισμικών μέσα στο εικονικό μηχάνημα με Windows, γίνεται μία αποθήκευση της κατάστασης σε ένα Snapshot με το όνομα snapshot1.

3.3 Απαιτήσεις σε υλικό

Για την υλοποίηση του εικονικού εργαστηρίου, είναι απαραίτητη η χρήση ενός υπολογιστικού συστήματος με αυξημένες δυνατότητες, ώστε να δημιουργηθούν οι απαραίτητες εικονικές μηχανές. Η αρχιτεκτονική που επιλέχθηκε, αποτυπώνεται στην Εικόνα 7.



Εικόνα 7: Η αρχιτεκτονική του εικονικού εργαστηρίου.

Ένα φυσικό μηχάνημα, με τα εξής χαρακτηριστικά:

- Επεξεργαστής: Intel Core i7-4790
- Μνήμη 16 GB
- Λειτουργικό Windows 10 Pro

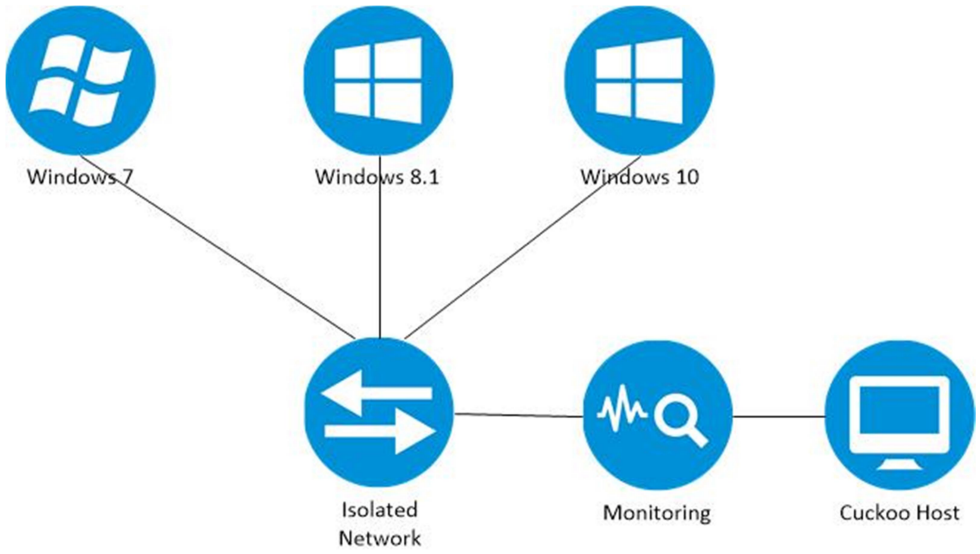
διαμορφώθηκε κατάλληλα για να δημιουργηθεί το εικονικό εργαστήριο. Συγκεκριμένα με την χρήση του Vmware Workstation Pro 12 (VMware, 2016) δημιουργήθηκε εικονικό μηχάνημα με τα εξής χαρακτηριστικά:

- Επεξεργαστής: τέσσερις πυρήνες από τον Intel Core i7-4790
- Μνήμη 8 GB
- Λειτουργικό Ubuntu 14.04

Το συγκεκριμένο μηχάνημα, εγκαταστάθηκε το Cuckoo. Στην συνέχεια, μέσα στο Cuckoo, δημιουργήθηκαν τρία εικονικά μηχανήματα με τα εξής χαρακτηριστικά:

- Επεξεργαστής: ένας πυρήνας από τον Intel Core i7-4790
- Μνήμη 2 GB
- Λειτουργικό Windows

Σε κάθε ένα από αυτά τα τρία εικονικά μηχανήματα εγκαταστάθηκε διαφορετική έκδοση των Windows και συγκεκριμένα 7, 8.1 και 10. Για να επικοινωνεί το Cuckoo με το κάθε λειτουργικό που είναι υπο δοκιμή, η μία κάρτα δικτύου του Cuckoo και η μοναδική της εικονικής μηχανής Windows, συνδέονται σε ένα εικονικό απομονομένο δίκτυο (192.168.56.0/24) όπως φαίνεται στην εικόνα 2.



Εικόνα 8: Διαμόρφωση δικτύου εικονικού εργαστηρίου.

3.4 Δείγματα κακόβουλου λογισμικού

Συχνά, οι αναλυτές κακόβουλου λογισμικού έχουν να αντιμετωπίσουν νέες απειλές και άγνωστα εκτελέσιμα. Αλλά υπάρχουν σενάρια όπου μπορεί κανείς να χειριστεί κακόβουλο λογισμικό που γνωρίζει ήδη το όνομά του κι έχει ταξινομηθεί, π.χ. για λόγους έρευνας όπως σε αυτή την εργασία. Για να αναλυθεί ένα τέτοιο κακόβουλο λογισμικό, υπάρχουν πολλά μέρη όπου μπορεί κάποιος να συλλέξει γνωστά δείγματα. Ο Lenny Zeltser, ο οποίος είναι ο επικεφαλής της ιδιωτικής SANS Institute (SANS, 2016), συνιστά αρκετές ελεύθερες πηγές στην ιστοσελίδα του (L. Zeltser, 2016). Τα δείγματα κακόβουλου κώδικα που χρησιμοποιούνται στην παρούσα μεταπτυχιακή διατριβή λήφθηκαν από το Malware.lu (Malware.lu, 2016), Virussign (VirusSign, 2016), Vx Heaven (VxHeaven, 2016), Malekal (Malekal, 2016) και το MalwareTips (MalwareTips, 2016). Για τη λήψη κάποιου δείγματος απαιτείται συνήθως εγγραφή του χρήστη. Στην

συνέχεια, αφού εγκριθεί ο λογαριασμός του, έχει πρόσβαση στα δείγματα. Τα δείγματα συμπιέζονται και ο κωδικός πρόσβασης συνήθως είναι infected.

Επιλέχθηκαν εξήντα δείγματα κακόβουλου κώδικα, είκοσι (20) Trojan, είκοσι (20) Worm και είκοσι (20) Bot. Επιλέχθηκαν αυτές οι τρεις οικογένειες κακόβουλου λογισμικού, καθώς αποτελούν τις σημαντικότερες κατηγορίες που ανιχνεύονται. Παρακάτω υπάρχει μία συνοπτική περιγραφή των τριών αυτών κατηγοριών.

Στην επιστήμη των υπολογιστών, ο δούρειος ίππος (trojan horse ή απλά trojan) είναι ένα κακόβουλο πρόγραμμα που ξεγελάει τον χρήστη και τον κάνει να πιστεύει ότι εκτελεί κάποια χρήσιμη λειτουργία, ενώ στα κρυφά εγκαθιστά στον υπολογιστή του άλλα κακόβουλα προγράμματα (Wikipedia, 2016). Η τακτική που χρησιμοποιούν οι δούρειοι ίπποι είναι παρόμοια με την τακτική που χρησιμοποίησε ο Οδυσσέας, οπότε πήραν κι αυτήν την ονομασία. Συγκεκριμένα, κρύβουν μέσα τους κακόβουλο κώδικα ο οποίος μπορεί να μολύνει τον υπολογιστή. Εξωτερικά μοιάζουν με προγράμματα τα οποία εκτελούν χρήσιμες λειτουργίες, είναι ενδιαφέροντα και δίνουν την εντύπωση στον χρήστη ότι είναι ακίνδυνα. Όταν όμως ο χρήστης εκτελέσει αυτό το πρόγραμμα, τότε ενεργοποιείται ο κακόβουλος κώδικας με αποτέλεσμα ο υπολογιστής να μολυνθεί. Συνήθως αποτέλεσμα της μόλυνσης από δούρειο ίππο είναι η εγκατάσταση κάποιου προγράμματος που επιτρέπει σε μη εξουσιοδοτημένους χρήστες να έχουν πρόσβαση στον μολυσμένο υπολογιστή και να τον χρησιμοποιούν για να ξεκινήσουν άλλες επιθέσεις προς άλλους υπολογιστές του διαδικτύου. Σε αντίθεση με τους ιούς και τα worms, οι δούρειοι ίπποι δεν αναπαράγονται μολύνοντας άλλα αρχεία (Panda Security, 2007; Cisco, 2016).

Τα worms υπολογιστών είναι παρόμοια με τους ιούς στην λειτουργία τους καθώς αναπαράγουν αντίγραφα του εαυτού τους και μπορούν να προκαλέσουν τον ίδιο τύπο βλάβης. Σε αντίθεση με τους ιούς, οι οποίοι απαιτούν την εξάπλωση ενός μολυσμένου αρχείου υποδοχής, τα worms είναι αυτόνομο λογισμικό και δεν απαιτούν ένα πρόγραμμα ξενιστή ή ανθρώπινη βοήθεια για να εξαπλωθούν. Για να εξαπλωθούν είτε εκμεταλλεύονται μια ευπάθεια στο σύστημα στόχο είτε χρησιμοποιηθούν κάποιο είδος κοινωνικής μηχανικής για να ξεγελάσουν τους χρήστες για να τα εκτελέσουν. Ένα worm εισέρχεται σε ένα υπολογιστή μέσω ενός τρωτού σημείου του συστήματος και εκμεταλλεύεται τα χαρακτηριστικά μεταφοράς αρχείων ή πληροφοριών (Cisco, 2016; Gehringer, 2016).

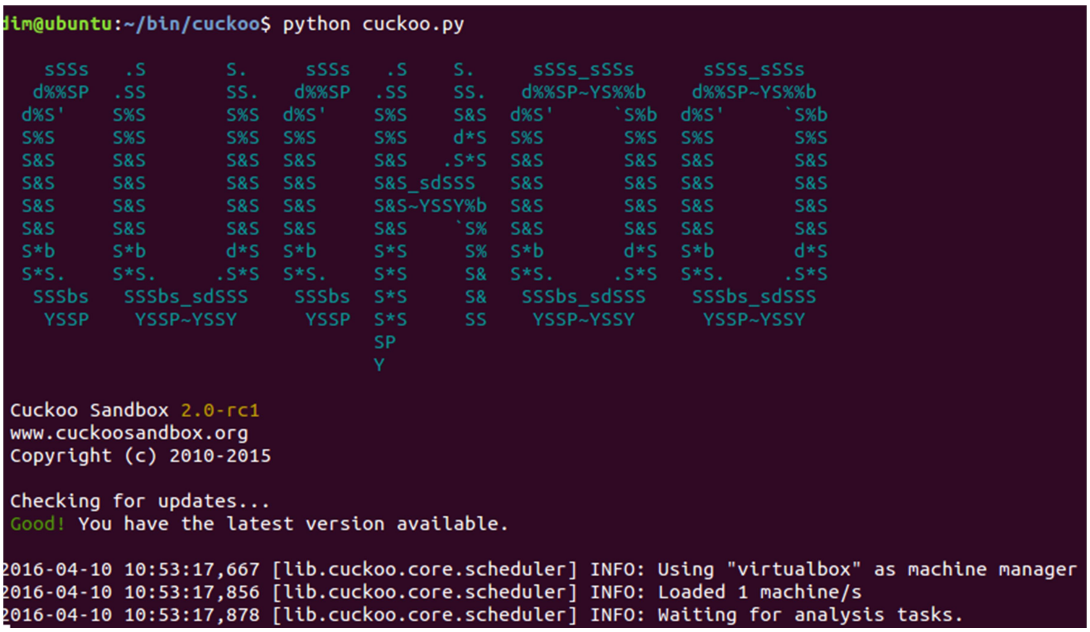
Ένα bot είναι ένα κακόβουλο λογισμικό ελεγχόμενο από έναν εισβολέα γνωστό ως botmaster. Ένα botnet είναι ένα δίκτυο μολυσμένων υπολογιστών με bot. Η φύση της αρχιτεκτονικής botnet παρέχει σε αυτό μια τεράστια ισχύ επίθεσης με ταυτόχρονη κινητοποίηση των μελών του στρατού bot για να επιτεθούν σε έναν στόχο. Επιπλέον, το botnet είναι ο ελβετικός σουγιάς του επιτιθέμενου λόγω της ευχρηστίας του. Μια επίθεση botnet μπορεί να διαιωνιστεί με κατανεμημένη άρνηση υπηρεσίας (DDoS), κλοπής διαπιστευτηρίων, εγκατάσταση ad-ware, μαζικό spamming κλπ. Τα botnets χρησιμοποιούνται ευρέως όχι μόνο για την απόκτηση οικονομικών οφελών, αλλά και για να αποκτηθούν στρατιωτικά ή πολιτικά πλεονεκτήματα από ιδιώτες, οργανισμούς, ακόμη και από κυβερνήσεις (Bächer *et al.*, 2005). Τα Botnets έχουν γίνει μια από τα πιο σημαντικές απειλές του Διαδικτύου την τελευταία δεκαετία (Cooke, Jahanian & McPherson, 2005). Δεδομένου ότι το Διαδίκτυο έχει γίνει μια ζώνη σύγκρουσης στο οποίο οι εγκληματίες του κυβερνοχώρου κι οι υπερασπιστές του Διαδικτύου συμμετέχουν σε ένα ατέλειωτο πόλεμο. Σε αυτή την σύγκριση, τα botnets αναδύονται ως το κορυφαίο όργανο με την οποία οι επιθέσεις στον κυβερνοχώρο διαπράττονται (Micro & Paper, 2006; Ianelli & Hackworth, 2007).

Στο παράρτημα Α παρατίθεται το κακόβουλο λογισμικό που αναλύεται στην εργασία αυτή.

3.5 Πειραματική διαδικασία

Όπως αναφέρθηκε προηγουμένως, το Cuckoo επεξεργάζεται δείγματα malware κι αποθηκεύει τα αποτελέσματα της ανάλυσης σε ένα φάκελο. Για κάθε αίτημα ανάλυσης, δημιουργεί ένα ξεχωριστό υποφάκελο που περιέχει όλες τις εκθέσεις που παράγονται, τα ακατέργαστα αρχεία καταγραφής, .pcap αρχεία, εικόνες καθώς και κάθε άλλη πληροφορία που λαμβάνεται κατά τη διάρκεια της ανάλυσης. Χρησιμοποιώντας το Cuckoo ως το κύριο εργαλείο ανάλυσης κακόβουλο λογισμικού, κάθε δείγμα θα μελετηθεί σε τρία (3) διαφορετικά περιβάλλοντα λογισμικού (Windows 7, 8.1 και 10) και τα αποτελέσματα της ανάλυσης θα καταγράφονται σε κατάλληλη μορφή για περαιτέρω μελέτη.

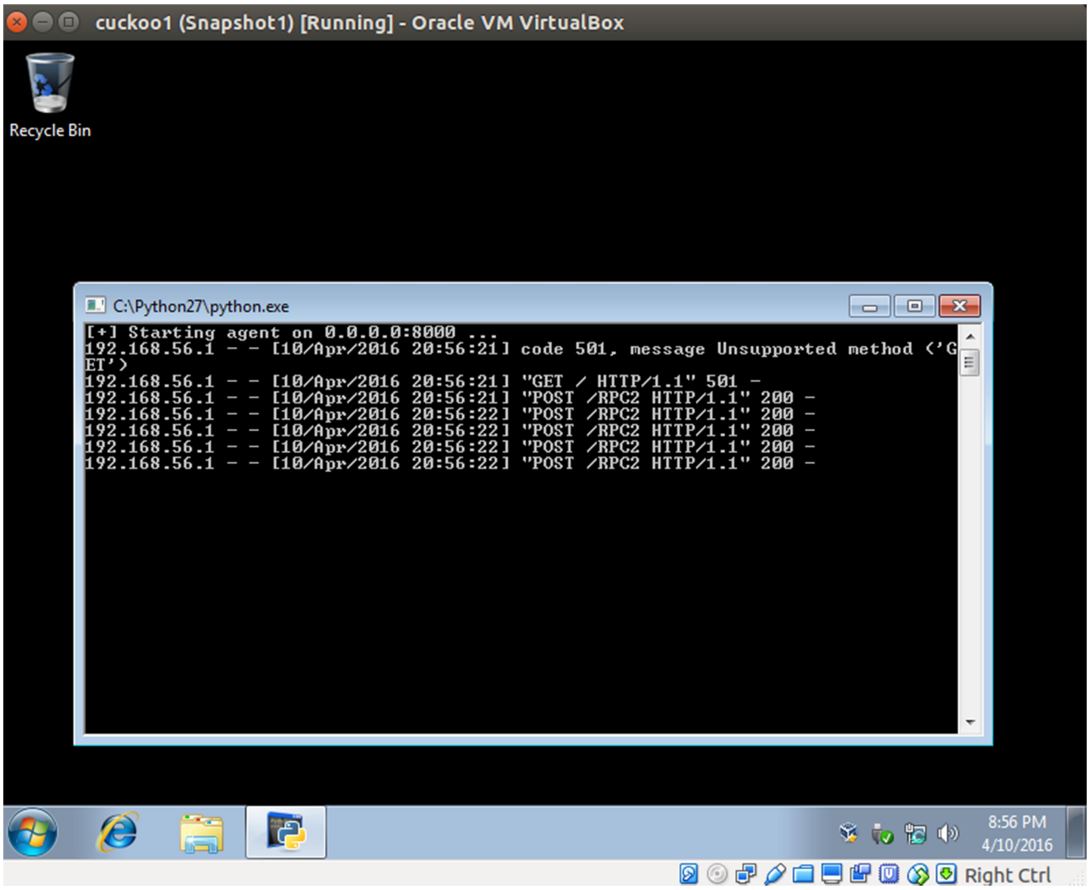
Το βασικό αρχείο για την εκτέλεση του Cuckoo, είναι το αρχείο cuckoo.py. Με την εκτέλεση του python cuckoo.py ξεκινάει το πρόγραμμα και μπαίνει σε αναμονή για την εκτέλεση ανάλυσης. Στο επόμενο βήμα πρέπει να στείλουμε στο Cuckoo το δείγμα που θέλουμε να ελεγχθεί με την εντολή sudo python submit.py /home/user/Desktop/sample.exe



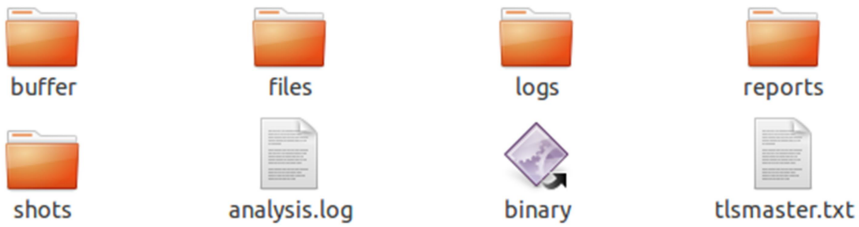
Εικόνα 9: Η εκτέλεση του Cuckoo

Το πρόγραμμα αναλαμβάνει πλέον να στείλει το δείγμα στο εικονικό μηχάνημα που έχουμε επιλέξει από το αρχείο ρυθμίσεων. Όταν γίνει η έκχυση του δείγματος στο λειτουργικό, παρακολουθείται όλη η δραστηριότητα του συστήματος και

καταγράφεται. Μόλις τελειώσει η ανάλυση το εικονικό μηχάνημα τερματίζεται και δημιουργείται το αρχείο .html, με την αναφορά της ανάλυσης.



Εικόνα 10: Η έναρξη του υπό εξέταση λειτουργικού



Εικόνα 11: Τα αρχεία και οι φάκελοι που δημιουργούνται στην αναφορά της ανάλυσης

cuckoo

Info | File | Signatures | Screenshots | Static | Dropped | Network | Behavior | Volatility

Category	Started On	Completed On	Duration	Cuckoo Version
FILE	2016-04-05 02:39:59	2016-04-05 02:40:31	32 seconds	2.0-rc1

Machine	Label	Manager	Started On	Shutdown On
cuckoo1	cuckoo1	VirtualBox	2016-04-05 02:40:02	2016-04-05 02:40:31

File Details

File name	officeclicktorun.exe
File size	1883320 bytes
File type	PE32 executable (GUI) Intel 80386, for MS Windows
CRC32	66A1C91D
MD5	79b418fbecf6f61badf6d1e025a98d6a
SHA1	70209ca74b26bd06ac5607821c80be2b74b27448
SHA256	23e82595d06d85f426703e454532dea9f0a1a5062ebbb276518bd82ba35bc3c3
SHA512	a388041f68efbf06048c0fc5e4f4d98cdd0a09adf0de8720f6a7964151990deb138f3a3cd9d4c3a00c5e16ca2981dd0e7c94192e5d3e956ecc4c1a1b13f57d3a
Ssdeep	24576:hN5IVCJglnKVmpBVj3RRQkDTZ5eiFdSFeer/8aokc+s59P9hk7ooizcOd1:hN5IfnyWv1vt5d1aoXJcOd1

Εικόνα 12: Πληροφορίες στην αναφορά για το αρχείο που αναλύθηκε.

Static Analysis

Sections

Name	Virtual Address	Virtual Size	Size of Raw Data	Entropy
.text	0x00001000	0x00006960	0x00007000	5.63396810324
.data\x00\x00.	0x00008000	0x00000a98	0x00001000	0.0
.rsrc\x00\xdd_	0x00009000	0x0001a138	0x0001b000	7.71301240678

Resources

Name	Offset	Size	Language	Sub-language	File type
RT_ICON	0x000094ec	0x000010a8	LANG_NEUTRAL	SUBLANG_NEUTRAL	data
RT_ICON	0x000094ec	0x000010a8	LANG_NEUTRAL	SUBLANG_NEUTRAL	data
RT_ICON	0x000094ec	0x000010a8	LANG_NEUTRAL	SUBLANG_NEUTRAL	data
RT_GROUP_ICON	0x000094bc	0x00000030	LANG_NEUTRAL	SUBLANG_NEUTRAL	MS Windows icon resource - 3 icons, 32x32, 16 colors
RT_VERSION	0x00009150	0x0000036c	LANG_ENGLISH	SUBLANG_ENGLISH_US	data

Imports

Library uSeR32.Dll:
• 0x401000 - CallWindowProcW

Εικόνα 13: Πληροφορίες στην αναφορά για τα αποτελέσματα του δείγματος στο Virustotal.

VirusTotal

Permalink
VirusTotal Scan Date: 2016-02-17 06:01:45
Detection Rate: 42/54 (Collapse)

Antivirus	Version	Result
Ad-Aware	3.0.2.1015	Gen:Variant.Symmi.52
AegisLab	4.2	Troj.W32.Jorik.Nrgbot.pur!c
Agnitum	5.5.1.3	Worm.Ngrbot!myUrhyoW8jc
AhnLab-V3	2016.02.17.00	Backdoor/Win32.Ruskill
Alibaba	1.0	Clean
ALYac	1.0.1.9	Gen:Variant.Symmi.52
Antiy-AVL	1.0.0.1	Trojan/Win32.VBKrypt
Arcabit	1.0.0.653	Trojan.Symmi.52
Avast	8.0.1489.320	Win32:Agent-ARQZ [Trj]
AVG	16.0.0.4530	Generic33.CDBN
Baidu-International	3.5.1.41473	Adware.Win32.Agent.Elnx
BitDefender	7.2	Gen:Variant.Symmi.52
Bkav	1.3.0.7400	Clean
ByteHero	1.0.0.1	Clean
CAT-QuickHeal	14.00	Worm.Dorkbot.r3
ClamAV	0.98.5.0	Clean
CMC	1.1.0.977	Clean
Comodo	24235	UnclassifiedMalware
Cyren	5.4.16.7	W32/Trojan.EQZQ-9356
DrWeb	7.0.17.11230	Trojan.Siggen5.35247
Emsisoft	3.5.0.642	Gen:Variant.Symmi.52 (B)
ESET-NOD32	13040	Win32/Dorkbot.B

Εικόνα 14: Πληροφορίες στην αναφορά για την στατική ανάλυση.

Behavior Summary

File-Read
- C:\Windows\Fonts\staticcache.dat

File-Opened
- C:\Windows\System32\dwmapi.dll
- C:\Windows\System32\en-US\uSeR32.Dll.mui
- C:\Windows\System32\uxtheme.dll
- C:\Windows\Fonts\staticcache.dat

Registry Key-Read
- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\CTF\EnableAnchorContext
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane13
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane12
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane11
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane14
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane9
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\DataStore_V1.0\Disable
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane10
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableMetaFiles
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQMClient\Windows\CEIPEnable
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane16
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane8
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane7
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane6
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\DataStore_V1.0\DataFilePath
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane4
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane5
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane2
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane3
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack\SurrogateFallback\Plane15

Εικόνα 15: Πληροφορίες στην αναφορά για την δυναμική ανάλυση.

Κεφάλαιο 4

Ανάλυση πειραματικών αποτελεσμάτων

4.1 Ψηφιακά τεκμήρια με βάση την λειτουργικότητα

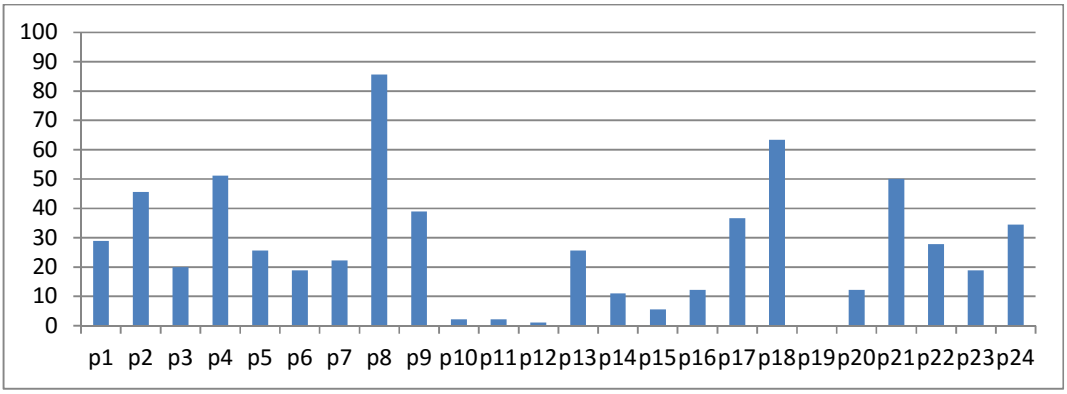
Η πρώτη ανάλυση που πραγματοποιήθηκε αφορούσε το ερευνητικό ερώτημα, αν η συχνότητα εμφάνισης ψηφιακών τεκμηρίων σε μία συγκεκριμένη θέση, επηρεάζεται από την λειτουργικότητα του κακόβουλου λογισμικού. Για το σκοπό αυτό έγινε ανάλυση των εξήντα (60) δειγμάτων του παραρτήματος Α και βρέθηκε αν διαθέτουν τέσσερεις μορφές λειτουργικότητας:

- Αν κατασκοπεύουν ή/και κλέβουν στοιχεία του χρήστη (Spy-Steal data)
- Αν επικοινωνούν με ένα κέντρο έλεγχου για να λαμβάνουν εντολές (Command and Control)
- Αν παρέχουν απομακρυσμένη πρόσβαση στο σύστημα (Backdoor)
- Αν χρησιμοποιούν μηχανισμούς απόκρυψης και παραπλάνησης (Stealth)

Στην συνέχεια πραγματοποιήθηκε συγκριτική στατιστική ανάλυση για τις τέσσερεις λειτουργικότητες στα σύνολο των εξήντα δειγμάτων ανεξάρτητα του λειτουργικού συστήματος. Παρακάτω καταγράφονται οι σημαντικότερες θέσεις ανά λειτουργικότητα όπου παρατηρείται συχνότητα ανίχνευσης σε ποσοστό άνω του 30%.

4.1.1 Δείγματα κακόβουλου λογισμικού Spy-Steal data

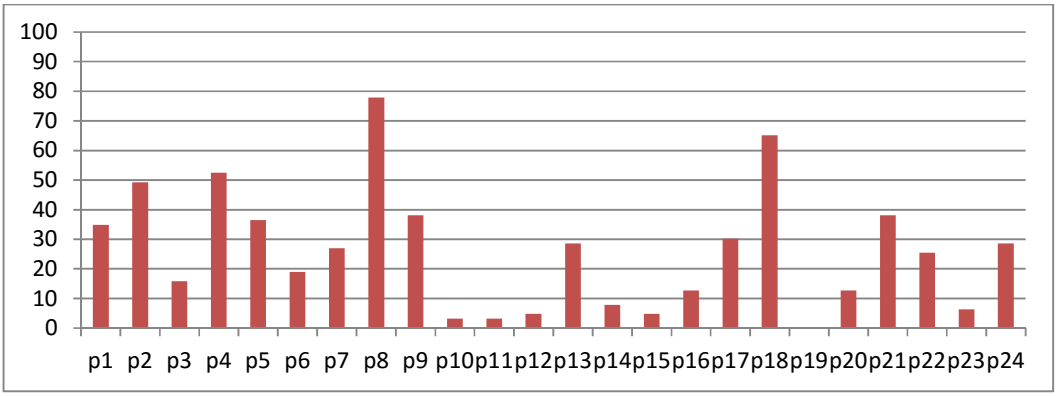
Οι θέσεις με φθίνουσα σειρά ανίχνευσης ψηφιακών τεκμηρίων για τη λειτουργικότητα Spy-Steal data του κακόβουλου λογισμικού είναι οι εξής: p8 (85,6%), p18 (63,3%), p4 (51,1%), p21 (50%), p2 (45,6%), p9 (38,9%), p17 (36,7%) και p24 (34,4%).



Διάγραμμα 1: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στο σύνολο των δειγμάτων κακόβουλου λογισμικού Spy-Steal data

4.1.2 Δείγματα κακόβουλου λογισμικού Command and Control

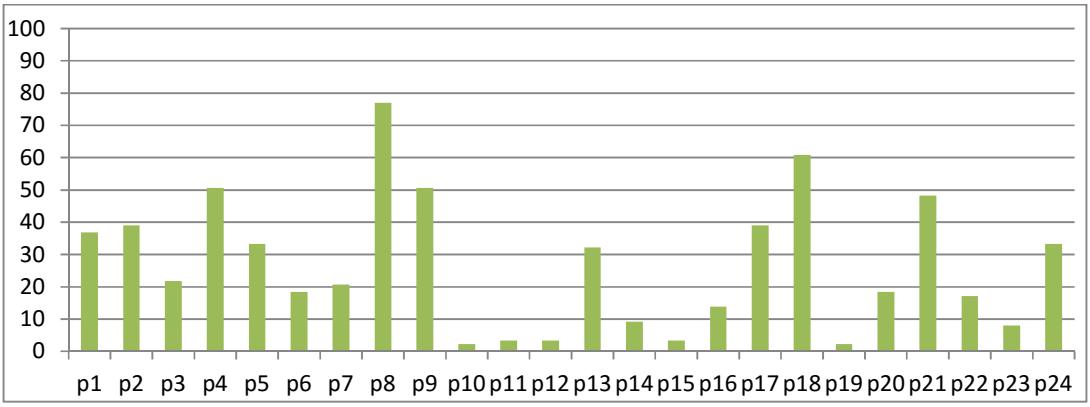
Οι θέσεις με φθίνουσα σειρά ανίχνευσης ψηφιακών τεκμηρίων για τη λειτουργικότητα Command and Control του κακόβουλου λογισμικού είναι οι εξής: p8 (77,8%), p18 (65,1%), p4 (52,4%), p2 (49,2%), p21 (38,1%), p9 (38,1%), p5 (36,5%), p1 (34,9%) και p17 (30,2%).



Διάγραμμα 2: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στο σύνολο των δειγμάτων κακόβουλου λογισμικού Command and Control

4.1.3 Δείγματα κακόβουλου λογισμικού Backdoor

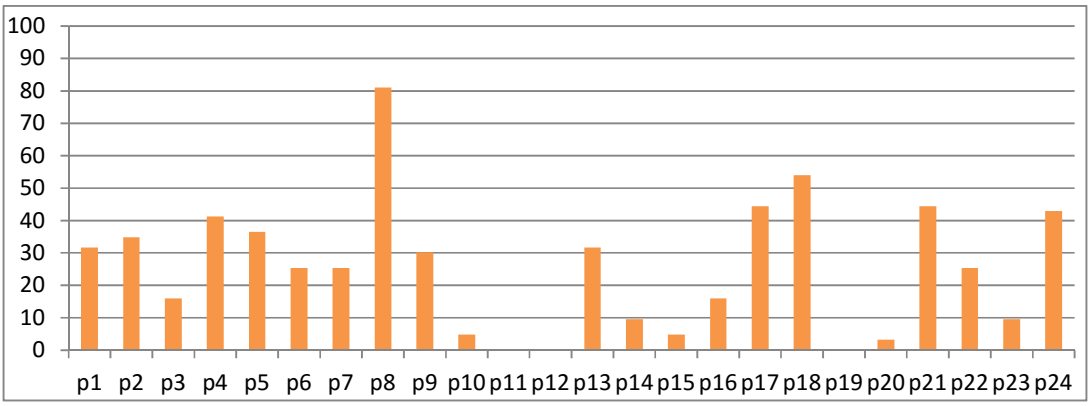
Οι θέσεις με φθίνουσα σειρά ανίχνευσης ψηφιακών τεκμηρίων για τη λειτουργικότητα Backdoor του κακόβουλου λογισμικού είναι οι εξής: p8 (77%), p18 (60,9%), p4 (50,6%), p9 (50,6%), p21 (48,3%), p2 (39,1%), p17 (39,1%), p1 (36,8%), p5 (33,3%), p24 (33,3%) και p13 (32,2%).



Διάγραμμα 3: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στο σύνολο των δειγμάτων κακόβουλου λογισμικού Backdoor

4.1.4 Δείγματα κακόβουλου λογισμικού Stealth

Οι θέσεις με φθίνουσα σειρά ανίχνευσης ψηφιακών τεκμηρίων για τη λειτουργικότητα Stealth του κακόβουλου λογισμικού είναι οι εξής: p8 (81%), p18 (54%), p21 (44,4%), p17 (44,4%), p24 (42,9%), p4 (41,3%), p5 (36,5%), p2 (34,9%), p1 (31,7%), p13 (31,7%) και p9 (30,2%)



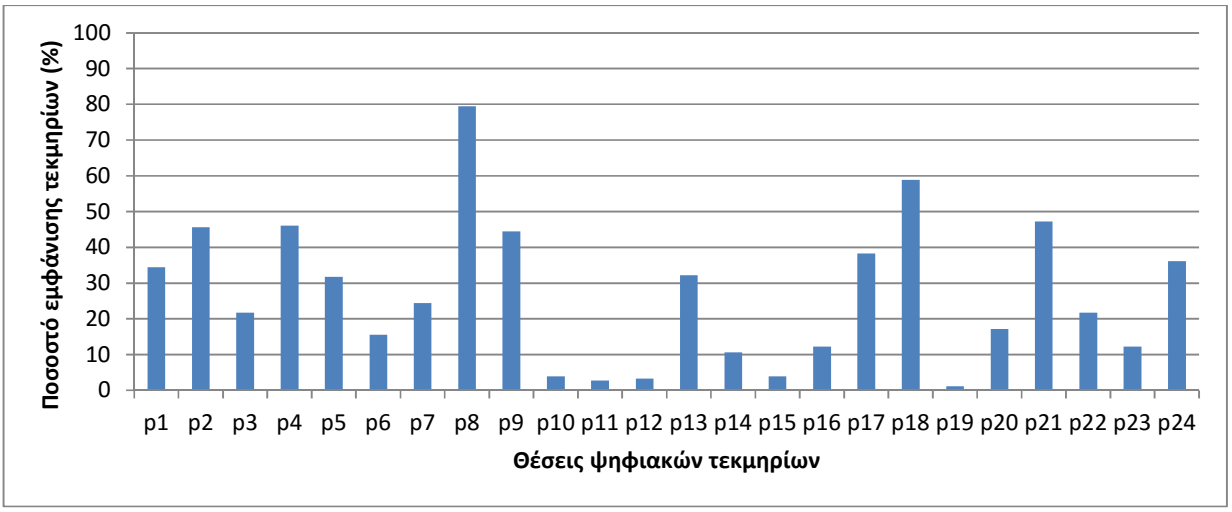
Διάγραμμα 4: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στο σύνολο των δειγμάτων κακόβουλου λογισμικού Stealth

4.2 Ψηφιακά τεκμήρια με βάση το λειτουργικό ή το είδος

Η δεύτερη ανάλυση που πραγματοποιήθηκε αφορούσε το ερευνητικό ερώτημα, αν η συχνότητα εμφάνισης ψηφιακών τεκμηρίων στο σύνολο των θέσεων που εξετάστηκαν, επηρεάζεται από το είδος του κακόβουλου λογισμικού ή από το λειτουργικό σύστημα. Για το σκοπό αυτό πραγματοποιήθηκε συγκριτική στατιστική ανάλυση για τα τρία είδη κακόβουλου λογισμικού και για τα τρία λειτουργικά συστήματα με βάση τα αποτελέσματα στο παράρτημα Β.

4.2.1 Ψηφιακά τεκμήρια στο σύνολο των δειγμάτων

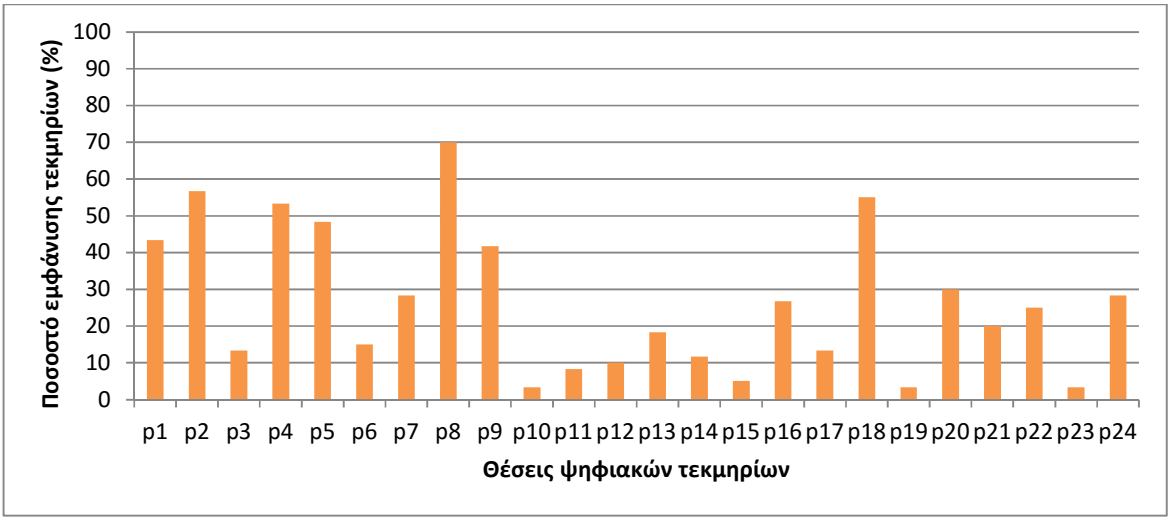
Με βάση το Διάγραμμα 5 παρατηρείται ότι ανεξαρτήτως τύπου κακόβουλου λογισμικού οι πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων είναι με φθίνουσα σειρά οι p8 (79,4%), p18 (58,9%), p21 (47,25), p4 (46,1%) και p2 (45,6%). Ενώ με οι θέσεις με χαμηλότερη συχνότητα είναι με αύξουσα σειρά p19 (1,1%), p11 (2,85), p12 (3,3%) , p10(3,9%) και p15(3,9%).



Διάγραμμα 5: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση ανεξαρτήτως τύπου κακόβουλου λογισμικού και λειτουργικών συστημάτων.

4.2.2 Ψηφιακά τεκμήρια στα δείγματα Trojan

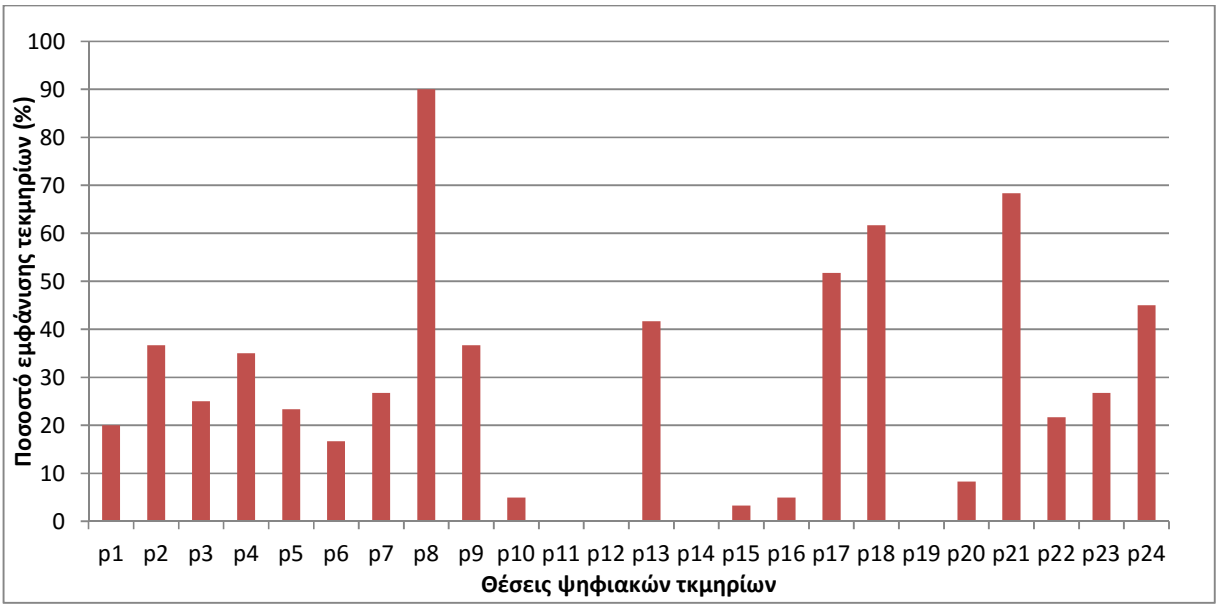
Με βάση το Διάγραμμα 6 παρατηρείται ότι για τα δείγματα των Trojan στις πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων περιλαμβάνονται ομοίως οι p8 (70%), p18 (55%), p4 (53,3%)και p2 (56,7%) κι επιπλέον η θέση p5 (48,3%). Ενώ με οι θέσεις με χαμηλότερη συχνότητα είναι επίσης οι p19 (3.3%), p11 (8.3%), p10(3,3%) και p15(5%), καθώς και η p23 (3,3%).



Διάγραμμα 6: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα δείγματα των Trojan, στο σύνολο των λειτουργικών συστημάτων.

4.2.3 Ψηφιακά τεκμήρια στα δείγματα Worm

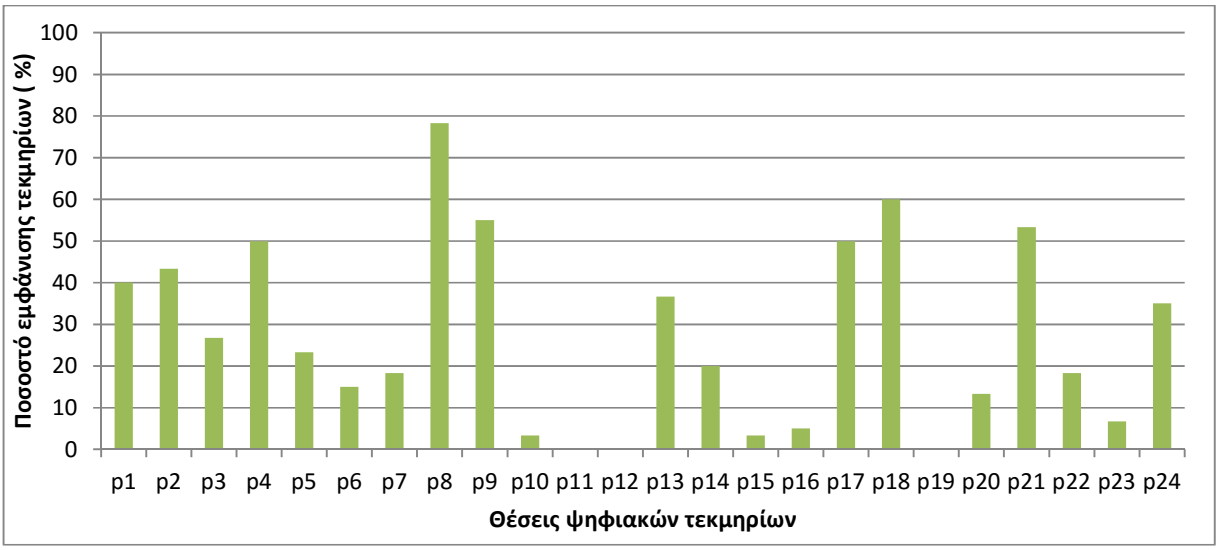
Με βάση το Διάγραμμα 7 παρατηρείται ότι για τα δείγματα των Worm στις πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων σε σχέση με το σύνολο περιλαμβάνονται ομοίως οι p8 (90%), p18 (61,7%), p21 (68,3%). Ενώ οι θέσεις p17 (51,7%) και p24 (45%) ανιχνεύονται ως σημαντικές σε αυτόν τον τύπο κακόβουλου λογισμικού. Στις θέσεις p19, p11, p12 και p14 δεν ανιχνεύονται ψηφιακά τεκμήρια και στη θέση p15 (3,3%) το ποσοστό ανίχνευσης είναι πολύ χαμηλό.



Διάγραμμα 7: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα δείγματα των Worm, στο σύνολο των λειτουργικών συστημάτων

4.2.4 Ψηφιακά τεκμήρια στα δείγματα Bot

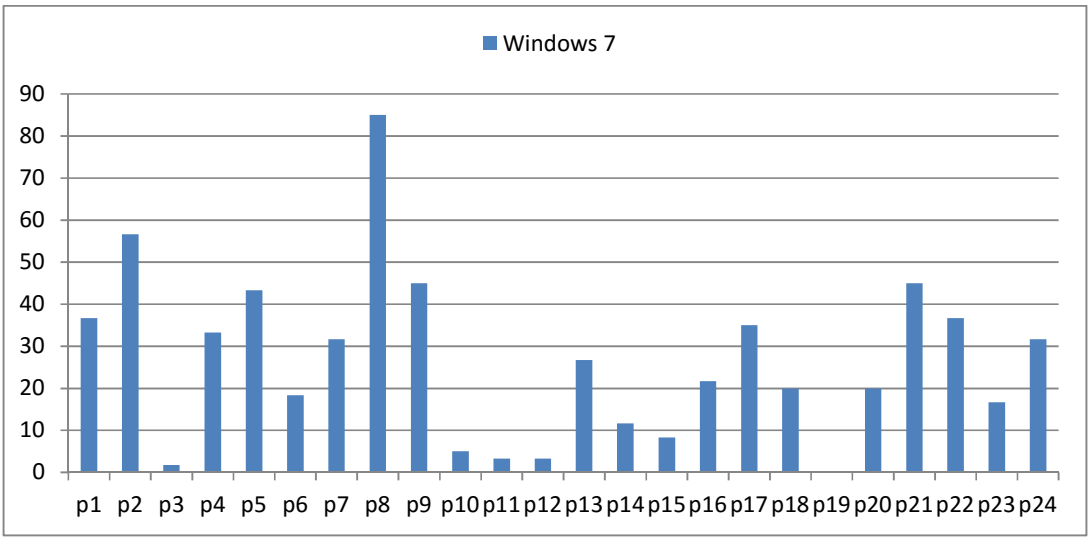
Με βάση το Διάγραμμα 8 παρατηρείται ότι για τα δείγματα των Bot στις πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων σε σχέση με το σύνολο περιλαμβάνονται ομοίως οι p8 (78.3%), p18 (60%), p21 (53.3%). Ενώ οι θέσεις p9 (55%) και p17 (50%) ανιχνεύονται ως σημαντικές σε αυτόν τον τύπο κακόβουλου λογισμικού. Συνεπώς η θέση p17 είναι σημαντική για τα Worm και Bot αλλά όχι για τα Trojan. Στις θέσεις p19, p11, p12 δεν ανιχνεύονται ψηφιακά τεκμήρια και στις θέσεις p15 (3,3%) και p10 (3,3%) το ποσοστό ανίχνευσης είναι πολύ χαμηλό.



Διάγραμμα 8: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα δείγματα των Bot, στο σύνολο των λειτουργικών συστημάτων

4.2.5 Ψηφιακά τεκμήρια στα Windows 7

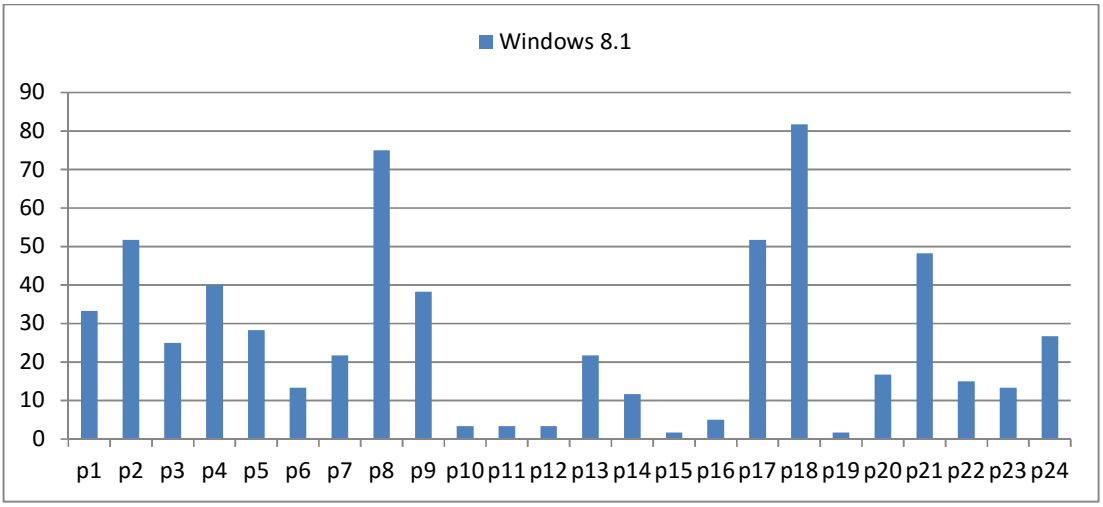
Με βάση το Διάγραμμα 9παρατηρείται ότι για τα Windows 7 οι πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων σε σχέση με το σύνολο περιλαμβάνονται οι p8 (85%), p2 (56,7%), p21 (45%), p9 (45%) και p5 (43,3%) Ενώ με οι θέσεις με χαμηλότερη συχνότητα είναι με αύξουσα σειρά p19 (0%), p3 (1,7), p11(3,3%) , p12(3,3%) και p10(5%).



Διάγραμμα 9: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα Windows 7, στο σύνολο των δειγμάτων κακόβουλου λογισμικού

4.2.6 Ψηφιακά τεκμήρια στα Windows 8.1

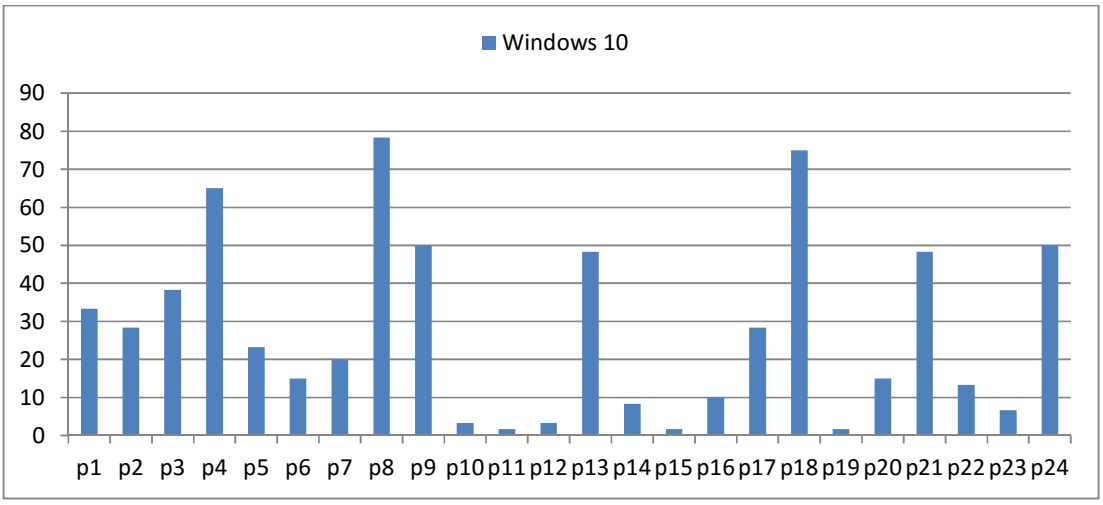
Με βάση το Διάγραμμα 10 παρατηρείται ότι για τα Windows 8.1 οι πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων σε σχέση με το σύνολο περιλαμβάνονται ομοίως οι p18 (81,7%), p8 (75%), p2 (51,7%), p17 (51,7%) και p21 (48,3%) Ενώ με οι θέσεις με χαμηλότερη συχνότητα είναι με αύξουσα σειρά p19 (1,7%), p15 (1,7), p11(3,3%) , p12(3,3%) και p10(3,3%).



Διάγραμμα 10: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα Windows 8.1, στο σύνολο των δειγμάτων κακόβουλου λογισμικού

4.2.7 Ψηφιακά τεκμήρια στα Windows 10

Με βάση το Διάγραμμα 11 παρατηρείται ότι για τα Windows 10 οι πέντε πιο σημαντικές θέσεις με την υψηλότερη συχνότητα εύρεσης ψηφιακών τεκμηρίων σε σχέση με το σύνολο περιλαμβάνονται ομοίως οι p8 (78,3%), p18 (75%), p4 (65%), p9 (50%) και p24 (50%) Ενώ με οι θέσεις με χαμηλότερη συχνότητα είναι με αύξουσα σειρά p19 (1,7%), p15 (1,7), p11(1,7%) , p12(3,3%) και p10(3,3%).



Διάγραμμα 11: Ποσοστό (%) εμφάνισης ψηφιακών τεκμηρίων ανά θέση στα Windows 10, στο σύνολο των δειγμάτων κακόβουλου λογισμικού

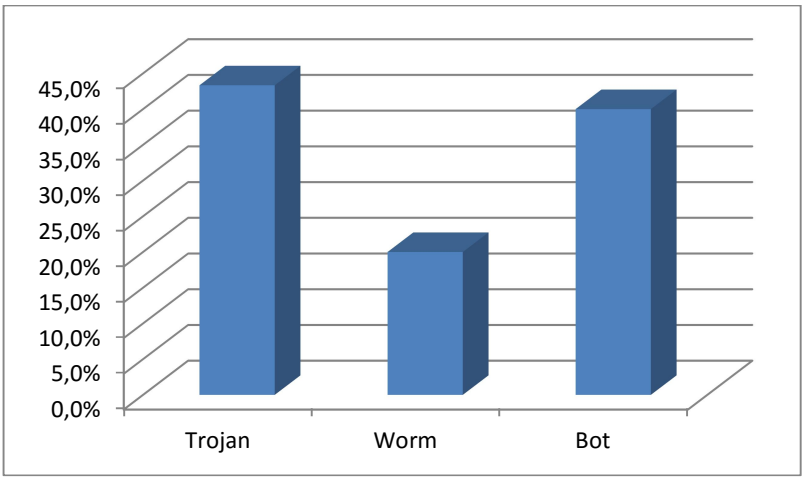
4.3 Ψηφιακά τεκμήρια ανά θέση

Η τρίτη ανάλυση που πραγματοποιήθηκε αφορούσε το ερευνητικό ερώτημα, αν η συχνότητα εμφάνισης ψηφιακών τεκμηρίων σε μία συγκεκριμένη θέση, επηρεάζεται από το είδος του κακόβουλου λογισμικού ή από το λειτουργικό σύστημα. Για το σκοπό αυτό πραγματοποιήθηκε συγκριτική στατιστική ανάλυση για τα τρία είδη κακόβουλου λογισμικού και για τα τρία λειτουργικά συστήματα με βάση τα αποτελέσματα στο παράρτημα Β.

4.3.1 Θέση 1

(HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CustomLocale\en-US)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (43,3%), Worm (20%) και Bot (40%). Συνεπώς, τα Worms εμφανίζονται με στατιστικώς μικρότερη συχνότητα σε σχέση με τα Trojan (P<0.05).



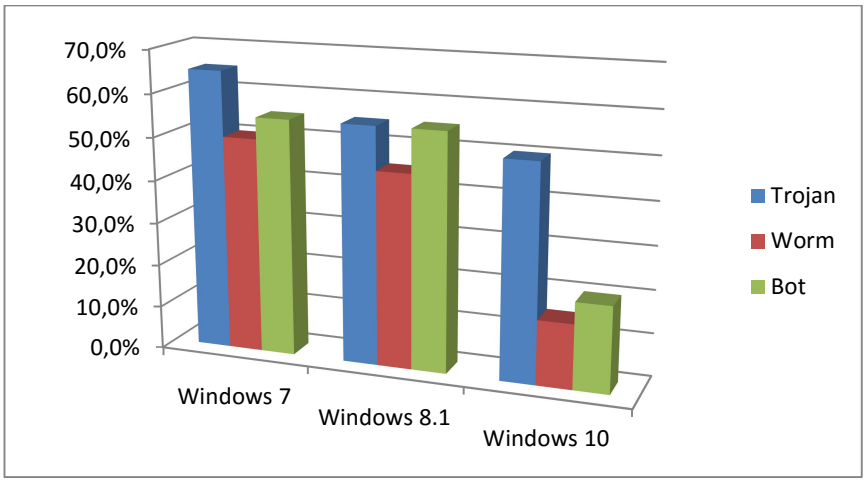
Διάγραμμα 12. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 1, ανεξαρτήτως λειτουργικού συστήματος.

4.3.2 Θέση 2

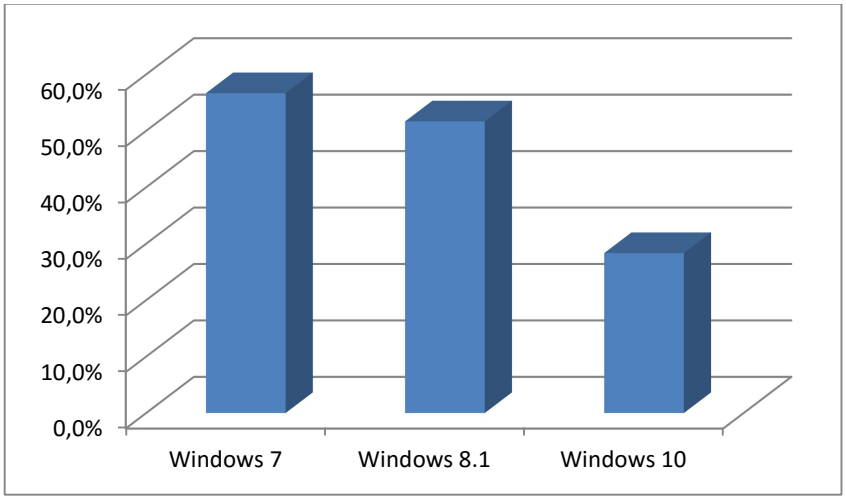
(HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls)

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (56,7%), Windows 8.1 (51,7%) και Windows 10 (28,3%). Συνεπώς, στα Windows 10, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (65%), Worm (50%) και Bot (55%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (55%), Worm (45%) και Bot (55%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (50%), Worm (15%) και Bot (20%). Συνεπώς, στα Windows 10 και στα είδη Worm και Bot, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).



Διάγραμμα 13. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 2



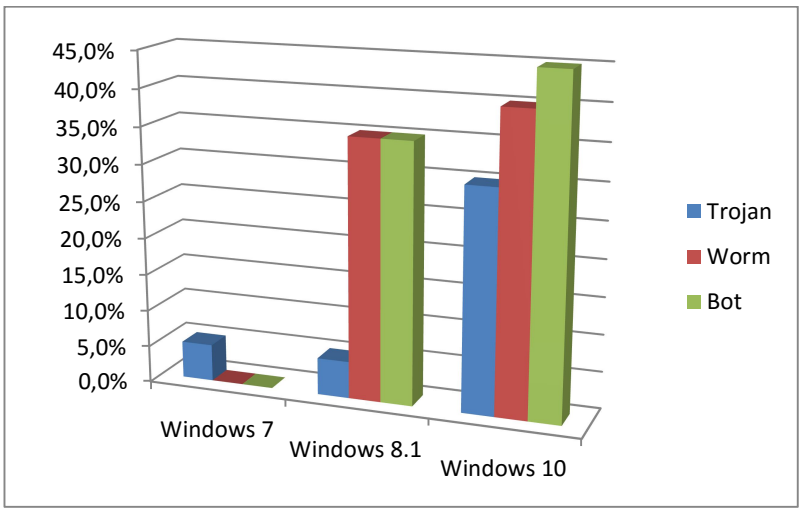
Διάγραμμα 14. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 2, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.3 Θέση 3

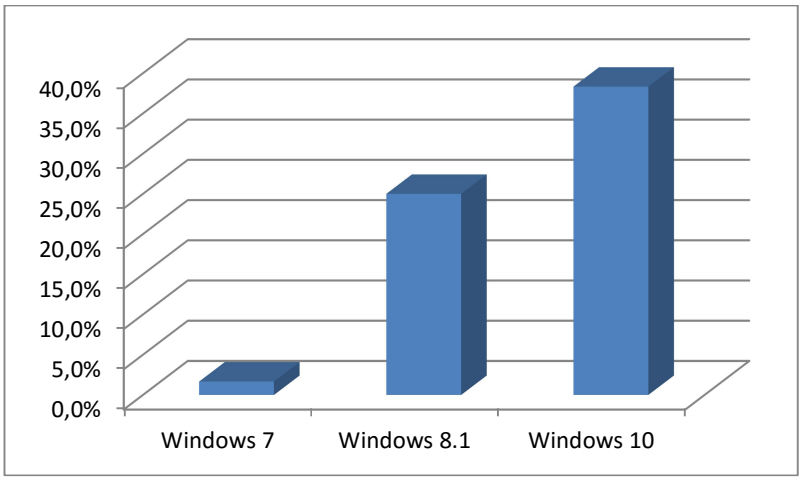
(HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SESSION)

Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (5%), Worm (0%) και Bot (0%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (5%), Worm (35%) και Bot (35%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (30%), Worm (40%) και Bot (45%). Συνεπώς, στα Windows 7 και στα είδη Worm και Bot, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (1,7%), Windows 8.1 (25%) και Windows 10 (38,3%). Συνεπώς, στα Windows 7, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).



Διάγραμμα 15. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 3

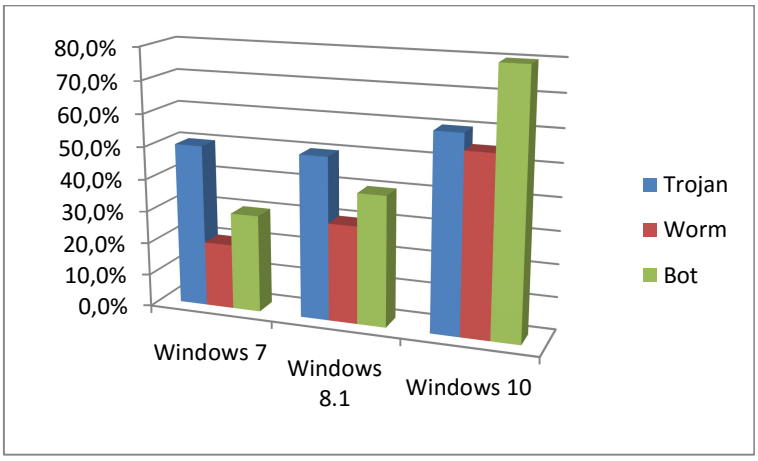


Διάγραμμα 16. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 3, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

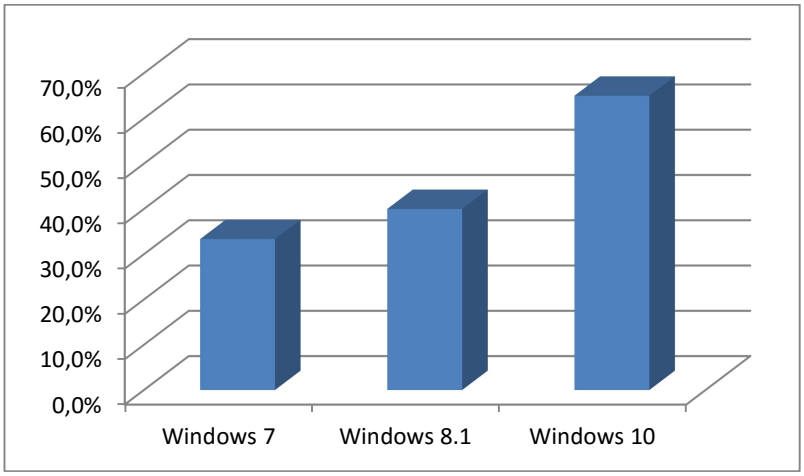
4.3.4 Θέση 4

(HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control)Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (50%), Worm (20%) και Bot (30%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (50%), Worm (30%) και Bot (40%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (60%), Worm (55%) και Bot (80%). Συνεπώς, στα Windows 10 και στο είδος Bot, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (33,3%), Windows 8.1 (40%) και Windows 10 (65%). Συνεπώς, στα Windows 10, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).



Διάγραμμα 17. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 4

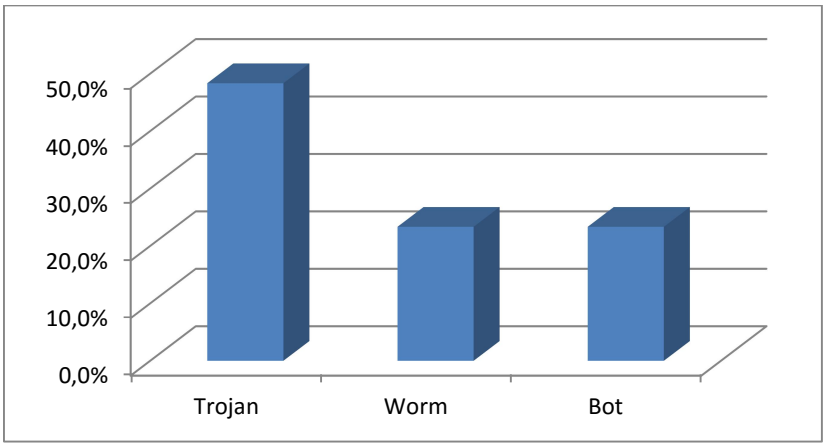


Διάγραμμα 18. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 4, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.5 Θέση 5

(HKEY_LOCAL_MACHINE\SYSTEM)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (48,3%), Worm (23,3%) και Bot (23,3%). Συνεπώς τα Trojan εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worm και Bot ($P<0.05$).

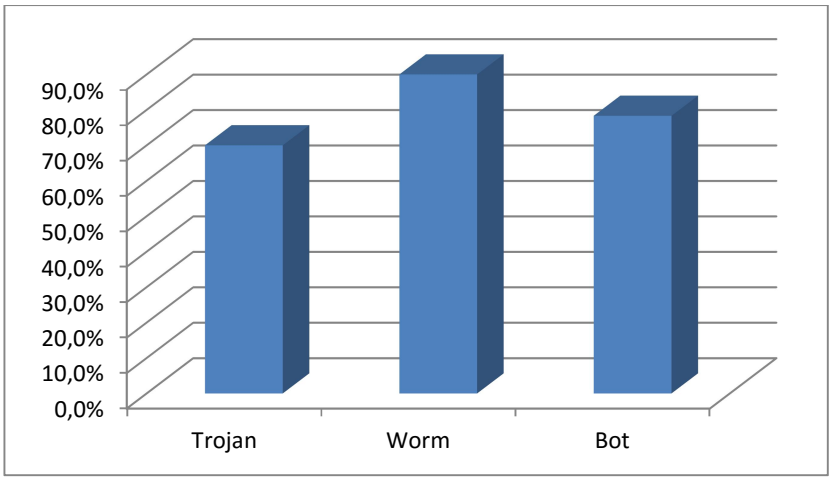


Διάγραμμα 19. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 5, ανεξαρτήτως λειτουργικού συστήματος.

4.3.6 Θέση 8

(HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (70%), Worm (90%) και Bot (78,3%). Συνεπώς τα Worms εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Trojan (P<0.05).

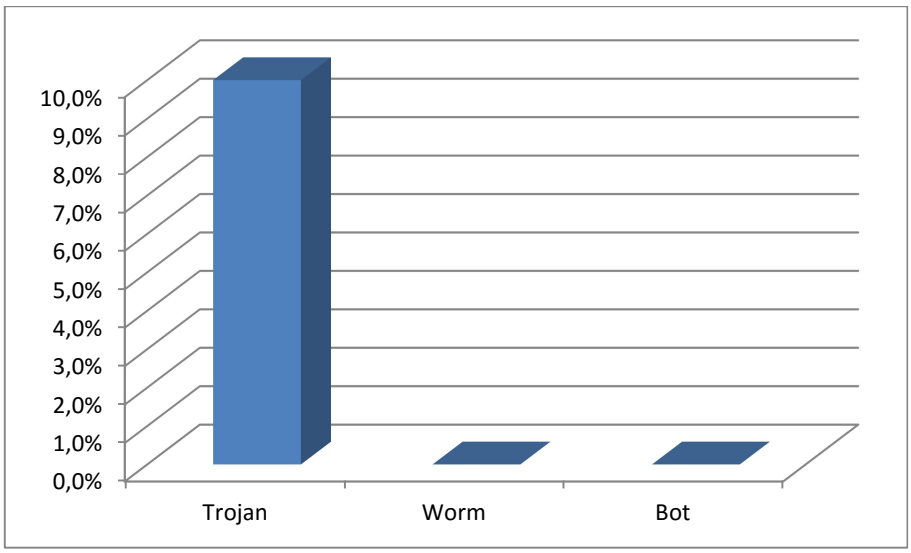


Διάγραμμα 20. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 8, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.7 Θέση 12

(HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (10%), Worm (0%) και Bot (0%). Συνεπώς αυτή τα Trojan εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worm και Bot (P<0.05).



Διάγραμμα 21. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 8, ανεξαρτήτως λειτουργικού συστήματος.

4.3.8 Θέση 13

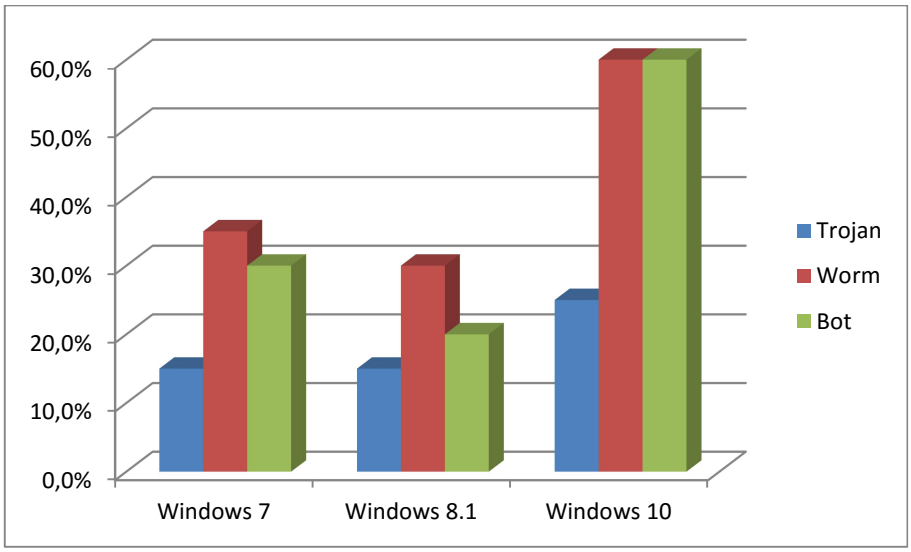
(HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\)

Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (15%), Worm (35%) και Bot (30%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (15%), Worm (30%) και Bot (20%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (25%), Worm (60%) και Bot (60%). Συνεπώς, στα Windows 10 και στο είδος Bot, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με Windows 8,1 (P<0.05).

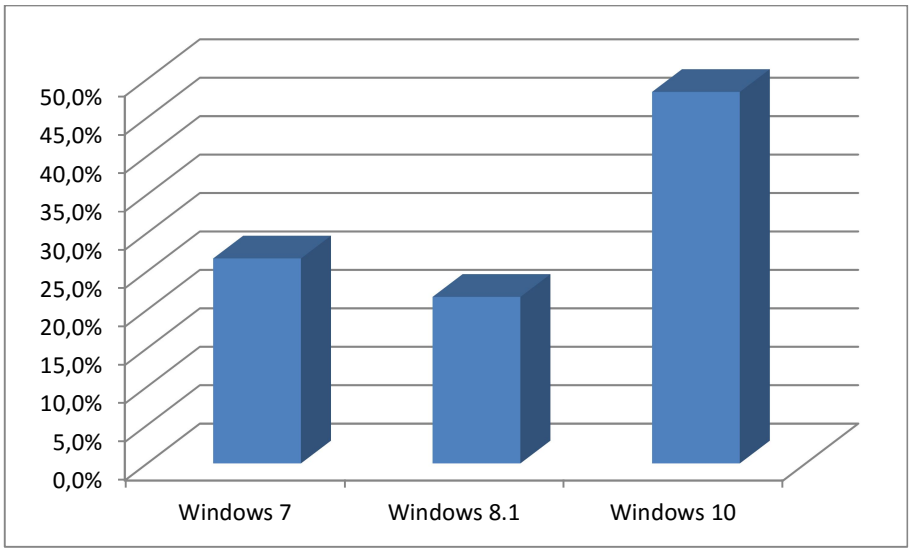
Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (26,7%), Windows 8.1 (21,7%) και Windows 10 (48,3%). Συνεπώς, στα

Windows 10, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

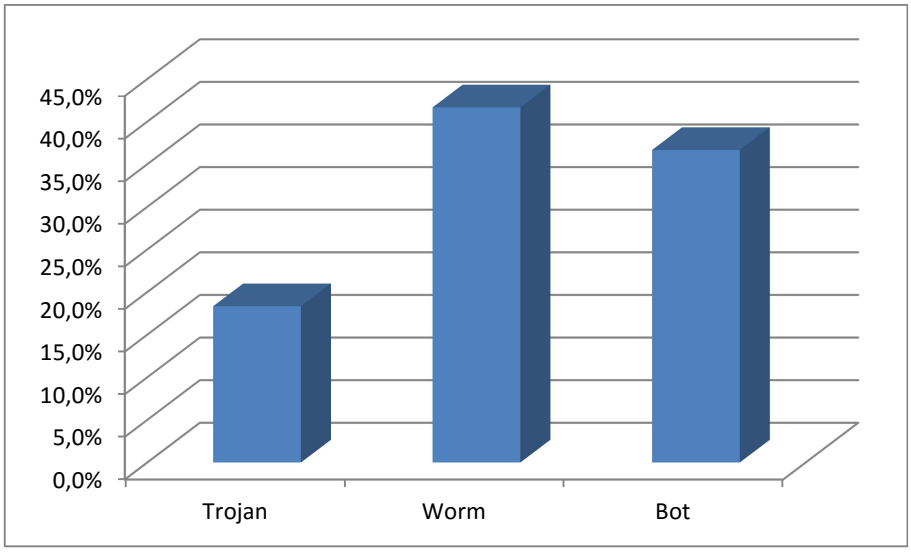
Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (18,3%), Worm (41,7%) και Bot (36,7%). Συνεπώς τα Trojan εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worms (P<0.05).



Διάγραμμα 22. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 13



Διάγραμμα 23. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 13, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

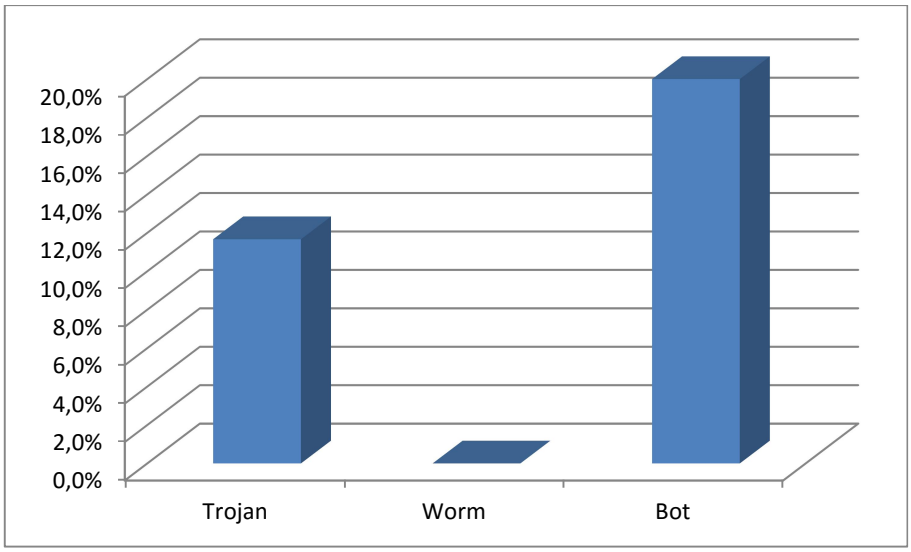


Διάγραμμα 24. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 13, ανεξαρτήτως λειτουργικού συστήματος.

4.3.9 Θέση 14

(HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (11,7%), Worm (0%) και Bot (20%). Συνεπώς τα Trojan και Bot εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worm (P<0.05).



Διάγραμμα 25. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 14, ανεξαρτήτως λειτουργικού συστήματος.

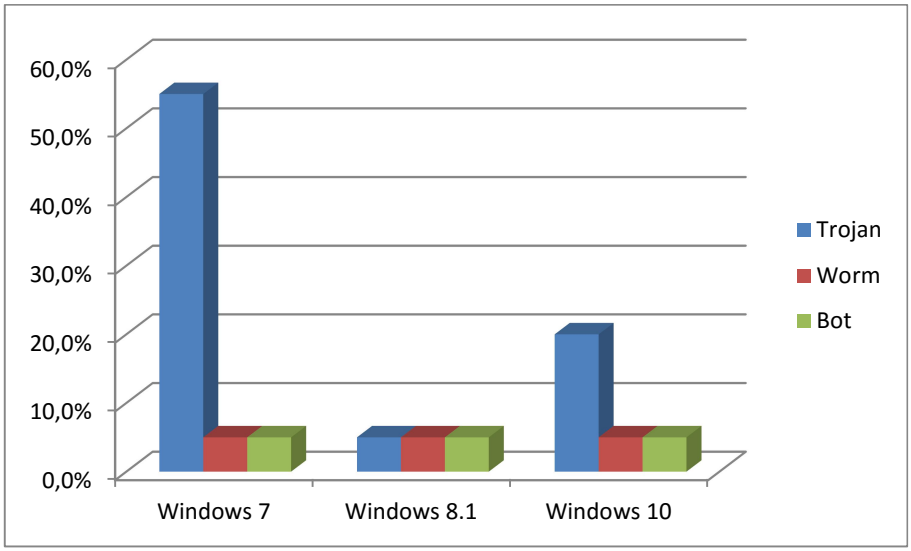
4.3.10 Θέση 16

(%systemdrive%\Documents and Settings\ [User Name]\Local Settings\Temp)

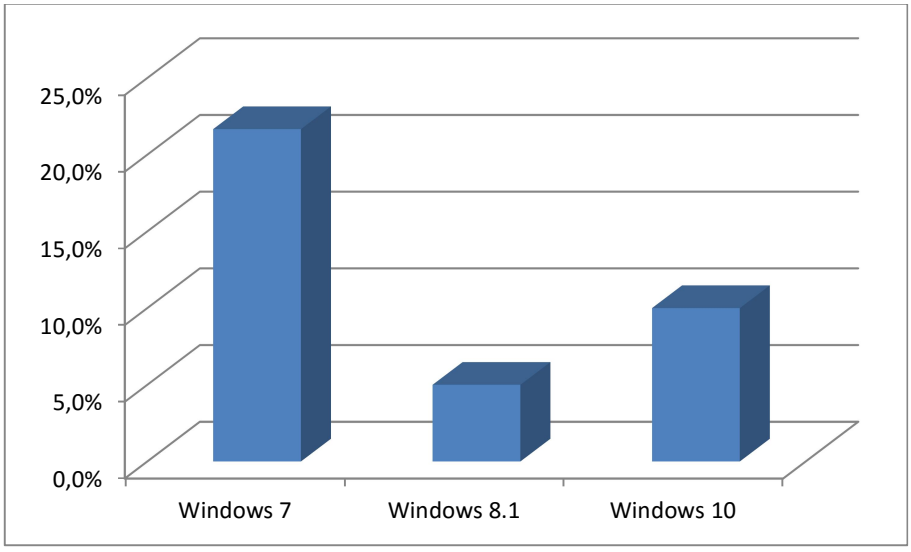
Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (55%), Worm (5%) και Bot (5%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (5%), Worm (5%) και Bot (5%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (20%), Worm (5%) και Bot (5%). Συνεπώς, στα Windows 7 και στο είδος Trojan, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με Windows 8,1. (P<0.05).

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (21,7%), Windows 8.1 (5%) και Windows 10 (10%). Συνεπώς, στα Windows 7, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με Windows 8,1 (P<0.05).

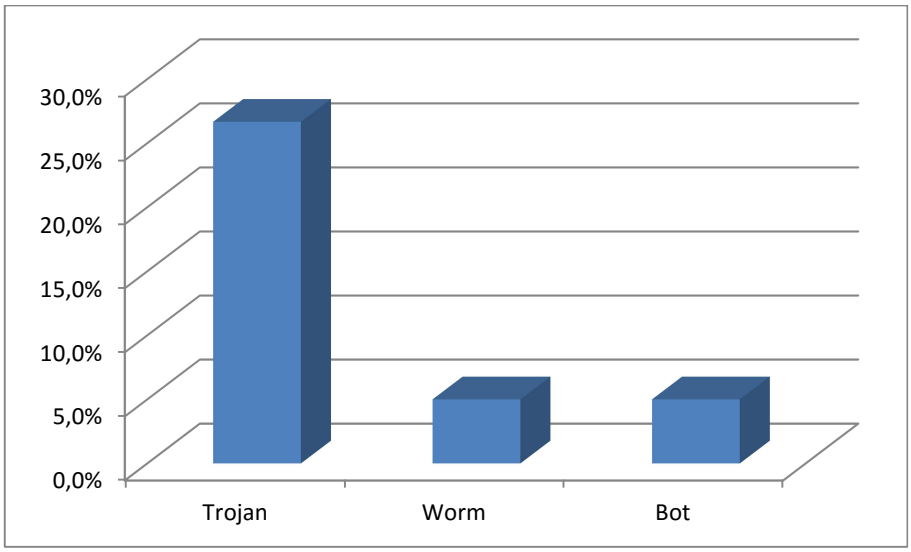
Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (26,7%), Worm (5%) και Bot (5%). Συνεπώς αυτή τα Trojan εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worm και Bot(P<0.05).



Διάγραμμα 26. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 16



Διάγραμμα 27. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 16, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού



Διάγραμμα 28 Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 16, ανεξαρτήτως λειτουργικού συστήματος.

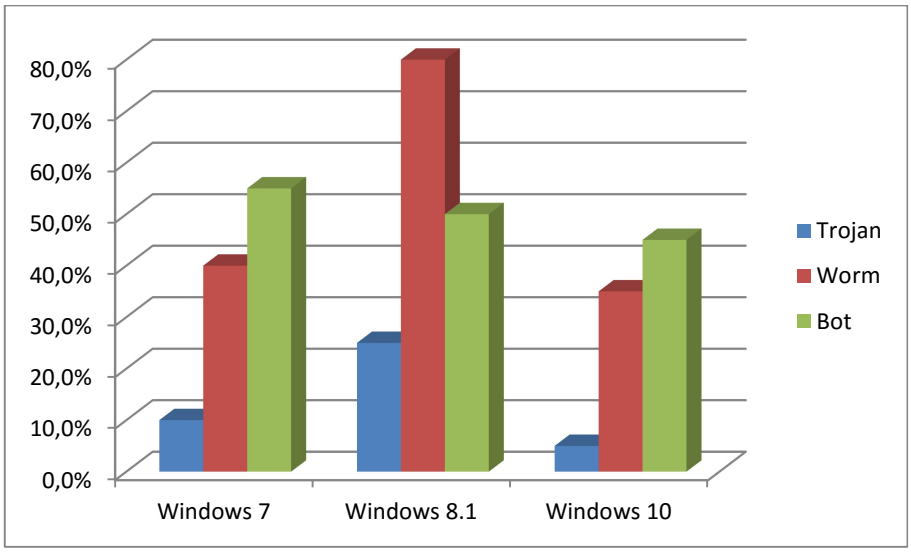
4.3.11 Θέση 17

(%Systemdrive%\Users\victim_user\AppData\)

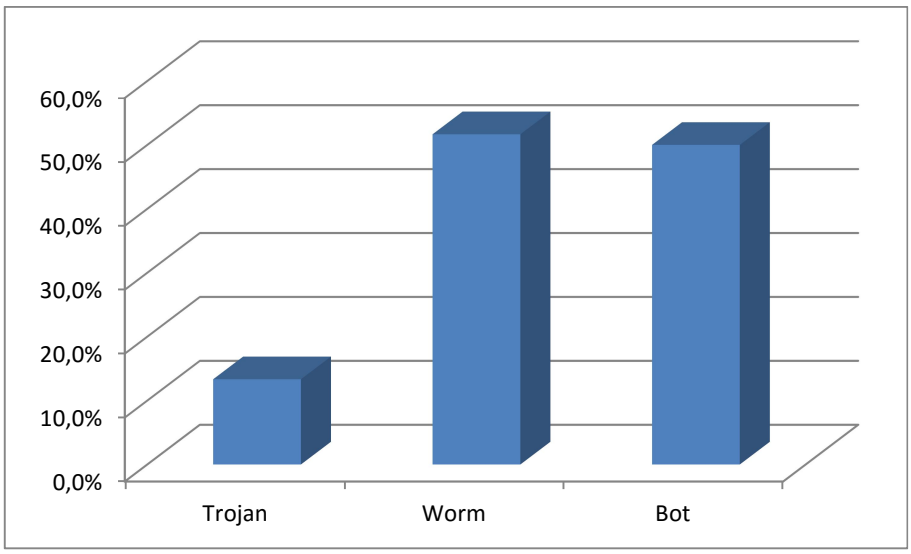
Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (10%), Worm (40%) και Bot (55%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (25%), Worm (80%) και Bot (50%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (5%), Worm (35%) και Bot (45%). Συνεπώς, στα Windows 8,1 και στο είδος Worm, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (35%), Windows 8.1 (51,7%) και Windows 10 (28,3%). Συνεπώς, στα Windows 8,1, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα Windows 10 (P<0.05).

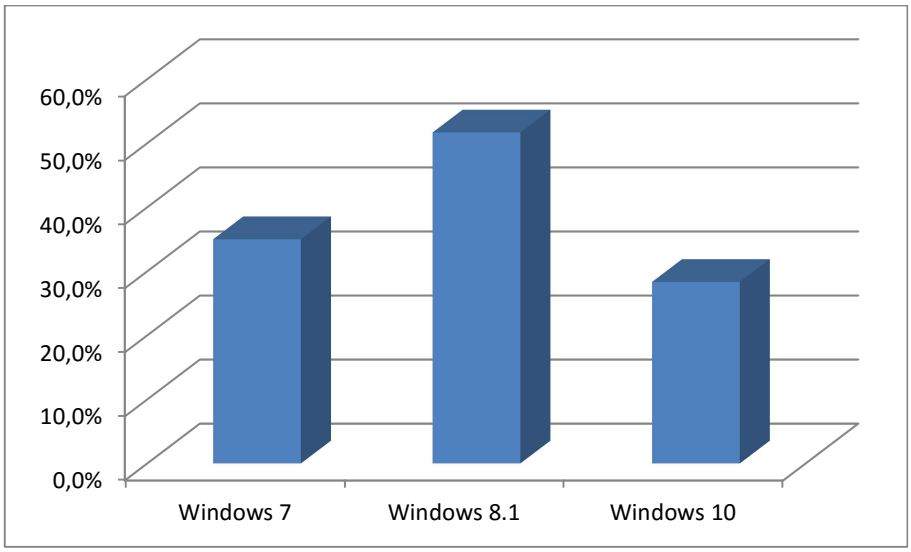
Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (13,3%), Worm (51,7%) και Bot (50%). Συνεπώς αυτή τα Trojan εμφανίζονται με στατιστικώς μικρότερη συχνότητα σε σχέση με τα Worm και Bot (P<0.05).



Διάγραμμα 29. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 17



Διάγραμμα 30. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 17, ανεξαρτήτως λειτουργικού συστήματος.



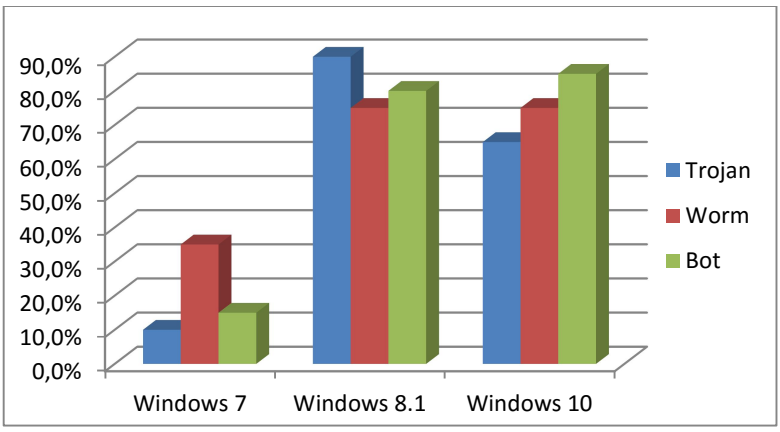
Διάγραμμα 31. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 17, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.12 Θέση 18

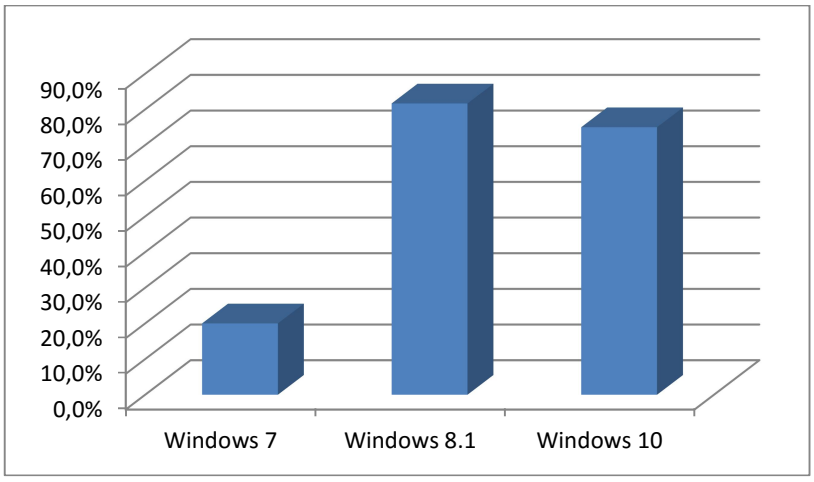
(%Systemdrive%\Windows\System32)

Στη θέση αυτή στα Windows 7 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (10%), Worm (35%) και Bot (15%). Στα Windows 8.1 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (90%), Worm (75%) και Bot (80%). Στα Windows 10 εμφανίζονται ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (65%), Worm (75%) και Bot (85%). Συνεπώς, στα Windows 7 και στα είδη Trojan, Worm και Bot, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (20%), Windows 8.1 (81,7%) και Windows 10 (75,0%). Συνεπώς, στα Windows 7, εμφανίζονται ευρήματα με στατιστικά μικρότερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).



Διάγραμμα 32. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού στη θέση 18

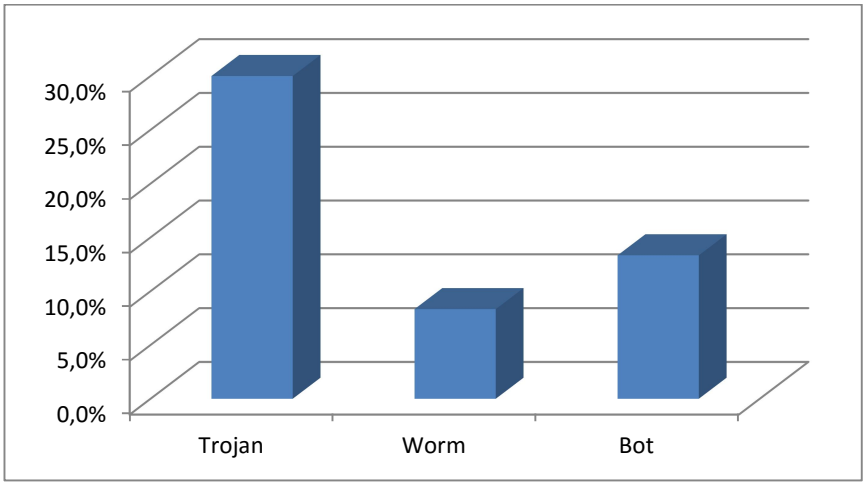


Διάγραμμα 33. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 18, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.13 Θέση 20

(%Systemdrive%\Windows\Globalization\Sorting\sortdefault.nls)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (30%), Worm (8,3%) και Bot (13,3%). Συνεπώς αυτή τα Trojan εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Worm (P<0.05).

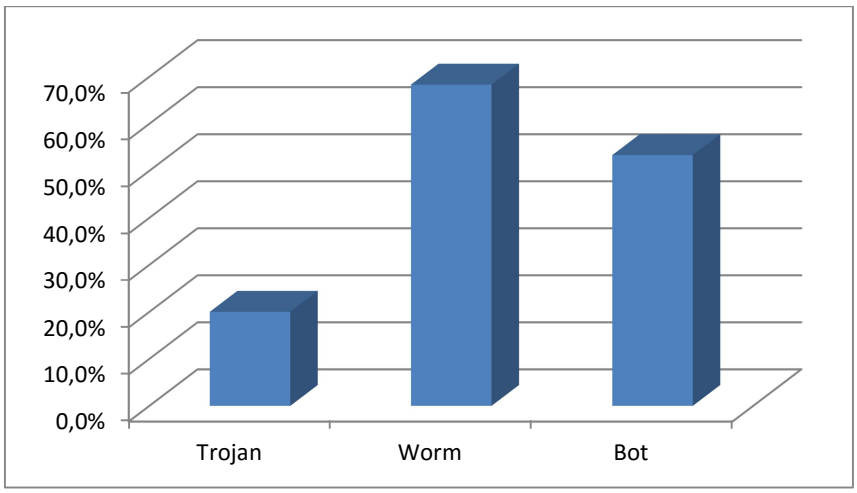


Διάγραμμα 34. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 20, ανεξαρτήτως λειτουργικού συστήματος.

4.3.14 Θέση 21

(%Systemdrive%\)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (20%), Worm (68,3%) και Bot (53,3%). Συνεπώς αυτή τα Trojan εμφανίζονται με στατιστικώς μικρότερη συχνότητα σε σχέση με τα Worm και Bot (P<0.05).

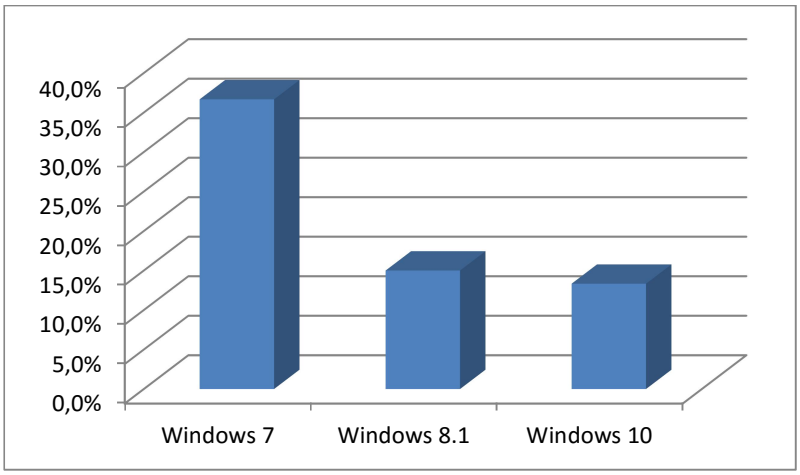


Διάγραμμα 35. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 21, ανεξαρτήτως λειτουργικού συστήματος.

4.3.15 Θέση 22

(HKEY_LOCAL_MACHINE\software\policies)

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (36,7%), Windows 8.1 (15%) και Windows 10 (13,3%). Συνεπώς, στα Windows 7, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα άλλα δύο λειτουργικά (P<0.05).

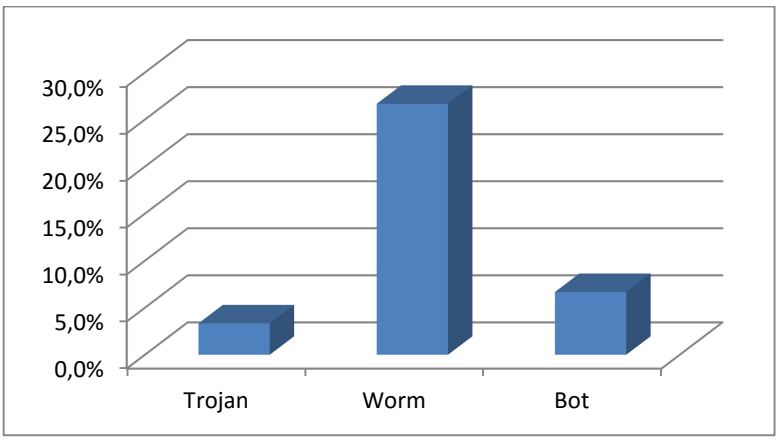


Διάγραμμα 36. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά λειτουργικό στη θέση 22, στο σύνολο των 3 κατηγοριών κακόβουλου λογισμικού

4.3.16 Θέση 23

(HKEY_LOCAL_MACHINE\SOFTWARE\Classes\)

Στην θέση αυτή οι 3 κατηγορίες κακόβουλου λογισμικού εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Trojan (3,3%), Worm (26,7%) και Bot (6,7%). Συνεπώς αυτή τα Worm εμφανίζονται με στατιστικώς μεγαλύτερη συχνότητα σε σχέση με τα Trojan και Bot (P<0.05).

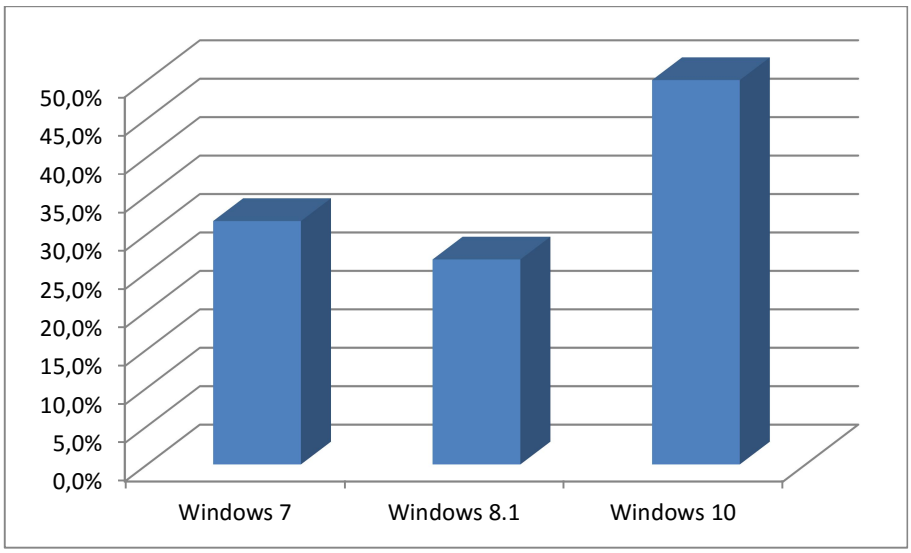


Διάγραμμα 37. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 23, ανεξαρτήτως λειτουργικού συστήματος.

4.3.17 Θέση 24

(HKEY_CURRENT_USER\Software\Microsoft)

Στην θέση αυτή τα 3 λειτουργικά εμφανίζουν ψηφιακά τεκμήρια σε ποσοστό για τα Windows 7 (31,7%), Windows 8.1 (26,7%) και Windows 10 (50%). Συνεπώς, στα Windows 10, εμφανίζονται ευρήματα με στατιστικά μεγαλύτερη συχνότητα σε σχέση με τα Windows 8,1 (P<0.05).



Διάγραμμα 38. Συχνότητα εμφάνισης ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού στη θέση 24, ανεξαρτήτως λειτουργικού συστήματος.

Κεφάλαιο 5

Συμπεράσματα

5.1 Λειτουργικότητα

Η εξέταση των δειγμάτων με κριτήριο τις διαφορετικές δυνατότητες τους, μας έδειξε ότι οι θέσεις

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
- %Systemdrive%\Windows\System32
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control
- %Systemdrive%\
- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\
- %Systemdrive%\Users\victim_user\AppData\

είναι εξίσου σημαντικές και για τις τέσσερεις κατηγορίες. Επιπλέον για τις υπόλοιπες θέσεις οι οποίες βρίσκονται σε ποσοστό πάνω από το 30% στον συνολικό αριθμό των δειγμάτων, δεν μπορεί να εξαχθεί ασφαλές συμπέρασμα για την λειτουργικότητα του δείγματος. Τα σημερινά δείγματα κακόβουλου λογισμικού έχουν πολλαπλές δυνατότητες και μηχανισμούς μόλυνσης, συνεπώς δεν μπορεί να γίνει διαχωρισμός τους ή να δημιουργηθούν μοναδικά χαρακτηριστικά με βάση αυτό το κριτήριο.

5.2 Λειτουργικό ή το είδος

Κατά την στατιστική ανάλυση με κριτήριο το είδος του κακόβουλου λογισμικού, παρατηρήθηκε ότι και για τα τρία είδη οι θέσεις HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ και %Systemdrive%\Windows\System32 βρίσκονται στις πέντε πρώτες ως ποσοστό εμφάνισης. Επίσης καταδεικνύεται ότι για κάθε είδος μία θέση είναι μοναδικά στην πρώτη πεντάδα. Συγκεκριμένα η θέση HKEY_LOCAL_MACHINE\SYSTEM για τα Trojan, η

HKEY_CURRENT_USER\Software\Microsoft για τα Worm και η HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\ για τα Bot. Δηλαδή υπάρχει ένα μοναδικό χαρακτηριστικό, που θα μπορούσε να δώσει μία πρώτη ένδειξη στον αναλυτή ψηφιακών τεκμηρίων για το είδος το κακόβουλου λογισμικού. Από την άλλη οι θέσεις

- %Systemdrive%\Windows\INF\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Setup
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CustomLocale\en-US
- HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
- Documents and Settings\[user name]\Start Menu\Programs\Startup

δεν έχουν να προσφέρουν κάποιες πληροφορίες, συνεπώς είναι θέσεις που δεν χρειάζεται να εξετάσει ο αναλυτής.

Αν η κύρια παράμετρος που εξετάζεται είναι το είδος του λειτουργικού συστήματος, τότε υπάρχουν οι παρακάτω κατευθύνσεις. Για τα τρία λειτουργικά που εξετάστηκαν η θέση HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ είναι σταθερά αυτή με το μεγαλύτερο ποσοστό ανεύρεσης τεκμηρίων. Για τα Windows 7 οι θέσεις HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\ και HKEY_LOCAL_MACHINE\SYSTEM είναι στην πρώτη πεντάδα, για τα Windows 8.1 η θέση %Systemdrive%\Users\victim_user\AppData\ και για τα Windows 10 οι θέσεις

- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control
- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\
- HKEY_CURRENT_USER\Software\Microsoft.

Κατά την διάρκεια της ιατροδικαστικής ανάλυσης ενός λειτουργικού συστήματος, με βάση την έκδοση του, ο εξεταστής μπορεί να εστιάσει σε διαφορετικά σημεία, πράγμα που θα επιταχύνει την έρευνα του και την εξαγωγή σημαντικών αποτελεσμάτων.

5.3 Σημαντικότερες θέσεις

Σημαντικά ευρήματα διαπιστώθηκαν όταν επίκεντρο της ανάλυσης ήταν οι θέσεις των ψηφιακών τεκμηρίων και εξετάστηκαν ως προς τα τρία λειτουργικά συστήματα, τις τρεις κατηγορίες κακόβουλου λογισμικού ή ο συνδυασμός αυτών των δύο.

5.3.1 Ενδείκτες ψηφιακών τεκμηρίων ανά τύπο κακόβουλου λογισμικού

Στις θέσεις HKEY_LOCAL_MACHINE\SYSTEM, HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall και %systemdrive%\Documents and Settings\[User Name]\Local Settings\Temp υπάρχει μεγαλύτερη πιθανότητα να ανιχνευθούν Trojan σε σχέση με Worm και Bot, ενώ στις θέσεις %Systemdrive%\Users\victim_user\AppData\ και %Systemdrive%\ αυτός ο τύπος κακόβουλου λογισμικού εμφανίζεται με μικρότερη συχνότητα. Τα ψηφιακά τεκμήρια που ανευρίσκονται στις θέσεις HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ και %Systemdrive%\Windows\Globalization\Sorting\sortdefault.nls , έχουν περισσότερες πιθανότητες να είναι Trojan παρά Worm.

Τα Worm έχουν μειωμένες πιθανότητες να ανιχνευθούν στις θέσεις HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer και HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ σε σχέση με τις άλλες δυο κατηγορίες. Όμοια τάση παρατηρείται στη θέση HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CustomLocale\en-US σε σχέση με τα Trojan, ενώ αντίθετα στη θέση HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ τα Worm έχουν μεγαλύτερη πιθανότητα εμφάνισης σε σχέση με αυτά.

5.3.2 Ενδείκτες ψηφιακών τεκμηρίων ανά λειτουργικό

Η μελέτη των Windows 10 αποφέρει περισσότερες πιθανότητες να ανιχνευθούν ψηφιακά τεκμήρια στις θέσεις HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control , HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ σε σχέση με

τα Windows 7 και 8.1, ενώ αντίθετη τάση παρατηρείται στη θέση HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls .

Στα Windows 7 και στις θέσεις HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SESSION και %Systemdrive%\Windows\System32 οι πιθανότητες να ανιχνευθούν ψηφιακά τεκμήρια είναι μειωμένες σε σχέση με τα άλλα 2 λειτουργικά. Το αντίθετο παρατηρείται στη θέση HKEY_LOCAL_MACHINE\software\policies σε σχέση με τα άλλα 2 λειτουργικά, αλλά και στη θέση %systemdrive%\Documents and Settings\[User Name]\Local Settings\Temp σε σχέση μόνο με τα Windows 8,1.

Τα Windows 8.1. εμφανίζουν μεγαλύτερη και μικρότερη συχνότητα ανεύρεσης ψηφιακών τεκμηρίων στη θέση %Systemdrive%\Users\victim_user\AppData\ και στη θέση HKEY_CURRENT_USER\Software\Microsoft αντίστοιχα σε σχέση με τα Windows 10.

5.3.3 Ενδείκτες ψηφιακών τεκμηρίων ανά λειτουργικό και ανά τύπο κακόβουλου λογισμικού

Τα ψηφιακά τεκμήρια που θα ανιχνευθούν σε συγκεκριμένες θέσεις και σε συγκεκριμένο λειτουργικό μπορούν να παρέχουν μια πρώτη ένδειξη για τον τύπο του κακόβουλου λογισμικού. Συγκεκριμένα, αν το λειτουργικό είναι Windows 10 και ανιχνευθούν ψηφιακά τεκμήρια στη θέση HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls τότε πιθανότατα πρόκειται για Trojan, ενώ στη θέση HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control πρόκειται για Bot. Αν εξετάζονται τα Windows 8.1. και απομονωθούν τεκμήρια τη θέση %Systemdrive%\Users\victim_user\AppData\ τότε πρόκειται με σημαντική πιθανότητα να αντιστοιχούν σε Worm. Στα Windows 7 και στη θέση HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SESSION τα ψηφιακά τεκμήρια με μεγαλύτερη συχνότητα αφορούν Trojan.

Αν το κακόβουλο λογισμικό που μελετάται είναι Bot τότε στη θέση HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ υπάρχει μεγαλύτερη πιθανότητα να ανιχνευθούν τεκμήρια στα Windows 10 σε σχέση με τα 8.1. Ενώ αν μελετάται Trojan στη θέση %systemdrive%\Documents and Settings\[User

Name]\Local Settings\Temp είναι πιο συχνή η εμφάνιση τους στα Windows 7 σε σχέση με τα 8.1.

5.4 Μελλοντική έρευνα

Με την πειραματική διαδικασία που ακολουθήθηκε και την στατιστική ανάλυση των αποτελεσμάτων, εντοπίστηκαν και αξιολογήθηκαν οι σημαντικότερες πιθανές θέσεις για την ανίχνευση ψηφιακών τεκμηρίων από κακόβουλο λογισμικό. Επιπλέον ανά τύπο λογισμικού, ανά λειτουργικό σύστημα κι ανά λειτουργικότητα της εφαρμογής καταγράφηκαν οι πλέον κοινές θέσεις που δημιουργούνται ψηφιακά τεκμήρια. Σε μελλοντική έρευνα θα μπορούσε η έρευνα αυτή να επεκταθεί σε άλλες κατηγορίες κακόβουλου λογισμικού (Ransomware, Backdoor κλπ) καθώς και σε άλλες μορφές λειτουργικότητας.

Τα συμπεράσματα τα παρούσας μεταπτυχιακής διατριβής θα μπορούσαν να μοντελοποιηθούν, με τελικό ζητούμενο την δημιουργία μιας λειτουργικής εφαρμογής. Με την χρήση της εφαρμογής αυτής ένας ερευνητής θα μπορεί με έναν εύκολο και γρήγορο τρόπο να πραγματοποιήσει μία προκαταρκτική ανάλυση για την ύπαρξη ή μη ενδείξεων κακόβουλου λογισμικού σε ένα σύστημα. Η ύπαρξη κακόβουλου λογισμικού σε ένα σύστημα θα μπορούσε να είναι η αιτία για την εκδίκαση μιας υπόθεσης με λάθος πειστήρια. Επιπλέον θα μπορούσε να βρει εφαρμογή σε Host based συστήματα ανίχνευσης εισβολής (HIDS) ή συστήματα προστασίας εισβολών (HIPS) καθώς και σε σαρωτές ευπαθειών.

Παράρτημα Α
Κακόβουλο λογισμικό

	Category	Virus name	SPY-STEAL DATA	C&C	BACKDOOR	STEALTH
1	Trojan	Trojan-Spy.Win32.Zbot.wijf	X	X		
2	Trojan	Trojan.GenericKD.3015891		X		
3	Trojan	Trojan.GenericKD.3015909			X	
4	Trojan	Trojan/Win32.Yakes			X	
5	Trojan	Trojan.GenericKD.3016131			X	
6	Trojan	Trojan/W32.KRBanker		X		
7	Trojan	Trojan-Spy.Win32.FlyStudio.ij			X	X
8	Trojan	Trojan-Dropper.Win32.Injector.nyds			X	
9	Trojan	Trojan.Zboter	X	X		
10	Trojan	Trojan-Spy.Win32.Recam.yue	X	X		
11	Trojan	Trojan. Tesla!1.A322		X		X
12	Trojan	Trojan.Win32.Waldek.cbp			X	
13	Trojan	Trojan.Win32.Waldek.cbm			X	
14	Trojan	Trojan.Win32.Dridex.v	X	X		X
15	Trojan	Trojan.Win32.Tepfer.psxezj	X	X		
16	Trojan	Trojan.Win32.Yakes.owmp			X	
17	Trojan	Trojan.Win32.KeyLogger.auqd	X			
18	Trojan	Trojan.GenericKD.3023498		X		
19	Trojan	Trojan.Generic.8742442	X	X		X
20	Trojan	Trojan.Generic.7738292	X			
21	Worm	Win32.Gamarue	X	X		X
22	Worm	W32.Cridex.A.worm	X	X		X
23	Worm	Worm.VBS.Agent		X		
24	Worm	Worm.Win32.3DStars			X	X
25	Worm	Worm.Generic3.PEM			X	
26	Worm	Worm.Win32.Mira.A	X			
27	Worm	Worm.Generic2.CMVO	X			
28	Worm	Worm.Win32.Cake				X
29	Worm	Worm.Win32.Fever	X			X
30	Worm	Worm.Win32.Monkey.exe	X			
31	Worm	Worm.Win32.Mydoom.a.exe			X	X
32	Worm	Worm.Win32.Pikachu.exe	X			
33	Worm	Worm.Win32.Postman.exe				X
34	Worm	Worm.Win32.Sharpei.a.exe				X
35	Worm	Worm.Win32.Silver.exe				X
36	Worm	Worm.Win32.Sobig.exe	X	X		
37	Worm	Worm.KOOBFCE.SMC	X		X	
38	Worm	W32/Wabot	X	X		

39	Worm	Email-Worm.Win32.Mydoom.l			X	X
40	Worm	Email-Worm.Win32.Naked	X			
41	Botnet	Win32.Lolbot.aoi			X	
42	Botnet	WORM/IrcBot.tlq	X	X	X	
43	Botnet	W32.Jorik_Lolbot.O!tr	X		X	
44	Botnet	Win32.SdBot.aamk	X	X	X	
45	Botnet	W32.ZBot.42352	X		X	X
46	Botnet	Win32.Jorik.SdBot.e			X	
47	Botnet	MSIL.NanoBot.ibh			X	
48	Botnet	Win32.Zbot.vtii	X		X	X
49	Botnet	Win32.Ngrbot.anak				X
50	Botnet	Win32.Alinaos.G	X	X		
51	Botnet	GenericKD.2143403			X	
52	Botnet	Win32/ChkBot.A			X	
53	Botnet	MSIL/Lizarbot.A	X	X	X	
54	Botnet	Win32.Jorik.Lolbot.f	X	X	X	
55	Botnet	Win32.Zbot.sbdj	X		X	X
56	Botnet	MSIL.NanoBot.bi	X		X	
57	Botnet	Win32.Ngrbot.uyk				X
58	Botnet	Win32.Boht.qo		X	X	
59	Botnet	W32/Zbot.AJJU!tr	X		X	X
60	Botnet	Win32.VBInject				X

Παράρτημα Β
Αποτελέσματα

Η αρίθμηση (1-24) πρώτη στήλη αφορά τις διαφορετικές θέσεις όπου βρέθηκαν τα forensics.

Σε κάθε Trojan (αρίθμηση i-xx) έχουν καταγραφεί για τα 3 λειτουργικά (7, 8.1, 10) τα σημεία στα οποία βρέθηκαν digital forensics.

	Trojan																													
	i			ii			iii			iv			v			vi			vii			viii			ix			x		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1	●	●	●							●	●		●	●					●	●		●		●		●	●			
2	●	●	●							●	●	●	●	●					●			●		●	●	●		●		●
3			●			●			●															●						●
4	●		●	●	●	●				●	●	●				●	●	●		●										
5	●			●	●	●	●	●		●	●	●				●	●	●												
6	●									●	●	●																		
7	●	●				●				●	●	●														●				
8	●	●	●	●	●	●				●	●	●	●			●	●	●	●	●	●		●		●	●		●	●	
9				●		●				●	●	●	●	●		●	●	●							●	●			●	
10																												●		●
11	●	●								●	●	●																		
12			●	●	●					●	●	●																		
13	●			●	●	●																								
14	●	●	●																							●				
15																									●	●				
16	●		●							●		●	●						●		●				●			●		●
17		●								●	●	●					●								●	●				
18		●	●		●	●		●	●		●	●		●		●	●	●	●	●						●			●	●
19											●	●																		
20	●		●							●	●	●	●	●								●		●	●	●				
21										●	●	●							●	●					●	●			●	
22	●		●		●					●																				
23																														
24	●	●	●			●																				●		●		●

	Trojan																													
	xi			xii			xiii			xiv			xv			xvi			xvii			xviii			xix			xx		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1	●	●														●	●	●				●	●		●	●	●	●	●	●
2	●	●											●	●		●	●	●	●	●		●	●		●	●	●		●	●
3																			●	●										●
4				●	●	●	●	●	●	●	●	●			●	●	●	●	●		●				●	●	●		●	●
5				●	●	●	●	●	●	●	●	●				●	●	●	●						●	●	●	●		
6																			●	●					●	●	●			
7			●		●		●		●	●		●							●	●					●			●		
8				●	●	●	●	●	●	●	●	●	●		●				●	●	●	●			●	●	●	●	●	●
9				●		●	●		●	●		●			●				●	●	●	●	●							
10																														
11																														
12																														
13						●			●			●		●					●	●	●									
14																			●	●										●
15																			●											
16	●															●			●	●		●			●					
17																				●										
18		●			●	●		●	●		●	●		●	●		●	●		●	●		●			●	●			
19																														
20																●	●	●				●	●					●	●	●
21														●	●				●	●										
22				●		●	●		●	●		●							●	●					●			●	●	●
23																			●	●										
24	●	●				●			●			●			●				●	●	●									●

Σε κάθε Worm (αρίθμηση i-xx) έχουν καταγραφεί για τα 3 λειτουργικά (7, 8.1, 10) τα σημεία στα οποία βρέθηκαν digital forensics.

	Worm																													
	i			ii			iii			iv			v			vi			vii			viii			ix			x		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1	●	●	●				●									●		●												
2	●	●	●				●			●	●		●	●		●	●											●	●	
3			●								●	●		●	●														●	●
4	●	●	●			●	●					●			●	●	●	●									●	●			●
5	●	●	●		●	●	●									●														
6	●	●	●		●	●	●								●															
7					●	●	●												●	●	●									
8	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
9		●	●				●			●	●	●	●	●	●		●											●	●	●
10																														
11																														
12																														
13					●	●	●			●		●			●	●	●	●												●
14																														
15							●																							
16																														
17		●			●						●			●			●						●			●			●	
18	●	●	●		●	●					●	●		●	●		●	●							●	●	●		●	●
19																														
20							●									●		●												
21	●	●	●		●	●	●			●	●	●	●	●	●	●	●	●				●		●	●		●	●	●	●
22	●	●	●		●	●	●									●														
23		●	●	●												●	●	●												
24		●	●				●			●		●			●	●	●	●												●

	Worm																													
	xi			xii			xii			xiv			xv			xvi			xvii			xviii			xix			xx		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1													●	●	●	●	●	●												
2				●	●								●	●	●	●	●	●										●	●	
3					●	●								●	●														●	●
4						●		●									●	●							●	●	●			●
5							●	●	●							●									●	●	●			
6																									●	●	●			
7									●				●	●	●	●	●	●							●	●	●			
8	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●
9				●	●	●	●		●																●			●	●	●
10													●	●	●															
11																														
12																														
13	●	●	●			●			●				●	●	●	●	●	●							●	●	●			●
14																														
15																									●					
16																									●	●	●			
17	●	●	●	●	●		●	●	●			●	●	●	●	●	●	●				●	●	●	●	●	●	●	●	
18	●	●	●	●	●	●		●	●		●	●					●	●	●			●	●	●		●	●	●	●	●
19																														
20																	●		●											
21				●	●	●					●	●	●	●	●					●	●	●	●	●	●	●	●	●	●	●
22				●			●		●							●									●			●		
23				●	●							●	●	●					●	●	●							●	●	
24	●	●	●			●	●	●	●	●	●	●	●	●	●										●	●	●			●

Σε κάθε Bot (αρίθμηση i-x) έχουν καταγραφεί για τα 3 λειτουργικά (7, 8.1, 10) τα σημεία στα οποία βρέθηκαν digital forensics.

•	Bots																													
	i			ii			iii			iv			v			vi			vii			viii			ix			x		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1	•		•						•							•	•	•	•	•	•	•		•			•	•			•
2	•						•	•								•	•		•	•							•	•	•	•	
3			•					•	•								•	•		•	•									•	•
4			•		•	•			•		•	•					•	•			•	•							•		•
5																•	•					•							•		
6																						•		•							
7	•															•	•									•	•				
8	•	•	•		•	•	•		•		•	•				•	•	•	•		•	•	•	•				•			•
9		•	•						•							•	•	•			•	•		•							•
10																			•	•											
11																															
12																															
13	•		•						•							•	•	•			•					•	•				•
14																•	•									•	•				
15																						•		•							
16													•	•	•																
17																•	•		•	•		•		•	•	•	•	•	•		
18		•	•		•	•		•	•		•	•					•	•			•		•	•					•	•	•
19																															
20	•																									•	•				
21		•	•						•							•	•	•	•									•	•	•	•
22																•	•					•							•		
23																•	•														
24	•		•						•							•	•	•	•	•	•	•		•		•	•				•

	Bots																													
	xi			xii			xii			xiv			xv			xvi			xvii			xviii			xix			xx		
	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10	7	8.1	10
1		●	●	●	●	●																●	●		●	●	●			
2		●	●	●	●	●				●	●								●			●	●		●	●	●	●	●	
3											●	●									●		●	●					●	●
4		●	●	●				●	●			●	●	●	●	●	●	●			●	●		●		●	●			●
5		●	●	●			●						●	●	●							●			●				●	
6		●	●				●						●	●	●							●								
7				●			●									●						●	●		●					
8	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
9	●	●	●	●	●	●	●	●	●			●	●			●	●	●	●	●	●	●	●	●		●	●			●
10																														
11																														
12																														
13				●	●	●	●					●				●					●	●	●	●			●			●
14				●	●	●	●												●						●	●				●
15																														
16																														
17	●	●	●	●	●	●	●	●	●			●				●	●	●	●		●		●		●	●	●			
18		●	●	●	●	●	●	●	●		●	●				●	●	●		●	●		●	●		●	●		●	●
19																														
20		●	●	●	●	●																								
21	●	●	●	●	●	●	●	●	●			●				●	●	●	●	●		●	●	●	●	●	●			●
22				●									●									●	●		●	●	●			
23							●									●														
24							●	●	●				●			●	●	●												

Παράρτημα Γ

Θέσεις ψηφιακών τεκμηρίων

	Digital forensics locations
1	HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls\CustomLocale\en-US
2	HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Nls
3	HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SESSION
4	HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control
5	HKEY_LOCAL_MACHINE\SYSTEM
6	HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc
7	HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion
8	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
9	HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\
10	HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
11	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Setup
12	HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
13	HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\
14	HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
15	Documents and Settings\[user name]\Start Menu\Programs\Startup
16	%systemdrive%\Documents and Settings\[User Name]\Local Settings\Temp
17	%Systemdrive%\Users\victim_user\AppData\
18	%Systemdrive%\Windows\System32
19	%Systemdrive%\Windows\INF\
20	%Systemdrive%\Windows\Globalization\Sorting\sortdefault.nls
21	%Systemdrive%\
22	HKEY_LOCAL_MACHINE\software\policies
23	HKEY_LOCAL_MACHINE\SOFTWARE\Classes\
24	HKEY_CURRENT_USER\Software\Microsoft

Παράρτημα Δ
Διαγράμματα

Παράρτημα Ε
Εικόνες

Βιβλιογραφία

Abdelhamid, N., Ayesh, A. & Thabtah, F. (2014) 'Phishing detection based Associative Classification data mining', *Expert Systems with Applications*, 41(13), pp. 5948–5959. doi: 10.1016/j.eswa.2014.03.019.

AV-TEST (no date). Available at: https://www.av-test.org/en/ (Accessed: 25 March 2016).

Babić, D., Reynaud, D. & Song, D. (2011) 'Malware analysis with tree automata inference'. Springer-Verlag, pp. 116–131. Available at: http://dl.acm.org/citation.cfm?id=2032305.2032315 (Accessed: 25 March 2016).

Bächer, P., Holz, T., Kötter, M. & Wicherski, G. (2005) *Know your Enemy: Tracking Botnets*. Available at: https://www.honeynet.org/papers/bots (Accessed: 10 April 2016).

Balthrop, J., Forrest, S., Newman, M. E. J. & Williamson, M. M. (2004) 'Computer science. Technological networks and the spread of computer viruses.', *Science (New York, N.Y.)*. American Association for the Advancement of Science, 304(5670), pp. 527–9. doi: 10.1126/science.1095845.

Bayer, U., Comparetti, P. M., Hlauschek, C., Kruegel, C. & Kirda, E. (2009) 'Scalable, behavior-based malware clustering', *Network and Distributed System Security Symposium (NDSS)*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.7690&rep=rep1&type=pdf.

Bayer, U., Habibi, I., Balzarotti, D., Kirda, E. & Kruegel, C. (2014) *A View on Current Malware Behaviors*. Available at: http://static.usenix.org/event/leet09/tech/full_papers/bayer/bayer_html/ (Accessed: 31 January 2016).

Bayer, U., Kirda, E. & Kruegel, C. (2010) 'Improving the efficiency of dynamic malware analysis', in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*. New York, New York, USA: ACM Press, p. 1871. doi: 10.1145/1774088.1774484.

Bayer, U., Moser, A., Kruegel, C. & Kirda, E. (2006) 'Dynamic analysis of malicious code', *Journal in Computer Virology*, 2(1), pp. 67–77. doi: 10.1007/s11416-006-0012-2.

Bitdefender (2010) *How To Remove A Fake Antivirus Infection*, *bitdefender*. Available at: http://www.bitdefender.com/tech-assist/self-help/how-to-remove-a-fake-antivirus-infection.html (Accessed: 27 March 2016).

Bremer, J. (2014) 'Secure 2014', in.

Bristow, J. S. (2013) *LEARNING ENTERPRISE MALWARE TRIAGE FROMAUTOMATIC DYNAMIC ANALYSIS*. AIR FORCE INSTITUTE OF TECHNOLOGY.

Brumley, D., Hartwig, C., Liang, Z., Newsome, J., Song, D. & Yin, H. (2008) 'Automatically Identifying Trigger-based Behavior in Malware', *Botnet Detection*, pp. 65–88. doi: http://dx.doi.org/10.1007/978-0-387-68768

Carrier, B. (2005) *File System Forensics Analysis*, *Addison Wesley*.

Carvey, H. (2009) *Windows Forensic Analysis DVD Toolkit*. Available at: https://books.google.com/books?hl=el&lr=&id=5hvSrBGVfIgC&pgis=1 (Accessed: 14 April 2016).

Carvey, H. (2011) *Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry*. Available at: http://www.abebooks.com/servlet/BookDetailsPL?bi=18065505231&searchurl=tn%3Dwindows%2520registry%2520forensics%2520advanced%2520digital%2520forensic%2520analysis%2520of%2520the%2520windows%2520registry (Accessed: 25 March 2016).

Carvey, H. & Altheide, C. (2005) 'Tracking USB storage: Analysis of windows artifacts generated by USB storage devices', *Digital Investigation*, 2(2), pp. 94–100. doi: 10.1016/j.diin.2005.04.006.

CERT-EU (2012) 'Data Acquisition Guidelines for Investigation Purposes'.

Cert-Eu (2012) 'Incident Response Methodology', *Cert-Eu*. Available at: http://cert.europa.eu/static/WhitePapers/CERT-EU-SWP_11_003_v2.pdf.

Checkpoint (2013) 'Malware Report Malware Screen Shots', *Checkpoint*.

Cheng, S.-M., Ao, W. C., Chen, P.-Y. & Chen, K.-C. (2011) 'On Modeling Malware Propagation in Generalized Social Networks', *IEEE Communications Letters*. IEEE, 15(1), pp. 25–27. doi: 10.1109/LCOMM.2010.01.100830.

Christodorescu, M. & Jha, S. (2003) 'Static analysis of executables to detect malicious patterns'. USENIX Association, p. 12. Available at: http://dl.acm.org/citation.cfm?id=1251353.1251365 (Accessed: 25 March 2016).

Christodorescu, M., Jha, S. & Kruegel, C. (2007) 'Mining specifications of malicious behavior', in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering - ESEC-FSE '07*. New York, New York, USA: ACM Press, p. 5. doi: 10.1145/1287624.1287628.

Christodorescu, M., Jha, S., Seshia, S. a., Song, D. & Bryant, R. E. (2005) 'Semantics-aware malware detection', *Proceedings - IEEE Symposium on Security and Privacy*, pp. 32–46. doi: 10.1109/SP.2005.20.

CIRCL (2009) *Malware Discovery and potential Removal*, *Computer Incident Response Center Luxembourg*. Available at: https://www.circl.lu/pub/tr-09/ (Accessed: 31 January 2016).

Cisco (2016) *What Is the Difference: Viruses, Worms, Trojans, and Bots?* Available at: http://www.cisco.com/c/en/us/about/security-center/virus-differences.html (Accessed: 10 April 2016).

Cnn (2016) *Nearly 1 million new malware threats released every day*. Available at: http://money.cnn.com/2015/04/14/technology/security/cyber-attack-hacks-security/ (Accessed: 16 April 2016).

Collberg, C., Thomborson, C. & Low, D. (1998) 'Manufacturing cheap, resilient, and stealthy opaque constructs', in *Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '98*. New York, New York, USA: ACM Press, pp. 184–196. doi: 10.1145/268946.268962.

Collie, J. (2013) 'The windows IconCache.db: A resource for forensic artifacts from USB connectable devices', *Digital Investigation*, 9(3-4), pp. 200–210. doi: 10.1016/j.diin.2013.01.006.

Comodo (2016) *How Does an Antivirus Work to Get Rid of Viruses?* Available at: https://antivirus.comodo.com/how-antivirus-software-works.php (Accessed: 8 May 2016).

Cooke, E., Jahanian, F. & McPherson, D. (2005) 'The zombie roundup: Understanding, detecting, and disrupting botnets', *Networks*, 7, pp. 39–44. Available at: http://www.usenix.org/events/sruti05/tech/full_papers/cooke/cooke_html.

Costa, M., Crowcroft, J., Castro, M., Rowstron, A., Zhou, L., Zhang, L. & Barham, P. (2005) 'Vigilante', *ACM SIGOPS Operating Systems Review*. ACM, 39(5), p. 133. doi: 10.1145/1095809.1095824.

Dalziel, H. (2014) *How to Defeat Advanced Malware: New Tools for Protection and Forensics*. Elsevier Science. Available at: https://books.google.com/books?id=LwIbBQAAQBAJ&pgis=1 (Accessed: 25 March 2016).

Dash, P. (2013) *Getting Started with Oracle VM VirtualBox*. PACKT Books. Available at: https://www.packtpub.com/virtualization-and-cloud/getting-started-oracle-vm-virtualbox (Accessed: 25 March 2016).

Dolan-Gavitt, B. (2008) 'Forensic analysis of the Windows registry in memory', *Digital Investigation*, 5, pp. S26–S32. doi: 10.1016/j.diin.2008.05.003.

EC-Council (2010) *Computer Forensics Investigating Hard Disks, File & Operating Systems*, *EC-Council*. Cengage Learning.

Egele, M., Scholte, T., Kirda, E. & Kruegel, C. (2012) 'A survey on automated dynamic malware-analysis techniques and tools', *ACM Computing Surveys*. ACM, 44(2), pp. 1–42. doi: 10.1145/2089125.2089126.

Enigmasoftware (2014) *Trojan-Banker.Win32.Banker.auzi Removal Report*, *enigmasoftware*. Available at: http://www.enigmasoftware.com/trojanbankerwin32bankerauzi-removal/ (Accessed: 27 March 2016).

FakeNet (2012) *FakeNet*. Available at: http://practicalmalwareanalysis.com/fakenet/ (Accessed: 25 March 2016).

FireEye (2012) *Flamer/sKyWIper Malware: Analysis « Threat Research*, *FireEye*. Available at: https://www.fireeye.com/blog/threat-research/2012/05/flamerskywiper-analysis.html (Accessed: 27 March 2016).

Fnal (2016) *Common Windows Trojan/Application Startup Locations*. Available at: https://security.fnal.gov/cookbook/WinStartup.html (Accessed: 27 March 2016).

Franolich, J. (2012) *Four Focus Areas of Malware Analysis*. Available at: https://digital-forensics.sans.org/blog/2012/07/26/four-focus-areas-of-malware-analysis (Accessed: 26 March 2016).

Fredrikson, M., Jha, S., Christodorescu, M., Sailer, R. & Yan, X. (2010) 'Synthesizing Near-Optimal Malware Specifications from Suspicious Behaviors', in *2010 IEEE Symposium on Security and Privacy*. IEEE, pp. 45–60. doi: 10.1109/SP.2010.11.

Gehringer, E. (2016) *Worms Viruses Trojans*. Available at: https://ethics.csc.ncsu.edu/abuse/wvt/ (Accessed: 10 April 2016).

Gil, S., Kott, A. & Barabási, A.-L. (2014) 'A genetic epidemiology approach to cyber-security.', *Scientific reports*, 4, p. 5659. doi: 10.1038/srep05659.

Gordon, S. (no date) 'What Is Wild?' Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.9914 (Accessed: 26 March 2016).

Greamo, C. & Ghosh, A. (2011) 'Sandboxing and Virtualization: Modern Tools for Combating Malware', *IEEE Security & Privacy Magazine*. IEEE Educational Activities Department, 9(2), pp. 79–82. doi: 10.1109/MSP.2011.36.

Gwallgofi (2016) *Cuckoo Sandbox :Quick Start Guide*. Available at: http://gwallgofi.com/cuckoo-guide/ (Accessed: 9 April 2016).

Harrell, C. (2014) 'Finding Malware'.

Hofmeyr, S. A., Forrest, S. & Somayaji, A. (1998) 'Intrusion detection using sequences of system calls', *Journal of Computer Security*. IOS Press, 6(3), pp. 151–180. Available at: http://dl.acm.org/citation.cfm?id=1298081.1298084 (Accessed: 25 March 2016).

Horsman, G., Laing, C. & Vickers, P. (2014) 'A case-based reasoning method for locating

evidence during digital forensic device triage', *Decision Support Systems*. Elsevier B.V., 61(1), pp. 69–78. doi: 10.1016/j.dss.2014.01.007.

Hosmer, C. (2014) *Python Forensics*. Available at: http://store.elsevier.com/Python-Forensics/Chet-Hosmer-/isbn-9780124186767/ (Accessed: 25 March 2016).

Hunt, G. & Brubacher, D. (1999) 'Detours: binary interception of Win32 functions'. USENIX Association, p. 14. Available at: http://dl.acm.org/citation.cfm?id=1268427.1268441 (Accessed: 25 March 2016).

Ianelli, N. & Hackworth, A. (2007) 'Botnets as a Vehicle for Online Crime', *The International Journal of Forensic Computer Science*, pp. 19–39. doi: 10.5769/J200701002.

Jason T. Luttgens, Pepe, M. & Mandia, K. (2014) *Incident Response & Computer Forensics*. McGraw-Hill Education. Available at: http://www.amazon.com/Incident-Response-Computer-Forensics-Edition/dp/0071798684 (Accessed: 25 March 2016).

Kleiman, D. & Hunter, L. E. (2006) *Winternals Defragmentation, Recovery, and Administration Field Guide*. Syngress. Available at: https://books.google.com/books?id=F4Cw4ny6nNQC&pgis=1 (Accessed: 27 March 2016).

Kolbitsch, C., Comparetti, P. M., Kruegel, C., Kirda, E., Zhou, X. & Wang, X. (2009) 'Effective and efficient malware detection at the end host'. USENIX Association, pp. 351–366. Available at: http://dl.acm.org/citation.cfm?id=1855768.1855790 (Accessed: 25 March 2016).

Kruegel, C., Robertson, W. and Vigna, G. (no date) 'Detecting Kernel-Level Rootkits Through Binary Analysis', in *20th Annual Computer Security Applications Conference*. IEEE, pp. 91–100. doi: 10.1109/CSAC.2004.19.

Li, Y., Hui, P., Jin, D., Su, L. & Zeng, L. (2014) 'Optimal Distributed Malware Defense in Mobile Networks with Heterogeneous Devices', *IEEE Transactions on Mobile Computing*. IEEE, 13(2), pp. 377–391. doi: 10.1109/TMC.2012.255.

Lindorfer, M., Di Federico, A., Maggi, F., Comparetti, P. M. & Zanero, S. (2012) 'Lines of malicious code', in *Proceedings of the 28th Annual Computer Security Applications Conference on - ACSAC '12*. New York, New York, USA: ACM Press, p. 349. doi:

10.1145/2420950.2421001.

Linn, C. & Debray, S. (2003) 'Obfuscation of executable code to improve resistance to static disassembly', in *Proceedings of the 10th ACM conference on Computer and communication security - CCS '03*. New York, New York, USA: ACM Press, p. 290. doi: 10.1145/948109.948149.

Liu, W., Liu, C., Liu, X., Cui, S. & Huang, X. (2016) 'Modeling the spread of malware with the influence of heterogeneous immunization', *Applied Mathematical Modelling*. Elsevier Inc., 40(4), pp. 3141–3152. doi: 10.1016/j.apm.2015.09.105.

Maio, G. De (2014) 'On The Evolution of Digital Evidence : Novel Approaches for Cyber Investigation', pp. 1–150.

Malekal (2016) *Malekal*. Available at: http://malwaredb.malekal.com/ (Accessed: 10 April 2016).

Malicious-streams (2014) *Digging for Malware: Suspicious Filesystem Geography*. Available at: http://www.malicious-streams.com/resources/articles/DGMW1_Suspicious_FS_Geography.html (Accessed: 31 January 2016).

Malin, C. H., Casey, E. & Aquilina, J. M. (2013) *Malware Forensics Field Guide for Windows Systems*, *Journal of Chemical Information and Modeling*. doi: 10.1017/CBO9781107415324.004.

Malware.lu (2016) *Malware.lu*. Available at: https://malware.lu/ (Accessed: 10 April 2016).

Malwarebytes (2014) *Removal instructions for Cryptowall*, *malwarebytes*. Available at: https://forums.malwarebytes.org/topic/150193-removal-instructions-for-cryptowall/ (Accessed: 27 March 2016).

Malwareremovalguides (2013) *Trojan.Ransom removal instructions*, *malwareremovalguides*. Available at: http://www.malwareremovalguides.info/pum-userwload-trojan-ransom-removal-instructions/ (Accessed: 27 March 2016).

MalwareTips (2016) *MalwareTips*. Available at: https://malwaretips.com/ (Accessed: 10

April 2016).

Martignoni, L., Stinson, E., Fredrikson, M., Jha, S. & Mitchell, J. C. (2008) *Recent Advances in Intrusion Detection*. Edited by R. Lippmann, E. Kirda, & A. Trachtenberg. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science). doi: 10.1007/978-3-540-87403-4.

McAfee Labs (2015) 'McAfee Labs Threats Report', (November). Available at: www.mcafee.com/us/mcafee-labs.aspx.

Mee, V., Tryfonas, T. & Sutherland, I. (2006) 'The Windows Registry as a forensic artefact: Illustrating evidence collection for Internet usage', *Digital Investigation*, 3(3), pp. 166–173. doi: 10.1016/j.diin.2006.07.001.

MHR (no date) *Malware Hash Registry - Team Cymru*. Available at: http://www.team-cymru.org/MHR.html (Accessed: 25 March 2016).

Micro, T. & Paper, W. (2006) 'Taxonomy of Botnet Threats', *Micro*, (November), pp. 1–15. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Taxonomy+of+Botnet+Threats#0.

Microsoft (2016a) *Visual Studio - Microsoft Developer Tools*. Available at: https://www.visualstudio.com/ (Accessed: 8 May 2016).

Microsoft (2016b) Windows lifecycle fact sheet. Available at: http://windows.microsoft.com/en-us/windows/lifecycle(Accessed: 8 May 2016).

Microsoft (2016c) *Windows registry information for advanced users*. Available at: https://support.microsoft.com/en-us/kb/256986 (Accessed: 27 March 2016).

Microsoft (no date) *HKEY_CLASSES_ROOT Key*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms724475%28v=vs.85%29.aspx?f=255&MSPPError=-2147217396 (Accessed: 27 March 2016).

Mishra, B. K. & Pandey, S. K. (2014) 'Dynamic model of worm propagation in computer network', *Applied Mathematical Modelling*, 38(7-8), pp. 2173–2179. doi: 10.1016/j.apm.2013.10.046.

Misra, A. K., Verma, M. & Sharma, A. (2014) 'Capturing the interplay between malware and anti-malware in a computer network', *Applied Mathematics and Computation*. Elsevier Inc., 229, pp. 340–349. doi: 10.1016/j.amc.2013.12.059.

Moser, A., Kruegel, C. & Kirda, E. (2007a) 'Exploring Multiple Execution Paths for Malware Analysis', in *2007 IEEE Symposium on Security and Privacy (SP '07)*. IEEE, pp. 231–245. doi: 10.1109/SP.2007.17.

Moser, A., Kruegel, C. & Kirda, E. (2007b) 'Limits of Static Analysis for Malware Detection', in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, pp. 421–430. doi: 10.1109/ACSAC.2007.21.

National Institute of Justice (2008) 'Electronic crime scene investigation: A guide for first responders', *Nij*. Available at: www.ojp.usdoj.gov/nij.

NetMarketShare (2016) *Operating system market share*. Available at: https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0&qpcustomb= (Accessed: 21 March 2016).

Norton (2010) 'Malware Removal Guide'.

Oberhumer,M., M. (2004) *UPX: the Ultimate Packer for eXecutables - Homepage*. Available at: http://upx.sourceforge.net/ (Accessed: 25 March 2016).

Oktavianto, D. & Muhardianto, I. (2013) *Cuckoo Malware Analysis*. Available at: http://books.google.com/books?hl=en&lr=&id=KXNZAQAAQBAJ&pgis=1.

Oracle (2016) *VirtualBox*. Available at: https://www.virtualbox.org/ (Accessed: 9 April 2016).

Panda Security (2007) *Trojans*. Available at: http://www.pandasecurity.com/homeusers/security-info/classic-malware/trojan/ (Accessed: 10 April 2016).

Park, Y., Reeves, D. S. & Stamp, M. (2013) 'Deriving common malware behavior through graph clustering', *Computers and Security*. Elsevier Ltd, 39(PART B), pp. 419–430. doi: 10.1016/j.cose.2013.09.006.

Park, Y., Zhang, Q., Reeves, D. & Mulukutla, V. (2010) 'AntiBot: Clustering Common Semantic Patterns for Bot Detection', in *2010 IEEE 34th Annual Computer Software*

and Applications Conference. IEEE, pp. 262–272. doi: 10.1109/COMPSAC.2010.33.

Pillow (2016) *Pillow*. Available at: https://pypi.python.org/pypi/Pillow/3.2.0 (Accessed: 10 April 2016).

Play It Safe (no date). Available at: http://www.play-it-safe.net/ (Accessed: 26 March 2016).

Primalsecurity (2016) *I'm Cuckoo for Malware – with a spice of Reverse Engineering*. Available at: http://www.primalsecurity.net/im-cuckoo-for-malware-with-a-spice-of-reverse-engineering/ (Accessed: 9 April 2016).

Purcell, D. M. & Lang, S. D. (2008) 'Forensic artifacts of microsoft windows vista system', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5075 LNCS, pp. 304–319. doi: 10.1007/978-3-540-69304-8_31.

Python (2016) *Python*. Available at: https://www.python.org/download/releases/2.7/ (Accessed: 10 April 2016).

RFC 3227 - Guidelines for Evidence Collection and Archiving (no date). Available at: http://www.rfc-base.org/rfc-3227.html (Accessed: 25 March 2016).

Robin, J. S. & Irvine, C. E. (2000) 'Analysis of the Intel Pentium's ability to support a secure virtual machine monitor'. USENIX Association, p. 10. Available at: http://dl.acm.org/citation.cfm?id=1251306.1251316 (Accessed: 25 March 2016).

RSA (2013) 'The Cyber Espionage Blueprint'.

Runwal, N., Low, R. M. & Stamp, M. (2012) 'Opcode graph similarity and metamorphic detection', *Journal in Computer Virology*, 8(1-2), pp. 37–52. doi: 10.1007/s11416-012-0160-5.

Russinovich, M., Cogswell, B. (no date) *Windows Sysinternals: Documentation, downloads and additional resources*. Available at: https://technet.microsoft.com/el-gr/sysinternals (Accessed: 25 March 2016).

Rutkowska, J. (2004) 'Red Pill... or how to detect VMM using (almost) one CPU instruction.'

Sandbox, C. (2016) *Cuckoo Sandbox*. Available at: http://docs.cuckoosandbox.org/en/latest/installation/ (Accessed: 9 April 2016).

SANS (2016) *SANS*. Available at: https://www.sans.org/ (Accessed: 10 April 2016).

Santos, I., Brezo, F., Nieves, J., Penya, Y. K., Sanz, B., Laorden, C. & Bringas, P. G. (2010) *Engineering Secure Software and Systems*. Edited by F. Massacci, D. Wallach, & N. Zannone. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science). doi: 10.1007/978-3-642-11747-3.

Schultz, M. G., Eskin, E., Zadok, E. & Stolfo, S. J. (2001) 'Data Mining Methods for Detection of New Malicious Executables'. IEEE Computer Society, p. 38. Available at: http://dl.acm.org/citation.cfm?id=882495.884439 (Accessed: 27 March 2016).

Schultz, M. G., Eskin, E., Zadok, F. & Stolfo, S. J. (2001) 'Data mining methods for detection of new malicious executables', in *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*. IEEE Comput. Soc, pp. 38–49. doi: 10.1109/SECPRI.2001.924286.

Schuster, A. (2006) 'Searching for processes and threads in Microsoft Windows memory dumps', *Digital Investigation*, 3, pp. 10–16. doi: 10.1016/j.diin.2006.06.010.

Shabtai, A., Moskovitch, R., Feher, C., Dolev, S. & Elovici, Y. (2012) 'Detecting unknown malicious code by applying classification techniques on OpCode patterns', *Security Informatics*. Springer Berlin Heidelberg, 1(1), p. 1. doi: 10.1186/2190-8532-1-1.

Shanks, W. (2014) 'Enhancing incident response through forensic, memory analysis and malware sandboxing techniques'. SANS Institute, p. 39.

Sharif, M., Lanzi, A., Giffin, J. & Lee, W. (2008) 'Impeding Malware Analysis Using Conditional Code Obfuscation', *Informatica*.

Shipley, T. G. (2007) 'Collecting Legally Defensible Online Evidence ':

Shukla, J. B., Singh, G., Shukla, P. & Tripathi, A. (2014) 'Modeling and analysis of the effects of antivirus software on an infected computer network', *Applied Mathematics and Computation*, 227, pp. 11–18. doi: 10.1016/j.amc.2013.10.091.

Sikorski, M. & Honig, A. (2012) *PRACTICAL MALWARE ANALYSIS:The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.

Ssdeep (2016) *Ssdeep*. Available at: http://ssdeep.sourceforge.net/ (Accessed: 10 April 2016).

Stormo, J. (2013) 'Analysis of Windows 8 Registry Artifacts'. Available at: http://scholarworks.uno.edu/td/1779/.

Sun, W.-C. & Chen, Y.-M. (2009) 'A rough set approach for automatic key attributes identification of zero-day polymorphic worms', *Expert Systems with Applications*, 36(3), pp. 4672–4679. doi: 10.1016/j.eswa.2008.06.037.

Symantec (2009a) *Common loading points for viruses, worms, and Trojan horse programs*. Available at: https://support.symantec.com/en_US/article.TECH99331.html (Accessed: 2 February 2016).

Symantec (2009b) *Most common registry key to check while dealing with Virus issue*, *Symantec*. Available at: http://www.symantec.com/connect/articles/most-common-registry-key-check-while-dealing-virus-issue (Accessed: 27 March 2016).

Symantec (2010) *How Symantec Antivirus system detects viruses | Symantec Connect*. Available at: http://www.symantec.com/connect/articles/how-symantec-antivirus-system-detects-viruses (Accessed: 8 May 2016).

Symantec (2014) *HTTP Trojan Bayrob Activity: Attack Signature*, *symantec*. Available at: https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=22970 (Accessed: 27 March 2016).

Symantec (2015) *Internet Security Threat Report 2015*, *Symantec*. Available at: https://www.symantec.com/security_response/publications/threatreport.jsp (Accessed: 25 March 2016).

Szor, P. (2005) 'The Art of Computer Virus Research and Defense'. Addison-Wesley Professional. Available at: http://dl.acm.org/citation.cfm?id=1050957 (Accessed: 25 March 2016).

Tech Anarchy (2016) *Cuckoo - ESXi*. Available at: https://techanarchy.net/lab/cuckoo-esxi/ (Accessed: 9 April 2016).

Thomas, S., Sherly, K. K. & Dija, S. (2013) 'Extraction of memory forensic artifacts from windows 7 RAM image', in *2013 IEEE CONFERENCE ON INFORMATION AND*

COMMUNICATION TECHNOLOGIES. IEEE, pp. 937–942. doi: 10.1109/CICT.2013.6558230.

Trendmicro (2015a) *BKDR_VAWTRAK.DOKR - Threat Encyclopedia*, *trendmicro*. Available at: http://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/bkdr_vawtrak.dokr (Accessed: 27 March 2016).

Trendmicro (2015b) *RANSOM_CRYPAURA.SVF - Threat Encyclopedia*, *trendmicro*. Available at: http://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/ransom_crypaura.svf (Accessed: 27 March 2016).

Trendmicro (2015c) *RTKT_DOTTUN.VTH - Threat Encyclopedia*. Available at: http://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/rtkt_dottun.vth (Accessed: 27 March 2016).

U.S. Legal, I. D. (no date) *Digital Evidence and Forensics | National Institute of Justice*. Available at: http://www.nij.gov/topics/forensics/evidence/digital/Pages/welcome.aspx (Accessed: 25 March 2016).

Vespignani, A. (2005) 'Complex networks: Behind enemy lines', *Nature Physics*, 1(3), pp. 135–136. doi: 10.1038/nphys183.

VirusSign (2016) *VirusSign*. Available at: http://www.virussign.com/ (Accessed: 10 April 2016).

VirusTotal (no date) *VirusTotal - Free Online Virus, Malware and URL Scanner*. Available at: https://www.virustotal.com/ (Accessed: 25 March 2016).

Vlachos, V., Ilioudis, C. & Papanikolaou, A. (2012) *Global Security, Safety and Sustainability & e-Democracy*, *7th ICGS3 -- International Conference in Global Security, Safety and Sustainability / 4th e-Democracy Joint Conferences 2011*. Edited by C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-Nemrat. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering). doi: 10.1007/978-3-642-33448-1.

VMware (2016) *VMware Virtualization for Desktop & Server, Application, Public & Hybrid Clouds*. Available at: http://www.vmware.com/ (Accessed: 25 March 2016).

Volatility (no date) *Volatility*. Available at: http://www.volatilityfoundation.org/ (Accessed: 25 March 2016).

VxHeaven (2016) *VX Heaven*. Available at: http://vxheaven.org/ (Accessed: 10 April 2016).

Wang, C. (2001) 'A security architecture for survivability mechanisms'. University of Virginia. Available at: http://dl.acm.org/citation.cfm?id=933240 (Accessed: 25 March 2016).

Wikipedia (2016) *Δούρειος Ίππος*. Available at: https://el.wikipedia.org/wiki/%CE%94%CE%BF%CF%8D%CF%81%CE%B5%CE%B9%CE%BF%CF%82_%CE%8A%CF%80%CF%80%CE%BF%CF%82_(%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AD%CF%82) (Accessed: 10 April 2016).

Willems, C., Holz, T. & Freiling, F. (2007) 'Toward Automated Dynamic Malware Analysis Using CWSandbox', *IEEE Security and Privacy Magazine*. IEEE Educational Activities Department, 5(2), pp. 32–39. doi: 10.1109/MSP.2007.45.

Williams, J. (2011) 'ACPO Good Practice Guide for Digital Evidence', p. 41. Available at: http://www.acpo.police.uk/documents/crime/2014/Revised Good Practice Guide for Digital Evidence_Vers 5_Oct 2011_Website.pdf.

Yara (2016) *Yara*. Available at: https://github.com/plusvic/yara/releases/tag/v3.4.0.

Ye, Y., Wang, D., Li, T., Ye, D. & Jiang, Q. (2008) 'An intelligent PE-malware detection system based on association mining', *Journal in Computer Virology*, 4(4), pp. 323–334. doi: 10.1007/s11416-008-0082-4.

Zeltser (2016) *Free Automated Malware Analysis Sandboxes and Services*. Available at: https://zeltser.com/automated-malware-analysis/ (Accessed: 25 March 2016).

Zeltser, L. (2016) *Lenny Zeltser*. Available at: https://zeltser.com/ (Accessed: 10 April 2016).

Zou, C. C., Towsley, D. & Gong, W. (2007) 'Modeling and Simulation Study of the Propagation and Defense of Internet E-mail Worms', *IEEE Transactions on Dependable and Secure Computing*. IEEE Computer Society Press, 5(2), pp. 105–118.