ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΘΕΜΑ: Κακόβουλο Λόγισμικό (Malware): Ανάλυση και Ανίχνευση

• Ονομ/νυμο: Καραμάνης Παναγιώτης Μ.Τ.Ε 1209

Επιβλέπων καθηγητής: Χ. Ξενάκης

Ακαδημαϊκό Έτος 2012-2013

Περιεχόμενα

Περίληψη

Στην παρούσα διπλωματική εργασία θα δείξουμε πώς μπορούμε να αναλύσουμε κακόβουλα προγράμματα, σαν και αυτά που καθημερινά μολύνουν πολλούς χρήστες σε όλο τον κόσμο. Πιο συγκεκριμένα θα δούμε πώς εξαπλώνεται ένα κακόβουλο πρόγραμμα, με ποια κριτήρια μπορούμε να χαρακτηρίσουμε ένα πρόγραμμα ως κακόβουλο, καθώς και ποιος είναι ο σκοπός του. Επίσης από τη στιγμή που έχει εισέλθει στο σύστημα τι ενέργειες πραγματοποιεί κατά το στάδιο της εκτέλεσης του και ακόμα υπάρχουν τεχνικές που χρησιμοποιεί με σκοπό να αποκρύψει τη δράση του ή να δυσκολέψει κατά το στάδιο της ανάλυσής του και αν ναι ποιες είναι αυτές. Στην συνέχεια της εργασίας θα απαντήσουμε στα παραπάνω ερωτήματα καθώς και σε πολλά άλλα τα οποία είναι σημαντικά για την καλύτερη κατανόηση του θέματος. Τέλος θα δείξουμε με ποιο τρόπο μπορούμε να αναλύσουμε ένα κακόβουλο πρόγραμμα καθώς και ποια εργαλεία θα χρησιμοποιήσουμε με στόχο να είμαστε σε θέση να μπορέσουμε να εξηγήσουμε τι ακριβώς θα κάνει το συγκεκριμένο κακόβουλο πρόγραμμα.

1. Κακόβουλο λογισμικό

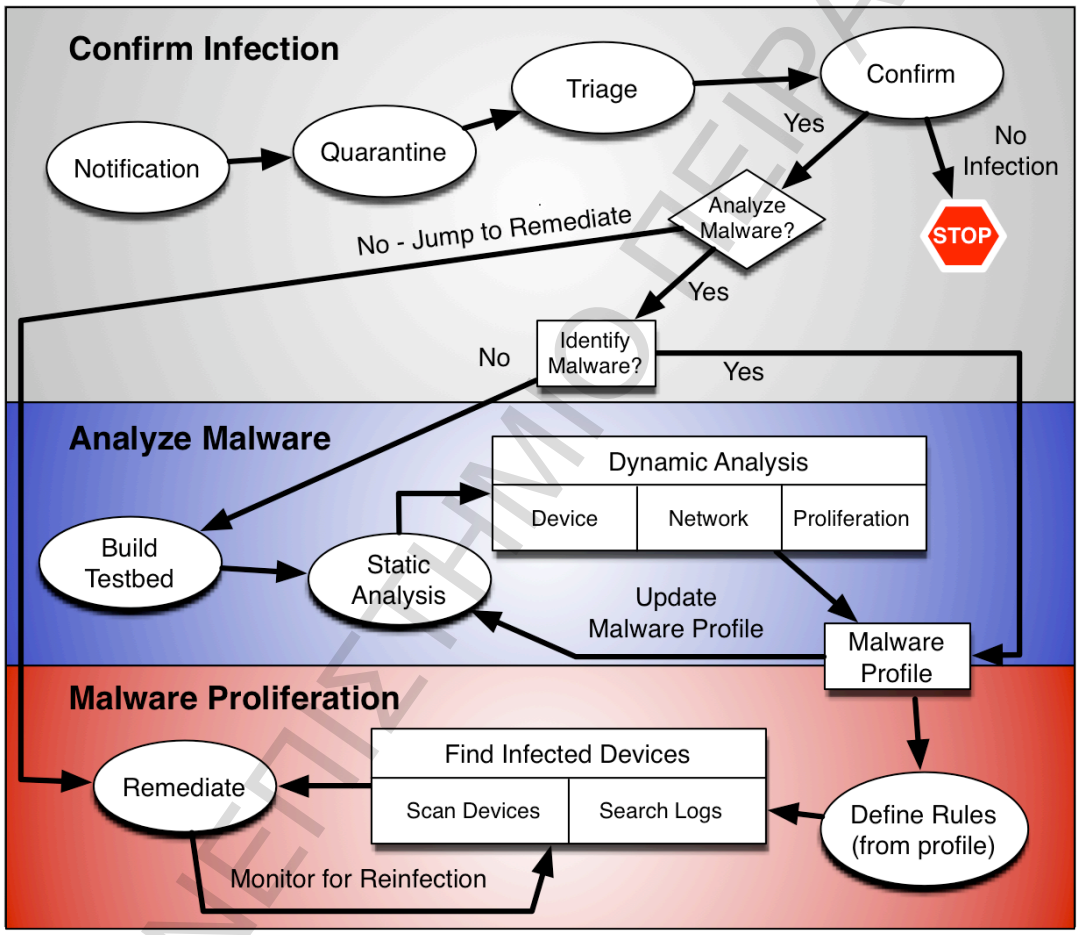
Το κακόβουλο λογισμικό ή malware (από το malicious software) είναι λογισμικό που χρησιμοποιείται για να εμποδίσει τη λειτουργία των υπολογιστών, να συλλέξει ευαίσθητες πληροφορίες ή και ακόμα να αποκτήσει πρόσβαση στον υπολογιστή μας. Μπορεί να εφαρμοστεί με τη μορφή κώδικα, με τη μορφή κάποιου script ή κάποιου ανοικτού περιεχομένου (active content). Πρόκειται για ένα γενικό όρο που χρησιμοποιείται για να αναφερθεί σε ποικίλες μορφές εχθρικού λογισμικού ή λογισμικού παρείσφρησης (intrusive software). Το πρώτο κακόβουλο πρόγραμμα δεν δημιουργήθηκε για PC αλλά για συστήματα που τρέχαν UNIX. Πιο συγκεκριμένα το όνομα του ήταν Internet Worm, εμφανίστηκε το 1988 και μόλυνε συστήματα που τρέχαν SunOS και VAX BSD. Διαδόθηκε εκμεταλλευόμενο τις τρύπες ασφαλείας στα προγράμματα κεντρικών υπολογιστικών δικτύων και ξεκίνησε ως χωριστή διαδικασία. Η τεχνική αυτή χρησιμοποιείται μέχρι και σήμερα. Σήμερα το ζήτημα του κακόβουλου λογισμικού έχει πάρει τεράστιες διαστάσεις. Κάποτε τα κακόβουλα προγράμματα που γράφονταν ήταν ιδιαίτερα απλά ή έστω έκαναν την εκτέλεσή τους προφανέστατη. Σήμερα έχουν φτάσει στο σημείο να θεωρούνται ως κρυφά επιθετικά όπλα, με γεωπολιτικά κίνητρα πίσω από την κατασκευή τους.

Ένα τέτοιο παράδειγμα αποτελεί και το Stuxnet το οποίο πρόκειται για το πιο περίπλοκο και εξελιγμένο malware που έχει κατασκευαστεί ποτέ και ο στόχος του ήταν οι εγκαταστάσεις απεμπλουτισμού ουρανίου στο Ιράν. Μισό megabyte κώδικα που μεταφέρθηκε με ένα USB stick, κατάφερε να κάνει μέσα σε μικρό χρονικό διάστημα όσα θα πετύχαινε και μια στρατιωτική επέμβαση, η οποία θα δημιουργούσε παγκόσμια αναταραχή και θα κόστιζε πολλές ανθρώπινες ζωές και ακόμα περισσότερα εκατομμύρια δολλάρια.

Ένα άλλο παράδειγμα είναι και ο Flame τον οποίο είχε ανακαλύψει η γνωστή εταιρεία ασφάλειας Kaspersky και όπως φαίνεται συνδέεται με το Stuxnet worm. Ο Flame είναι ένας ιός που σβήνει δεδομένα, ενώ δεν είναι ακόμα σαφές το τι βαθμό ζημιάς μπορεί να προκαλέσει σε συστήματα υποδομών. Μπορεί να συλλέξει δεδομένα, να αλλάξει τις ρυθμίσεις ενός υπολογιστή, να ανοίξει το μικρόφωνο για να κάνει εγγραφή των όσων λένε όσοι είναι μπροστά του, να τραβήξει screenshots και να δει όσα λέει κανείς στο chat. Σύμφωνα με τη Washington Post φέρετε πως οι ΗΠΑ και το Ισραήλ συνεργάστηκαν για τη δημιουργία του πολύ επικίνδυνου Flame malware αλλά και του Stuxnet, που μόλυνε χιλιάδες υπολογιστές στη Μέση Ανατολή, με σκοπό να πλήξουν το πρόγραμμα πυρηνικών όπλων του Ιράν**.** Πιο συγκεκριμένα στην όλη προσπάθεια συμμετείχαν η NSA, η CIA και ο Ισραηλινός στρατός και το malware δημιουργήθηκε τουλάχιστον πριν 5 χρόνια, με την απόρρητη κωδική ονομασία "Olympic Games"

(http://www.theverge.com/2012/6/19/3098080/us-israel-flame-malware-iran). Εκτιμάται ότι το Flame είναι η τρίτη σε σε κίνδυνο cyber απειλή, μετά το Stuxnet και το Duqu, αν και η κατάταξη δεν είναι ακόμα σαφής, μιας και δεν έχει κατανοηθεί πλήρως η λειτουργία του.

Ο κυβερνοπόλεμος είναι πια μια πραγματικότητα και τα malware/cyber weapons όπως το Stuxnet, το Flame και το Duqu κατασκευάζονται αρκετά συχνά και στοχεύουν συστήματα SCADA (Supervisory Control and Data Acquisition).



Σχήμα 1: Η αντιμετώπιση κακόβουλου λογισμικού δεν είναι μια απλή, ακολουθιακή διαδικασία. Υπάρχουν πολλά βήματα που εκτελούνται επαναληπτικά, ειδικά στο κομμάτι της ανάλυσης.

Γενικά το κακόβουλο λογισμικό έχει σχεδιαστεί προκειμένου να προκαλεί διακοπή ή άρνηση κάποιας υπηρεσίας, να συγκεντρώνει παράνομα πληροφορίες και ιδιωτικά/προσωπικά στοιχεία χρηστών, να επιτρέπει την απομακρυσμένη διαχείρηση των μολυσμένων μηχανημάτων κ.λπ. Τα επίσημα συμπτώματα από την

μόλυνση ενός υπολογιστή με κακόβουλο λογισμικό, είναι πιθανόν να είναι ένα από τα ακόλουθα.

• Καθυστέρηση στη λειτουργία (αυτό που λέμε το μηχάνημα σέρνεται). Αυτό συμβαίνει διότι το malware χρησιμοποιεί πολλούς από τους διαθέσιμους πόρους του συστήματος.

• Εμφάνιση νέων άγνωστων εκτελέσιμων αρχείων στο σύστημα. Αυτά παράγονται συνήθως από το ίδιο το κακόβουλο αρχείο μετά την εκτέλεσή του ή λαμβάνονται αυτόματα από το Διαδίκτυο.

• Εμφάνιση ανεπιθύμητης δικτυακής δραστηριότητας, όπως είναι η μεταφορά πακέτων από και προς άγνωστες διευθύνσεις IP ή και ιστοσελίδες τις οποίες ο χρήστης δεν επισκέπτεται καν.

• Αλλαγές σε ρυθμίσεις του συστήματος ή εγκατεστημένων προγραμμάτων, όπως π.χ., η αρχική σελίδα του web browser.

• Εμφάνιση παραθύρων pop-up καθώς και διαφημίσεων που πριν προσβληθεί το μηχάνημα από το malware δεν εμφανίζονταν.

Τώρα για να μπορέσουμε να χαρακτηρίσουμε ένα πρόγραμμα ως κακόβουλο θα πρέπει να εκτελεί μια από τις ακόλουθες λειτουργίες.

• Αλλαγή των ρυθμίσεων ενός άλλου προγράμματος.

• Αναπαραγωγή του εαυτού του τοπικά ή ακόμα και αποστολή του εαυτού του σε άλλους υπολογιστές στο δίκτυο, εννοείται πως αυτό γίνεται στο background και πως δεν υπάρχει καμιά περίπτωση να εμφανιστεί στην οθόνη του θύματος.

• Παροχή απομακρυσμένης πρόσβασης στο σύστημα όπου βρίσκεται το κακόβουλο πρόγραμμα, από κάποιον επιτιθέμενο.

• Συγκέντρωση προσωπικών στοιχείων ή ευαίσθητων πληροφοριών (κωδικοί πρόσβασης, αριθμοί πιστωτικών καρτών κ.λπ) καθώς και αποστολή τους σε απομακρυσμένη τοποθεσία.

• Άνοιγμα ενός ή περισσοτέρων ports στο σύστημα και αναμονή για συνδέσεις σε αυτά.

• Καταγραφή όσων πληκτρολογούνται στο σύστημα και αποστολή των πληροφοριών σε απομακρυσμένο κακόβουλο χρήστη.

• Σύνδεση σε απομακρυσμένους υπολογιστές, χωρίς την άδεια του χρήστη.

• Αυτόματη λήψη και εκτέλεση αρχείων από το Διαδίκτυο.

• Εισαγωγή κώδικα (code injection) σε άλλα προγράμματα ή ακόμα και διεργασίες.

• Τροποποίηση των ρυθμίσεων ασφαλείας του συστήματος.

• Αλλαγή των παραμέτρων του registry σε λειτουργικά συστήματα (Windows, MacOS κ.λπ). Οι αλλαγές αυτές συνήθως αφορούν στην αυτόματη εκτέλεση προγραμμάτων κατά την εκκίνηση του λειτουργικού.

Υπάρχουν πολλές κατηγορίες κακόβουλων προγραμμάτων, πρωτού όμως αναφέρουμε τις πιο σημαντικές από αυτές θα πρέπει να αναφέρουμε ότι συχνα ο μέσως χρήστης αποκαλεί «ιό» ή αλλιώς «virus» οτιδήποτε θυμίζει κακόβουλο λογισμικό. Στην πραγματικότητα όμως ένας ιός αποτελεί απλά μία μόνο κατηγορία κακόβουλου λογισμικού, από τις πολλές που υπάρχουν.

Ιός (Virus)

Είναι ένα πρόγραμμα που τρέχει στον υπολογιστή μας. Σκοπός του ιού είναι να βλάψει το σύστημά μας. Η διαφοροποίηση εδώ με τα υπόλοιπα, είναι ότι ο ιός έχει την ιδιότητα να μεταδίδεται όπου είναι δυνατόν. Συνυπάρχει σε πολλά αρχεία –κρυμμένο- αποτελεί φαινομενικά αθώο λογισμικό και είναι έτοιμο να μεταδοθεί σε άλλα συστήματα μέσω αυτών, αν ένα από αυτά τα αρχεία μεταφερθεί σε άλλα συστήματα, όπου και εκεί κάνει τα ίδια (αφού βέβαια προκαλέσει και κάποια ζημιά που είναι και ο στόχος του αυτός -ανάλογα τον ιό). Ο τρόπος εξάπλωσής του μοιάζει με αυτόν μιας ασθένειας και απο εκεί παίρνει και το όνομά του.

Σκουλήκι-Αναπαραγωγός (Worm)

Είναι συνήθως ένα μικρό, αυτόνομο πρόγραμμα το οποίο έχει την ικανότητα να αυτό-αναπαράγεται και να μεταφέρεται από υπολογιστή σε υπολογιστή και με αυτό το τρόπο μολύνει τους υπολογιστές που είναι συνδεμένοι σε κάποιο δίκτυο. Αν ένας υπολογιστής έχει προσβληθεί από

ένα worm και συνδεθεί σε ένα δίκτυο, θα μολυνθούν και όλοι οι υπόλοιποι υπολογιστές που είναι συνδεμένοι σε αυτό. Το worm απλά επιβαρύνει το δίκτυο φορτώνοντάς το με άχρηστη δραστηριότητα. Η διαφορά του με τον ιό είναι πως κατά κανόνα δεν αλλοιώνει άλλα προγράμματα ή αρχεία ενώ δεν χρειάζεται και κάποιο πρόγραμμα-ξενιστή.

Δούρειος Ίππος (Trojan Horse)

Το όνομά του ως γνωστών προέρχεται από την ελληνική μυθολογία. Πρόκειται για κατασκευή εμπνευσμένη από τον Οδυσσέα με σκοπό να παραπλανήσει τους Τρώες. Σήμερα η έκφραση «Δούρειος Ίππος» υποδηλώνει είσοδο με δόλο, τέχνασμα και πονηριά (http://el.wikipedia.org/wiki/Δούρειος_Ίππος). Στην επιστήμη των υπολογιστών είναι προγράμματα που μιμούνται τη χρησιμότητα του Δούρειου Ίππου: Φαινομενικά δείχνουν αθώα και δεν αποκλείεται να εκτελούν και κάποια εργασία ωστόσω εμπεριέχει κάποιο κρυφό κακόβουλο μηχανισμό. Ένας Δούρειος Ίππος μπορεί να έχει από εντελώς ακίνδυνη δράση, έως πολύ επικίνδυνη (διαγραφή αρχείων, υποκλοπή προσωπικών δεδομένων κ.λπ).

Spyware

Είναι ένα πρόγραμμα που χρησιμοποιείται για να παρακολουθεί τις δραστηριότητες του μολυσμένου υπολογιστή.
Ενας υπολογιστής που έχει κολλήσει spyware, παρακολουθείται από κάποιο τρίτο άτομο και πιο συγκεκριμένα, αυτό το τρίτο άτομο μπορεί να δει π.χ τι πληκτρολογούμε (usernames, passwords σε sites), κλπ.

Adwares

Είναι τα προγράμματα που κολλάμε και πολύ απλά μας φέρνουν με ύπουλο τρόπο διαφημίσεις στον υπολογιστή

μας, χωρίς την έγκρισή μας. Χρησιμοποιούν το bandwidth της σύνδεσής μας στο internet, περιορίζοντάς μας σε πιο χαμηλές ταχύτητες και εμφανίζονται συνέχεια διαφημίσεις, χωρίς να μπορούμε να κάνουμε κάτι.

Backdoor

Πρόκειται για πρόγραμμα που επιτρέπει στον κακόβουλο χρήστη την απομακρυσμένη πρόσβαση ή και διαχείρηση του υπολογιστή όπου βρίσκεται εγκατεστημένο.

Rootkit

Πρόκειται για λογισμικό που χρησιμοποιεί προχωρημένες τεχνικές απόκρυψης, προκειμένου να εισχωρήσει όσο το δυνατόν βαθύτερα στο λειτουργικό σύστημα και ταυτόχρονα να μην είναι δυνατή η ανακάλυψή του. Ένα από τα πιο γνωστά rootkits ήταν αυτό που τοποθετούσε κάποτε η γνωστή εταιρία Sony, στα μουσικά CD της. Η δικαιολογία αφορούσε στην αποτροπή της πειρατείας.

2. Ανάλυση κακόβουλου λογισμικού

Η ανάλυση ενός άγνωστου δείγματος κακόβουλου λογισμικού, αυτό που ονομάζουμε malware sample, είναι ουσιαστικά η διαδικασία μελέτης καθώς και η ανάλυση του συγκεκριμένου προγράμματος στα κύρια συστατικά του. Αυτό γίνεται με σκοπό να είμαστε σε θέση να δώσουμε απαντήσεις σε κρίσιμα ερωτήματα που ανακύπτουν. Στα ζητήματα που αφορούν ένα malware, συγκαταλέγονται τα ακόλουθα.

• Ποιος είναι ο σκοπός του malware;

• Πώς εισέρχεται σε ένα σύστημα;

• Τι ενέργειες πραγματοποιεί μετά την εκτέλεσή του;

• Πώς μπορώ να απαλλαχτώ από αυτό;

• Τί προσπαθεί να υποκλέψει ή να αλλοιώσει

• Πόσο καιρό βρίσκεται μέσα στο σύστημα

• Πώς μεταφέρεται αν μεταφέρεται από σύστημα σε σύστημα;

• Πώς μπορώ να το εντοπίσω σε άλλα μολυσμένα μηχανήματα;

• Ποια είναι τα σημάδια στο δίκτυο/υπολογιστές θύματα, που μαρτυρούν τη δράση του;

• Πότε και πώς δημιουργήθηκε το malware; Βασίζεται σε άλλα γνωστά malware ή βιβλιοθήκες;

• Χρησιμοποιεί τεχνικές κατά της ανάλυσής του; Αν ναι, ποιες;

• Πώς μπορώ να το εμποδίσω από το να εμφανιστεί κάποιο παρόμοιο στο μέλλον;

Ο στόχος του malware analysis είναι ουσιαστικά η συγκέντρωση επαρκών πληροφοριών, ώστε να μπορέσουμε να αντιμετωπίσουμε το συγκεκριμένο κακόβουλο πρόγραμμα. Οι πληροφορίες αυτές αποτελούν ορισμένα μοναδικά χαρακτηριστικά του malware και με βάση αυτά τα μοναδικά χαρακτηριστικά μπορούμε να φτιάξουμε μια υπογραφή για το υπό εξέταση κακόβουλο πρόγραμμα. Τώρα με αυτή την υπογραφή μια εταιρία ασφάλειας θα μπορούσε να φτιάξει έναν κανόνα που θα ενσωμάτωνε σε ένα antivirus ή σε ένα σύστημα ανίχνευσης επιθέσεων (Intrusion Detection System, IDS). Έτσι, από εκεί και πέρα το συγκεκριμένο κακόβουλο λογισμικό θα ανιχνευόταν επακριβώς και θα αντιμετωπιζόταν αποτελεσματικά.

Η διαδικασία της ανάλυσης είναι ιδιαίτερα πολύπλοκη και χρονοβόρα ακόμα και για έμπειρους αναλυτές σε θέματα ασφάλειας. Συνεπώς το παραπάνω εγχείρημα δεν είναι καθόλου απλό. Ένας από τους λόγους είναι πως το κακόβουλο λογισμικό εξελίσσεται διαρκώς και μάλιστα ένα αρκετά μεγάλο μέρος του κέρδους επενδύεται στην περαιτέρω εξέλιξή του.

2.1 Οικονομικά ωφέλη του malware

Τα κέρδη από το κακόβουλο λογισμικό κάθε άλλο παρά λίγα είναι. Μια κατηγορία με πολλά κέρδη αποτελούν τα botnets και πολλά από αυτά διατίθονται προς πώληση στην underground αγορά. Για παράδειγμα ένα σχετικά νέο botnet είναι το Thor το οποίο είναι ένα αποκεντρωμένο P2P botnet, γραμμένο σε C/C++ και αναπτύχθηκε από τον "TheGrimReap3r" και είναι σχεδόν έτοιμο προς πώληση. Το Thor χρησιμοποιεί DLL injection, IAT hooking, ring3 rootkit. Ένα ακόμα ενδιαφέρον χαρακτηριστικό είναι ότι διαθέτει δικό του σύστημα, έτσι μπορούμε να γράψουμε τα δικά μας modules με το εύκολο σύστημα API. Θα περιλαμβάνει peer to peer επικοινωνία και επίσης χρησιμοποιεί 256-AES κρυπτογράφηση με τυχαίο κλειδί σε κάθε εκκίνηση. Λειτουργεί σε περιβάλλον Win 2000+, Win XP SP0/SP1/SP2/SP3, Win Vista SP0/SP1/SP2, Win 7 SP0/SP1 και υποστηρίζει x86 και x64 συστήματα. Οι προγραμματιστές του Thor θα πωλούν αυτό Botnet σε ανοιχτά underground market και διάφορα hacking φόρουμ με αρχική τιμή πώλησης τα 8000$. Βέβαια υπάρχουν και αρκετά πιο φθηνά botnets με τιμές που ξεκινούν από τα 59$.

Στην συνέχεια θα δούμε τι ενέργειες μπορεί να πραγματοποιήσει καθώς και τι οφέλη έχει ο botmaster. Ορισμένα από αυτά είναι τα εξής:

• Πρώτη περίπτωση είναι η διενέργεια επιθέσεων DDoS. Ο botmaster δίνει εντολή και εκατοντάδες ή ακόμα και χιλιάδες υπολογιστές αποστέλουν πολλαπλά αιτήματα σε συγκεκριμένους web servers με αποτέλεσμα να τους θέσουν εκτός λειτουργίας. Σύμφωνα με την Prolexic, οι επιθέσεις τύπου Distributed Denial-of-Service έχουν συνήθως ως κίνητρα εκβιασμούς, χακτιβισμό, πειράματα των χάκερ, θυμωμένους πελάτες ή υπαλλήλους και αντίποινα από τους ανταγωνιστές.

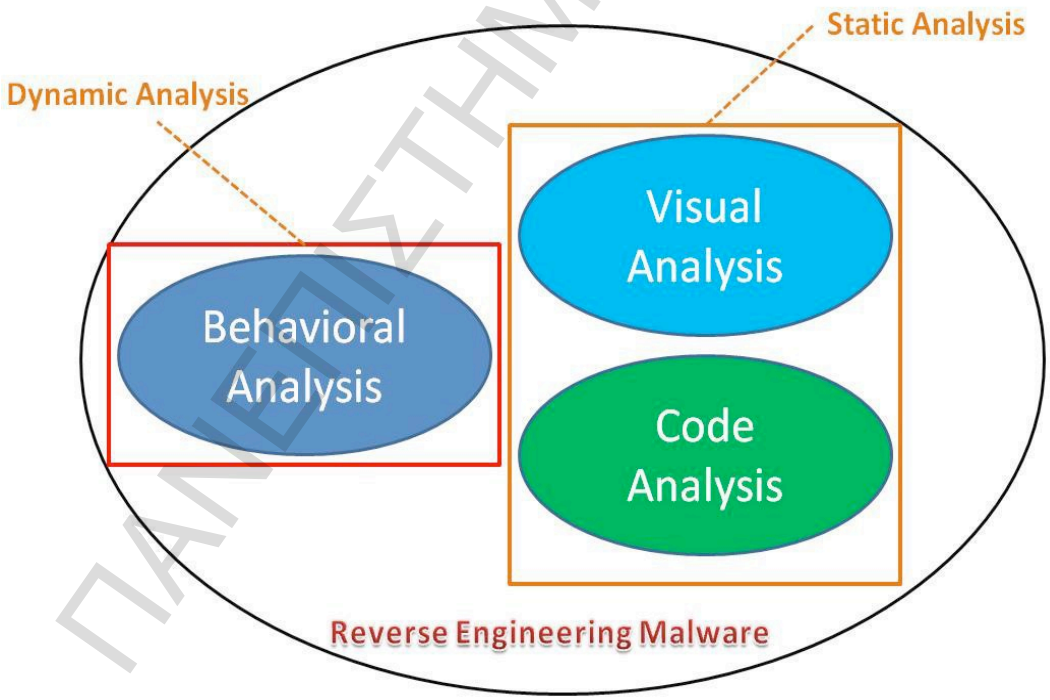
• Μια συνηθισμένη χρήση των botnets είναι και η αποστολή SPAM. Πρόκειται για διαφημιστικά mail και μάλιστα υπολογίζεται ότι περισσότερα από τα μισά που στέλνονται προέρχονται από υπολογιστές zombies.

• Μια άλλη παράνομη δραστηριότητα που σχετίζεται με τα botnets είναι το phishing. Είναι παραπλανητικά mail που στέλνονται με σκοπό να ξεγελάσουν το παραλήπτη πείθοντας τον ότι προέρχονται από μια σημαντική υπηρεσία όπως μια τράπεζα με τελικό στόχο να υποκλέψουν τα στοιχεία του. Επειδή αυτές οι ιστοσελίδες όταν ανακαλυφθούν κλείνουν πολλοί από αυτούς τους κακόβουλους χρησιμοποιούν την τεχνολογία fast flux των botnets η οποία εμποδίζει να ανακαλυφθουν οι επιθέσεις τύπου phishing.

Ένα άλλο γνωστό botnet ήταν το Shadow το οποίο πωλούταν προς 37000 δολλάρια και το οποίο διέθετε παραπάνω από 150.000 zombies. Διαχειριστής του ήταν ένας 19χρονος Ολλανδός και ο μικρότερος αδερφός του ηλικίας 16 χρονών. Συνελήφθησαν στη προσπάθεια τους να πουλήσουν το botnet σε ένα βραζιλιάνο ηλικίας 35 ετών.

2.2 Φάσεις του κακόβουλου λογισμικού

Η διαδικασία της ανάλυσης κακόβουλων προγραμμάτων αποτελείται κυρίως από δύο φάσεις.



Σχήμα 2.: Φάσεις του κακόβουλου λογισμικού

1. **Ανάλυση συμπεριφοράς.** Εξετάζουμε και καταγράφουμε το πώς αλληλεπιδρά το κακόβουλο δείγμα με το περιβάλλον του (με τα αρχεία που υπάρχουν στο σύστημα, το registry, τις συνδέσεις στο δίκτυο, άλλες ενεργές διεργασίες του συστήματος κ.ο.κ).
2. **Ανάλυση κώδικα.** Προσπαθούμε με την αντίστροφη μεταγλώττιση (reverse engineer) να κατανοήσουμε πώς λειτουργεί το εκτελέσιμο αρχείο καθώς και τη λειτουργικότητά του, εξετάζοντας τον κώδικα Assembly. Η διαδικασία αυτή περιλαμβάνει τη χρησιμοποίηση ενός disassembler, ενός debugger και ενός decompiler. Η Assembly είναι μια γλώσσα πολύ χαμηλού επιπέδου ακριβώς ένα επίπεδο πάνω από τη καθαρή γλώσσα μηχανής που καταλαβαίνει ο υπολογιστής.

Οι παραπάνω δυο φάσεις αντιπροσωπεύουν την ανάλυση malware σε υψηλό επίπεδο. Πιο πρακτικά οι δύο αυτές φάσεις μπορούν να ταξινομηθούν σε δύο κατηγορίες: της στατικής και της δυναμικής ανάλυσης.

Κάθε μια από αυτές έχει επιπλέον το βασικό αλλά και το προχωρημένο στάδιο. Συνολικά μιλάμε για τέσσερεις διαδικασίες.

3. Το εργαστήριο μας

Για τη διενέργεια της διαδικασίας ανάλυσης θα χρειατεί να δουλέψουμε σε εικονική μηχανή (Virtual Machine, VM). Τέτοιες μπορούμε πολύ εύκολα να κατασκευάσουμε με εφαρμογές όπως το VMware Workstation/Fusion ή το VirtualBox. Οι λόγοι για αυτήν την επιλογή είναι οι ακόλουθοι:

• Θα χρειαστεί σίγουρα να εκτελέσουμε κανονικά το malware sample στον υπολογιστή, οπότε μας βολεύει να το κάνουμε στα πλαίσια μιας απομονωμένης εικονικής μηχανής, χωρίς να διατρέξουμε κίνδυνο για το πραγματικό μας σύστημα.

• Είναι ιδιαίτερα χρήσιμη η δυνατότητα για λήψη στιγμιοτύπων (snapshots), που παρέχουν οι εφαρμογές διαχείρησης εικονικών μηχανών.

Ένα snapshot αποθηκεύει στην ουσία την τρέχουσα κατάσταση της εκάστοτε μηχανής στο δίσκο του αληθινού υπολογιστή με όλες τις ρυθμίσεις, τα αρχεία, τις ιδιότητές τις κ.λ.π. Έτσι μπορούμε σε μεταγενέστερο χρόνο να κάνουμε reset την κατάσταση της μηχανής και να επιστρέψουμε σε ένα snapshot της επιλογής μας. Επειδή όμως θα χρειατεί να εκτελούμε το malware αρκετά συχνά μας βολεύει πάρα πολύ να μπορούμε εύκολα να ξεκινάμε κάθε φορά από ένα αρχικό καθαρό στάδιο.

Πιο συγκεκριμένα θα χρειαστεί να δημιουργήσουμε μια εικονική μηχανή με το δωρεάν VirtualBox στην οποία θα εγκαταστήσουμε τα Windows XP SP3. Πέρα από το λειτουργικό θα πρέπει να κατεβάσουμε και να ενσωματώσουμε στο σύστημα τα ακόλουθα εργαλεία.

• **WinMD5Free**
http://www.winmd5.com

• **Strings v2.52**

http://technet.microsoft.com/en-us/sysinternals/bb897439

• **BinText 3.0.3**

http://www.mcafee.com/us/downloads/free-tools/bintext.aspx

• **PEiD v0.95**

http://www.softpedia.com/dyn-postdownload.php?p=4102&t=5&i=1

• **PE Detective v1.2.1.1**

http://www.ntcore.com/pedetective.php

•**Dependency Walker v2.2.6000**

http://www.dependencywalker.com

• **IDA Pro Advanced v5.5.0.925t**

http://thepiratebay.se/torrent/5235650/IDA_Pro_Advanced_v5.5.0.925t_and_Hex-Rays_v1.1.0.090909

Αφού έχουμε ετοιμάσει και κατεβάσει όλα τα παραπάνω εργαλεία μπορούμε να πάρουμε και το πρώτο snapshot της μηχανής. Για να το κάνουμε αυτό επιλέγουμε Machine ->Take Snapshot και δίνουμε ένα περιγραφικό όνομα όπως π.χ **Static Malware Analysis Lab**.

Με το εργαστήριό μας έτοιμο μένει μόνο να βρούμε ορισμένα malware samples, για να πραγματοποιήσουμε τη διαδικασία της στατικής ανάλυσης.

3.1 Honeypots

Για να πραγματοποιήσουμε τη διαδικασία της ανάλυσης θα πρέπει να πιάσουμε (να κάνουμε catch) τα δικά μας κακόβουλα προγράμματα όπως είναι π.χ τα worms, ή τα bots. Αυτό μπορεί να γίνει σχετικά εύκολα με το στήσιμο ενός malware honeypot. Το honeypot είναι ένα σύστημα που δεν προσπαθεί να αποτρέπει τους επιτιθέμενους αλλά αντίθετα προσπαθεί να τους έλκει. Συνεπώς η ιδιαίτερη αξία ενός honeypot δεν έγκειται στο ότι είναι ένα απόρθητο σύστημα αλλά αντιθέτως ο σκοπός του είναι να δεχθεί επιθέσεις και κάποια στιγμή να μην αντέξει και να πέσει. Ως εκ τούτου τα honeypots αποτελούν τρωτά (vulnerable) συστήματα που έχουν ως απότερο σκοπό να παραβιαστούν.

3.1.1 Είδη Honeypots

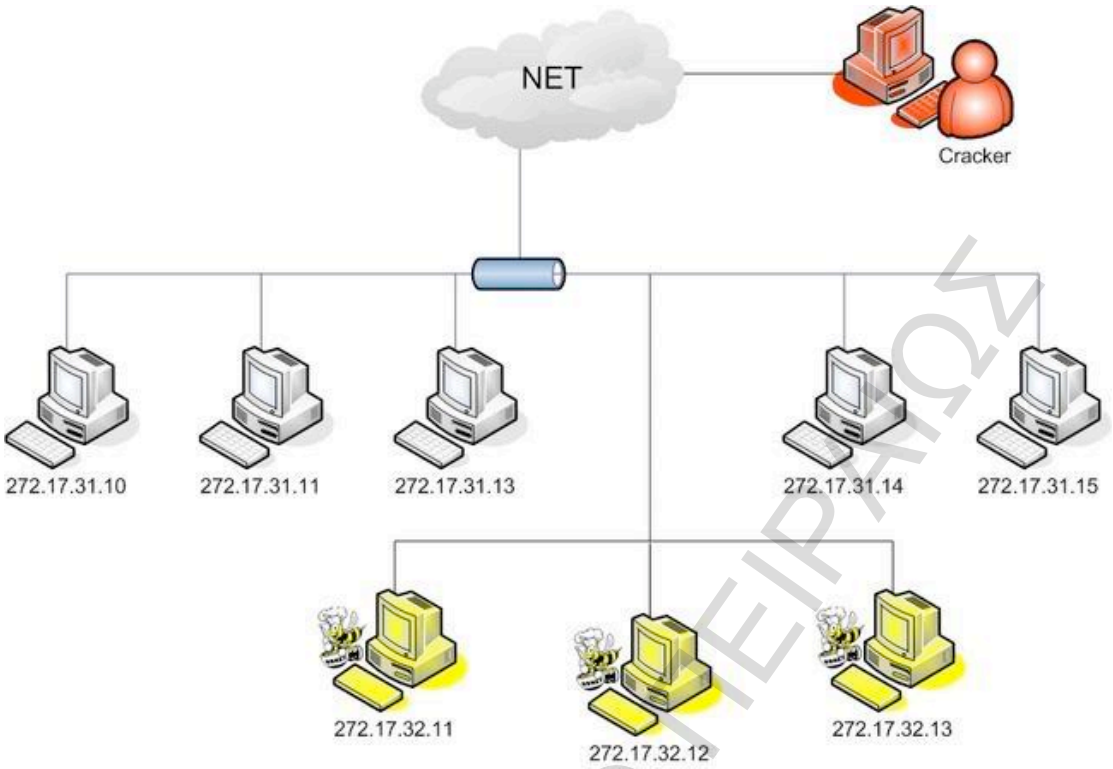
Μπορούμε να χωρίσουμε τα honeypots με βάση την επέκτασή τους και με βάση το επίπεδο συμμετοχής τους. Με βάση την επέκτασή τους τα honeypots μπορούν να χωριστούν σε:

• honeypots παραγωγής και

• ερευνητικά honeypots

Τα honeypots παραγωγής είναι εύκολα στη χρήση τους, συλλαμβάνουν ορισμένες μόνο πληροφορίες και χρησιμοποιούνται κυρίως από τις επιχειρήσεις ή τις εταιρίες. Τα honeypots παραγωγής επίσης τοποθετούνται μέσα στο δίκτυο παραγωγής με άλλους κεντρικούς υπολογιστές παραγωγής από μια εταιρεία ή έναν οργανισμό για να βελτιώσουν τη γενική κατάσταση ασφάλειάς τους. Συνήθως τα honeypots παραγωγής είναι χαμηλής αλληλεπίδρασης honeypots, τα οποία είναι ευκολότερο να επεκταθούν. Δίνουν τις λιγότερες πληροφορίες για τις επιθέσεις ή τους επιτιθέμενους σε σχέση πάντα με τα ερευνητικά honeypots.

Τα ερευνητικά honeypots χρησιμοποιούνται για να συγκεντρώσουν πληροφορίες σχετικές με τα κίνητρα και τις τακτικές των κακόβουλων hackers (blackhat) που στοχεύουν στα διάφορα δίκτυα. Η αξία αυτών των honeypots έγκειται στο γεγονός ότι χρησιμοποιούνται για να ερευνήσουν τις απειλές που οι οργανισμοί αντιμετωπίζουν και επίσης για να μάθουν πώς να προστατευτούν καλύτερα από τις υπάρχουσες απειλές. Έτσι λοιπόν από τις πληροφορίες που συγκεντρώνουν οι διαχειριστές της εκάστοτε εταιρίας μπορούν να πληροφορηθούν για τυχόν αδυναμίες που υπάρχουν στα κανονικά τους συστήματα καθώς επίσης και για τις τεχνικές που χρησιμοποιούν οι επιτιθέμενοι. Τα ερευνητικά honeypots είναι πολύπλοκα και δύσκολο να επεκταθούν και να συντηρηθούν. Όπως προδίδεται και από το ονομά τους είναι για ερευνητικούς σκοπούς και χρησιμοποιούνται κυρίως για στρατιωτικούς ή κυβερνητικούς οργανισμούς.



Σχήμα 3: Παράδειγμα τυπικής αρχιτεκτονικής ενός δικτύου, το οποίο ενσωματώνει honeypot. Όπως φαίνεται οι παγίδες τοποθετούνται ανάμεσα σε πραγματικά μηχανήματα.

3.1.2 Επίπεδο αλληλεπίδρασης

Πέρα από το διαχωρισμό με βάση το σκοπό που δημιουργούνται, τα honeypots χωρίζονται και με βάση το επίπεδο αλληλεπίδρασης. Τα επίπεδα αυτά είναι τρία:

• Χαμηλής αλληλεπίδρασης. Αυτά τα honeypots προσομοιώνουν μόνο υπηρεσίες που ζητούνται συχνά από τους επιτιθέμενους ώστε να αποκτήσουν πρόσβαση στο ίδιο το honeypot. Δεδομένου ότι καταναλώνουν σχετικά λίγους πόρους , μπορούμε να έχουμε πολλά εικονικά μηχανήματα στο κανονικό μας μηχάνημα. Τα εικονικά αυτά μηχανήματα έχουν μικρό χρόνο απόκρισης αφού είναι ελάχιστος ο κώδικας. Αυτό όμως μειώνει την ασφάλεια της εικονικής μηχανής. Ένα παράδειγμα τέτοιου honeypot είναι το γνωστό Honeyd.

• Μεσαίας αλληλεπίδρασης. Αυτά τα honeypots είναι πιο ανεπτυγμένα από τα προηγούμενα. Και αυτά με τη σειρά τους δεν τρέχουν κάποιο λειτουργικό. Ωστόσο παρέχουν περισσότερες υπηρεσίες και προσφέρουν περισσότερους στόχους στον επιτιθέμενο. Επίσης εκτελούν και ορισμένα scripts.

• Υψηλής αλληλεπίδρασης. Αυτά τα honeypots προσομοιώνουν τις δραστηριότητες των συστημάτων παραγωγής που φιλοξενούν ποικίλες υπηρεσίες και επομένως ο

επιτιθέμενος μπορεί να έχει πρόσβαση σε πολλές υπηρεσίες. Είναι ιδιαίτερα πολύπλοκα και δεν εξομοιώνουν κάποιες υπηρεσίες ή scripts αλλά ένα πλήρες λειτουργικό σύστημα. Σύμφωνα με μια πρόσφατη έρευνα που έγινε για τα honeypots υψηλής αλληλεπίδρασης ακόμα και όταν το honeypot καταληφθεί μπορεί να αποκατασταθεί γρηγορότερα. Γενικά τα honeypots υψηλής αλληλεπίδρασης παρέχουν υψηλότερη ασφάλεια αλλά η διατήρησή τους είναι ιδιαίτερα ακριβή. Σε περίπτωση που τα εικονικά μηχανήματα δεν είναι διαθέσιμα τότε το honeypot θα πρέπει να το στήσουμε αναγκαστικά σε κανονικό μηχάνημα και αυτό μπορεί να είναι ιδιαίτερα ακριβό. Ένα παράδειγμα υψηλής αλληλεπίδρασης honeypot είναι το Honeynet.

Πέρα από τα honeypots ένας δεύτερος τρόπος, πιο εύκολος και πιο γρήγορος είναι να βρούμε έτοιμα malware. Υπάρχουν διάφορες τέτοιες αποθήκες (repositories) malware, στα οποία σχεδόν καθημερινά ανεβαίνουν νέα δείγματα.

Ένα σχετικό αποθευτήριο (repository) το οποίο θα χρησιμοποιήσουμε και εμείς είναι το OffensiveComputing.net. Διαθέτει πάνω από πέντε εκατομμύρια malware samples και λειτουργεί από τη Georgia Tech.

Επισκεπτόμαστε λοιπόν τη σελίδα http://www.offensivecomputing.net και για να βρούμε ένα malware sample γράφουμε τη λέξη «Trojan» στη μπάρα αναζήτησης. Αμέσως μεταφερόμαστε στη πλατφόρμα Open Malware και βλέπουμε τα αποτελέσματα. Πρόκειται για αρχεία που εμπεριέχουν τη συγκεκριμένη λέξη. Θα κατεβάσουμε το πρώτο από αυτά τον δούρειο ίππο με την ονομασία **Mitglieder**. Δίνουμε λοιπόν τα στοιχεία του Google account μας και αποσυμπιέζουμε το αρχείο μέσα στην εικονική μηχανή που δημιουργήσαμε. Παρακάτω βλέπουμε τα στοιχεία αυτού του Trojan που μόλις κατεβάσαμε.

D5: `6cbbd4cf962649a4b51626ae8b603491`

HA1: `e2ac21026edb2dc3629d2c3214ec086b6e5499eb`

HA256: `d6dcfa69ef0e437fbcc60a1ea4f03019e4814fa90b789e0e80d5179022e2b1`

CID:	1293588068
riginal lename:	6cbbd4cf962649a4b51626ae8b603491.exe
lded:	2010-01-02 14:17:31.742921
/ Results:	**F-Prot:** W32/Mitglieder.BW (exact) **ClamAV:** Trojan.Proxy.Mitglieder-2 **BitDefender:** Trojan.Generic.551949
:her formation:	Virustotal ThreatExpert
ownload:	Download Sample - Authentication Required. Sample password is "infected"

4. Βασική στατική ανάλυση malware

Η στατική ανάλυση είναι συνήθως το πρώτο βήμα που εκτελείται κατά την εξέταση ενός κακόβουλου δείγματος. Περιγράφει στην ουσία, τη διαδικασία ανάλυσης του αρχείου και του κώδικά του, για την εξαγωγή συμπερασμάτων σχετικά με τη λειτουργία του. Το χαρακτηριστικό γνώρισμα της στατικής ανάλυσης είναι ότι κατά τη διάρκειά της δεν εκτελείται το κακόβουλο αρχείο. Γίνεται δηλαδή προσπάθεια να κατανοηθεί πώς ενεργεί, μόνο με τα στοιχεία που λαμβάνονται χωρίς αλληλεπίδραση του malware με το σύστημα. Όπως αναφέραμε, η στατική ανάλυση έχει δύο στάδια: το απλό (basic) και το προχωρημένο (advance).

Βήμα 1. Σάρωση του αρχείου από μηχανές antivirus

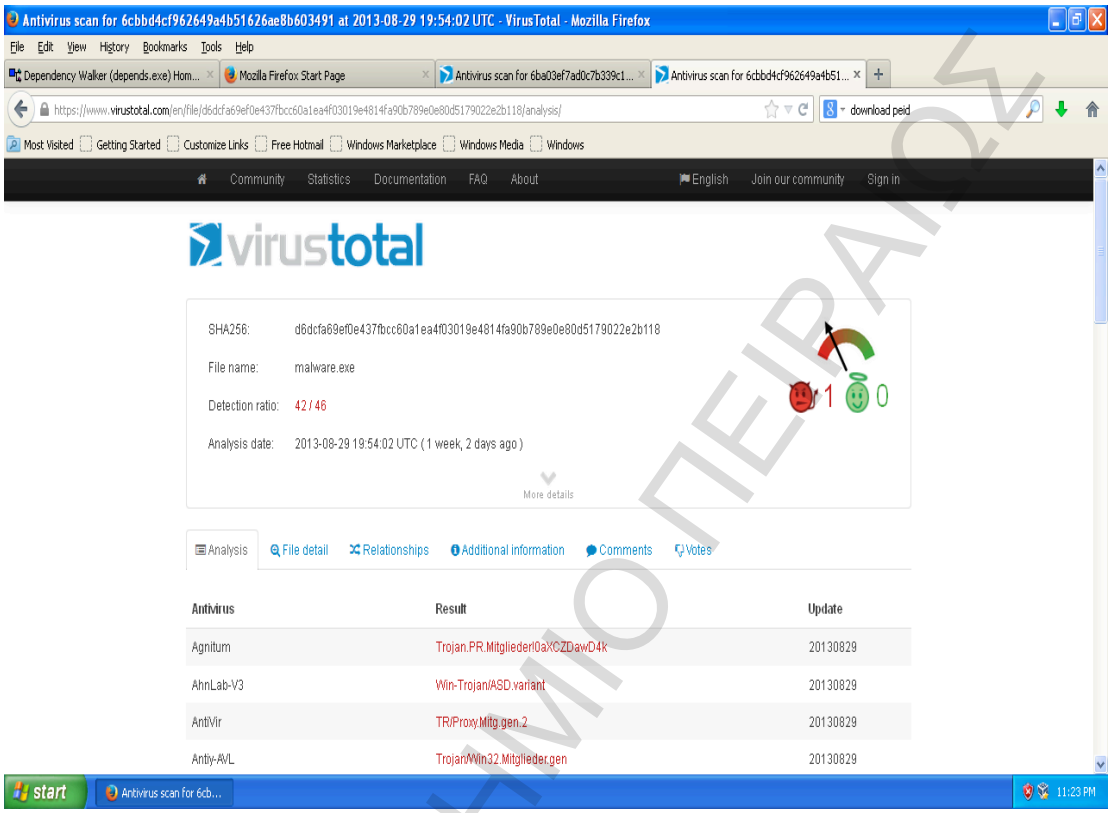
Όταν εκτελούμε τη στατική ανάλυση ένα χρήσιμο πρώτο βήμα είναι να περάσουμε το κακόβουλο αρχείο από διάφορες μηχανές antivirus για να δούμε αν έχει εκ των προτέρων αναγνωριστεί και καταχωρηθεί από κάποια εταιρεία ασφάλειας. Τα προγράμματα antivirus βασίζονται σε δύο τεχνικές:

• Η πρώτη έχει να κάνει με τις υπογραφές (signatures) των κακόβουλων προγραμμάτων. Αν λοιπόν υπάρχει μια τέτοια υπογραφή για το αρχείο μας, τότε η αναγνώριση από το antivirus θα είναι επιτυχής.

• Η δεύτερη τεχνική είναι η χρήση ευριστικών (heuristics) όπως ονομάζονται, μεθόδων. Δηλαδή, το antivirus ακόμα και αν δεν έχει την υπογραφή του malware, μπορεί να καταλάβει ότι πρόκειται για κάτι κακόβουλο παρατηρώντας τη συμπεριφορά του και συγκρίνοντας τη με ορισμένα γνωστά μοτίβα (pattern).

Ένα πρόβλημα εδώ είναι πως αυτοί που γράφουν malware μπορούν εύκολα να τροποποιήσουν το κώδικά με τέτοιο τρόπο ώστε να ξεγελούν τις μηχανές antivirus. Επίσης υπάρχει και η πιθανότητα ορισμένα σπάνια malware να μην μπορούν να ανιχνευθούν επειδή απλά δεν υπάρχει αντίστοιχη υπογραφή, στη βάση δεδομένων του όποιου antivirus. Τέλος αν και οι ευριστικές τεχνικές καταφέρνουν συχνά να ανιχνεύσουν τον άγνωστο κακόβουλο κώδικα, μπορεί να παρακαμφεί από ένα νέο και μοναδικό malware. Επειδή μάλιστα είναι σύνηθες η μηχανή ανάλυσης κάθε εταιρείας να χρησιμοποιεί διαφορετικές υπογραφές και ευριστικές μεθόδους, για κάθε κακόβουλο πρόγραμμα θα ήταν χρήσιμο να σαρώσουμε το αρχείο μας με πολλές μηχανές antivirus. Αυτή τη δυνατότητα ευτυχώς την έχουμε πολύ εύκολα, μέσω κάποιον websites όπως είναι το γνωστό VirusTotal

(https://www.virustotal.com). Σε αυτό το website μπορούμε να στείλουμε όποιο αρχείο θέλουμε (μέχρι 64MB) και αυτό θα σαρωθεί από 46 διαφορετικές μηχανές antivirus. Έτσι λοιπόν επισκεπτόμαστε το website και ανεβάζουμε το εκτελέσιμό μας το οποίο έχουμε ονομάσει **Mitglieder.exe.**



Σχήμα 4: Εδώ βλέπουμε τα αποτελέσματα από την εξέταση του Trojan horse που αναλύουμε. Όπως βλέπουμε στην περίπτωσή μας 42 από τις 46 μηχανές antivirus διαθέτουν την υπογραφή του, καθώς και ότι το αναγνώρισαν ως malware.

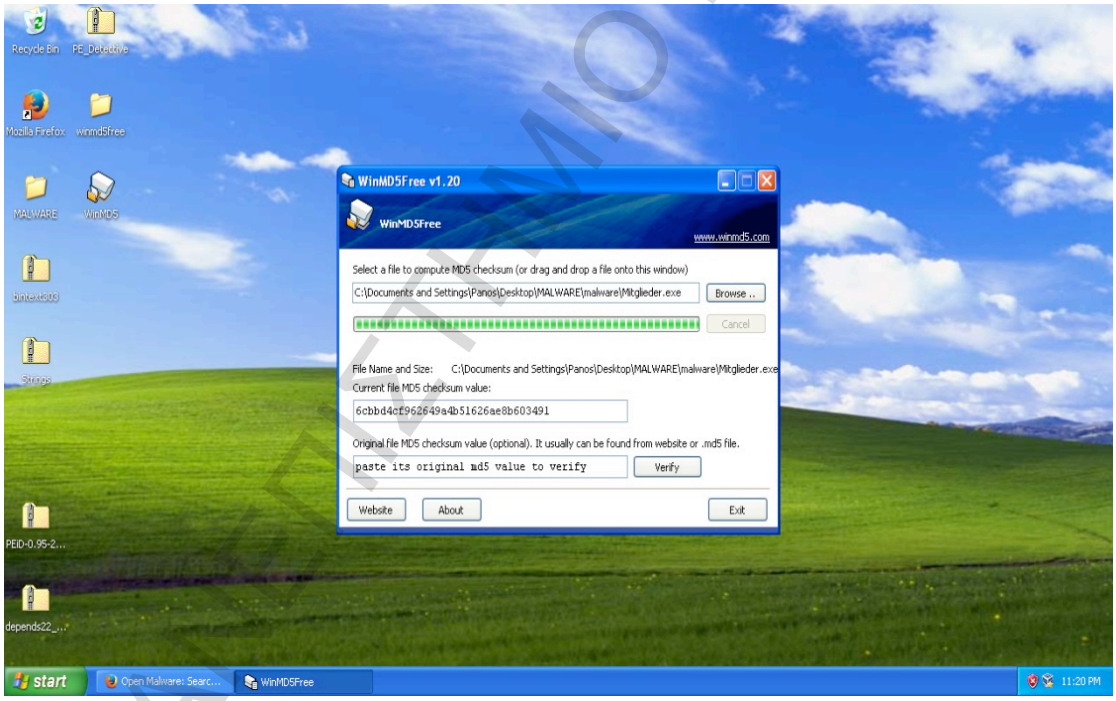
Βήμα 2. Παραγωγή αναγνωριστικού για το malware sample

Ένα κακόβουλο αρχείο συνήθως δεν παίρνει κάποιο συγκεκριμένο όνομα. Αυτό μπορούμε να το δούμε και από τα αποτελέσματα του VirusTotal, αφού διαφορετικές εταιρείες δίνουν διαφορετικά ονόματα στο ίδιο ακριβώς αρχείο. Για να είμαστε σε θέση να καταλάβουμε ποιο malware είναι πιο και να είμαστε σίγουροι ότι όλοι καταλαβαίνουν το ίδιο χρησιμοποιούμε την τεχνική του κατακερματισμού ή αλλιώς hashing. Το hashing είναι μια κοινή μέθοδος για να προσδιορίσουμε μοναδικά ένα κακόβουλο πρόγραμμα. Αυτό το κακόβουλο πρόγραμμα τρέχει μέσω ενός προγράμματος κατακερματισμού που παράγει ένα μοναδικό αλφαριθμητικό (hash) και αυτό προσδιορίζει το malware. Αυτό το αλφαριθμητικό είναι συγκεκριμένου πάντοτε μήκους και είναι αστρονομικά

απίθανο να παραχθεί από δυο διαφορετικές εισόδους επομένως στην πράξη μπορεί να χρησιμοποιηθεί ως αναγνωριστικό ή αλλιώς ένα είδος αποτυπώματος (fingerprint).

Αλγόριθμοι κατακερματισμού υπάρχουν διάφοροι. Ο πιο γνωστός από αυτούς είναι ο MD5 (Message-Digest Algorithm) καθώς και ο SHA-1 (Secure Hash Algorithm). Αφού παραχθεί το μοναδικό αποτύπωμα με τη χρήση του αλγορίθμου για το κακόβουλο δείγμα μας, μπορούμε κατόπιν να το χρησιμοποιήσουμε ως αναγνωριστικό και να το διαμοιράσουμε σε άλλους αναλυτές malware, ώστε να τους βοηθήσουμε στην αναγνώριση αλλά και στην ταυτοποίηση του υπό εξέταση κακόβουλου προγράμματος.

Για τους παραπάνω λόγους εγκαταστήσαμε στην εικονική μας μηχανή το εργαλείο WinMD5. Αν το τρέξουμε και επιλέξουμε το Mitglieder.exe θα πάρουμε ως έξοδο το αλφαριθμητικό **6cbbd4cf962649a4b51626ae8b603491**. Αυτό είναι ένα μοναδικό MD5 όνομα με το οποίο μπορούμε από εδώ και στο εξής να αναφερόμαστε στο συγκεκριμένο malware sample.

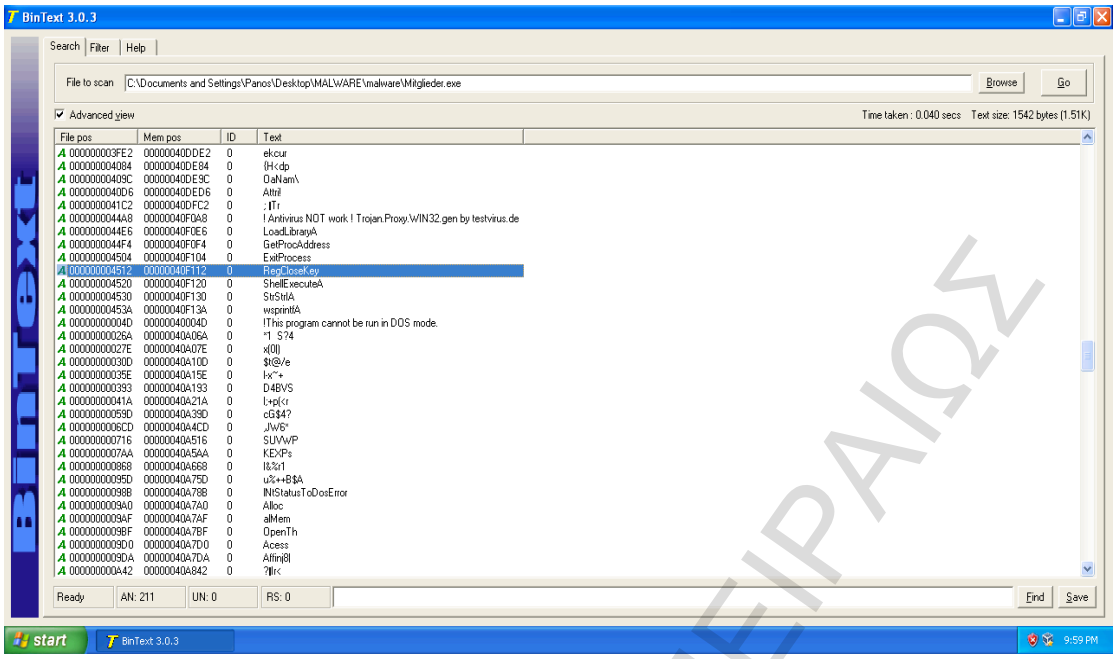


Σχήμα 4.1: Εδώ βλέπουμε την έξοδο του αλγορίθμου MD5 για το δείγμα που εξετάζουμε, όπως μας το δίνει το εργαλείο WinMD5Free.

Βήμα 3. Αναζήτηση αλφαριθμητικών

Όλα τα προγράμματα συνήθως εμπεριέχουν διάφορα αλφαριθμητικά ή αλλιώς strings. Ένα string είναι στην ουσία μια ακολουθία χαρακτήρων και ένα πρόγραμμα διαθέτει τέτοια, π.χ για την εκτύπωση μηνυμάτων στο χρήστη, τη σύνδεση σε ένα συγκεκριμένο website στο Internet ή σε μια διεύθυνση IP, την εκτέλεση ενεργειών σε αρχεία με συγκεκριμένη διαδρομή στο δίσκο κ.ο.κ. Μια πολύ σημαντική δυνατότητα των αλφαριθμητικών είναι πως ακόμα και όταν ο αρχικός πηγαίος κώδικας στη γλώσσα προγραμματισμού που γράφτηκε δεν είναι πάντα διαθέσιμος τα strings παραμένουν εντός του εκτελέσιμου (binary) και μπορούν να ανακτηθούν. Αρκετές φορές το να ψάχνεις τα strings μπορεί να είναι ένας απλός τρόπος για να πάρουμε χρήσιμες πληροφορίες και να αποκτήσουμε μια καλή εικόνα για το τι περίπου κάνει ένα κακόβουλο πρόγραμμα αρκεί να τα εξετάσουμε με προσοχή. Δυστυχώς, λόγω της φύσης των αλφαριθμητικών και του πώς αυτά αναπαρίστανται από τον υπολογιστή, ένα πρόγραμμα το οποίο εξάγει τα αλφαριθμητικά ενός άλλου αρχείου θα επιστρέψει και πάρα πολλά αλφαριθμητικά τα οποία δεν μας βοηθούν και ιδιαίτερα για την ανάλυση του malware sample. Τέτοια μπορεί να είναι διευθύνσεις μνήμης, εντολές προς τη CPU, κ.α τα οποία δεν είναι και τόσο χρήσιμα strings. Το καλό είναι πως είναι εύκολα αναγνωρίσιμα και μπορούμε να τα προσπερνάμε.

Για την εξαγωγή των strings έχουμε εγκαταστήσει δύο προγράμματα στην εικονική μας μηχανή: το Strings, που είναι εργαλείο κονσόλας, καθώς και το BinText, που εκτελείται σε παραθυρικό περιβάλλον. Ανοίγουμε λοιπόν το BinText διαλέγουμε το κακόβουλο αρχείο και πατάμε στο κουμπάκι Go, προκειμένου να εξάγουμε όλα τα strings. Αμέσως εμφανίζεται μια λίστα με όλα τα αλφαριθμητικά του προγράμματος. Σε αυτό το σημείο αρχίζει η διαδικασία της διαλογής. Εξετάζουμε τα strings ένα προς ένα απορρίπτωντας αυτά που δε μας χρειάζονται και κρατώντας οτιδήποτε άλλο φαίνεται ενδιαφέρον.



Σχήμα 4.2: Το πρόγραμμα BinText ανακτά τα strings, του κακόβουλου προγράμματος. Όπως φαίνεται δεν υπάρχει κάποιο ιδιαίτερα ενδιαφέρον string.

Βήμα 4: Ανίχνευση τεχνικών packing ή obfuscation

Οι προγραμματιστές κακόβουλου λογισμικού χρησιμοποιούν πολύ συχνά ειδικά εργαλεία για την υλοποίηση τεχνικών packing ή obfuscation, ώστε η ανίχνευση ή και η εκ των υστέρων ανάλυση να γίνεται αρκετά δύσκολα. Τα obfuscated προγράμματα είναι εκείνα που ο δημιουργός τους έχει προσπαθήσει να κρύψει τη λειτουργικότητά τους. Τα packed προγράμματα είναι ειδική περίπτωση των obfuscated. Στην ουσία έχουν συμπιεστεί ή αλλιώς έχουν πακεταριστεί όπως μαρτυρά και το όνομά τους. Για την ακρίβεια έχουν συμπιεστεί με είδικά εργαλεία που ονομάζονται packers. Όταν ένα εκτελέσιμο γίνεται packed, στην πραγματικότητα ενσωματώνεται στην αρχή του ένα δεύτερο πρόγραμμα. Δουλειά του είναι η αποσυμπίεση του κώδικα, όταν το πακεταρισμένο αρχείο εκτελείται. Αποτέλεσμα της διαδικασίας του packing είναι να αποκρύπτονται τα εμφανή στοιχεία του κακόβουλου αρχείου, όπως π.χ τα strings ή διάφορες κλήσεις συναρτήσεων. Και οι δύο τεχνικές περιορίζουν σημαντικά τις προσπάθειες για στατική ανάλυση του malware. Τα προγράμματα που δεν είναι packed σχεδόν πάντα περιλαμβάνουν πολλές συμβολοσειρές. Τα malware που είναι packed ή obfuscated περιέχουν πολύ λίγα strings. Εάν βρούμε λοιπόν ένα πρόγραμμα το

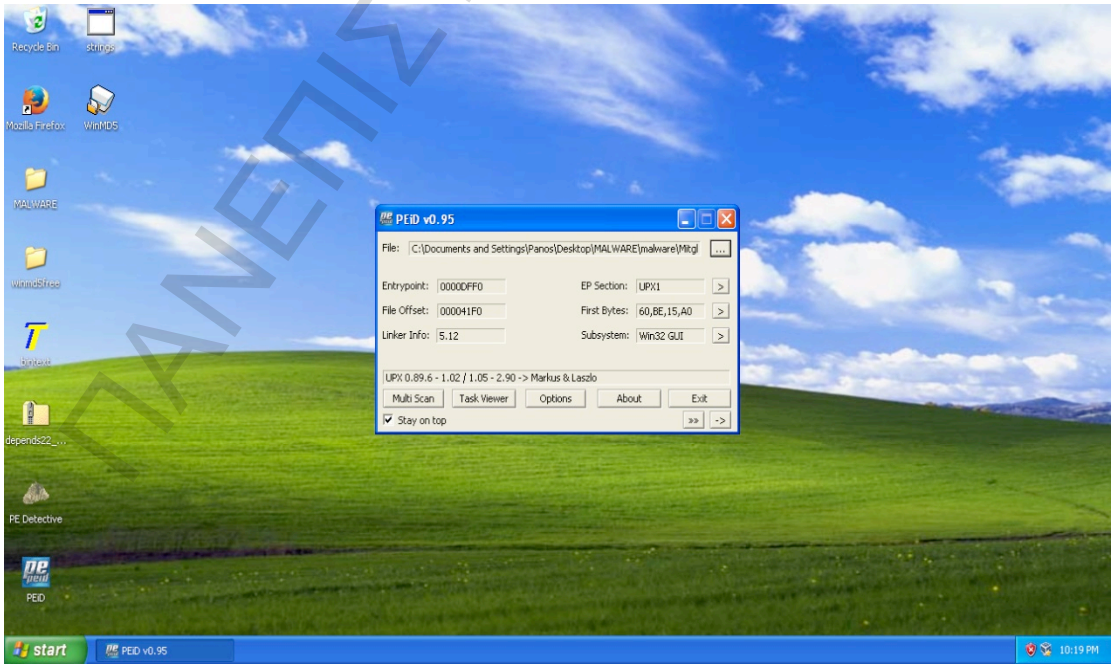
οποίο έχει λίγα strings, τότε είναι πολύ πιθανόν το πρόγραμμα να είναι είτε obfuscated είτε packed. Τώρα για να μπορέσουμε να επεξεργαστούμε ένα packed executable, θα πρέπει να το αποσυμπιέσουμε μόνιμα. Αυτό γίνεται σε δύο βήματα:

- βρίσκουμε με ποιον ακριβώς packer έχει συμπιεστεί
- εφαρμόζουμε πάνω του την αντίστροφη διαδικασία, χρησιμοποιώντας τον αντίστοιχο unpacker.

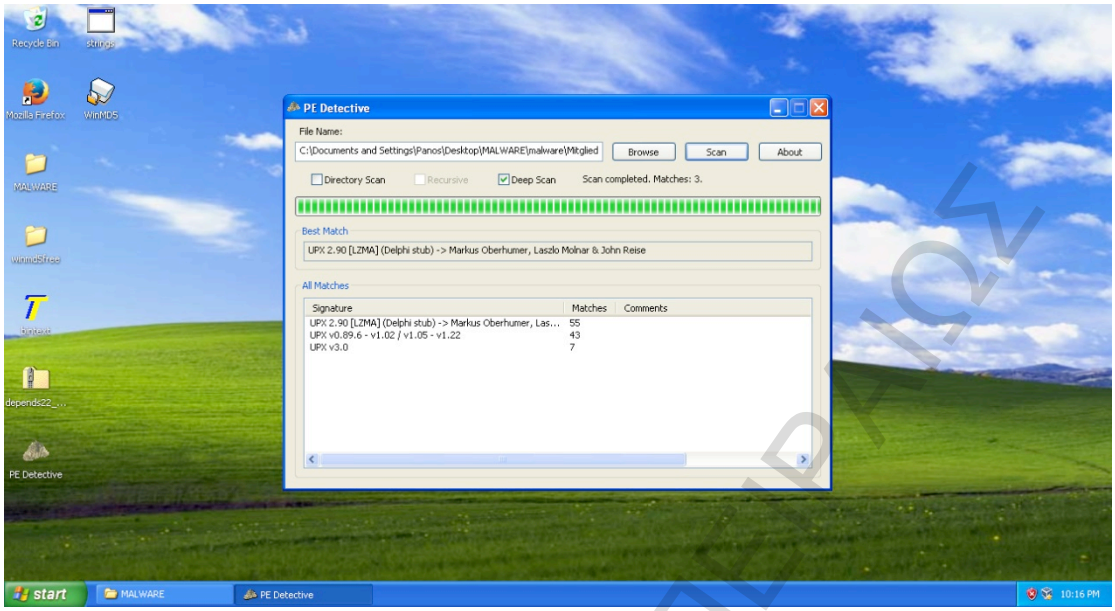
Πολλές φορές αυτό δεν είναι τόσο εύκολο, καθώς ένας προγραμματιστής μπορεί να έχει χρησιμοποιήσει custom packer! Σε αυτή τη περίπτωση πρέπει να φτιάξουμε το δικό μας unpacker. Συνήθως όμως οι περισσότεροι απλά χρησιμοποιούν γνωστούς packers που κυκλοφορούν στο Internet.

Τώρα προκειμένου να ανακαλύψουμε αν το κακόβουλο δείγμα μας είναι packed, θα χρησημοποιήσουμε τα εργαλεία PEiD και PE Detective που έχουμε εγκαταστήσει ήδη στην εικονική μηχανή. Χρησιμοποιούμε δύο ώστε να συγκρίνουμε τα αποτελέσματα μεταξύ τους. Σε γενικές γραμμές είναι καλύτερο να χρησιμοποιούμε περισσότερα του ενός εργαλεία για εργασίες σχετικά με την ανάλυση του malware διότι καθένα ενδέχεται να εφαρμόζει διαφορετική μέθοδο ανίχνευσης του χαρακτηριστικού που ψάχνουμε. Οπότε συγκρίνουμε τα αποτελέσματα για να καταλήξουμε σε πιο σίγουρο συμπέρασμα.

Αφού επιλέξουμε και για τα δύο προγράμματα (PEiD και PE Detective) το κακόβουλο αρχείο μας, αμφότερα συμπεραίνουμε ότι έχει συμπιεστεί με έναν από τους πιο γνωστούς packer: τον UPX.



*Σχήμα 4.3: Το **PEiD** ανίχνευσε ότι το κακόβουλο αρχείο έχει συμπιεστεί με τον γνωστό packer UPX*

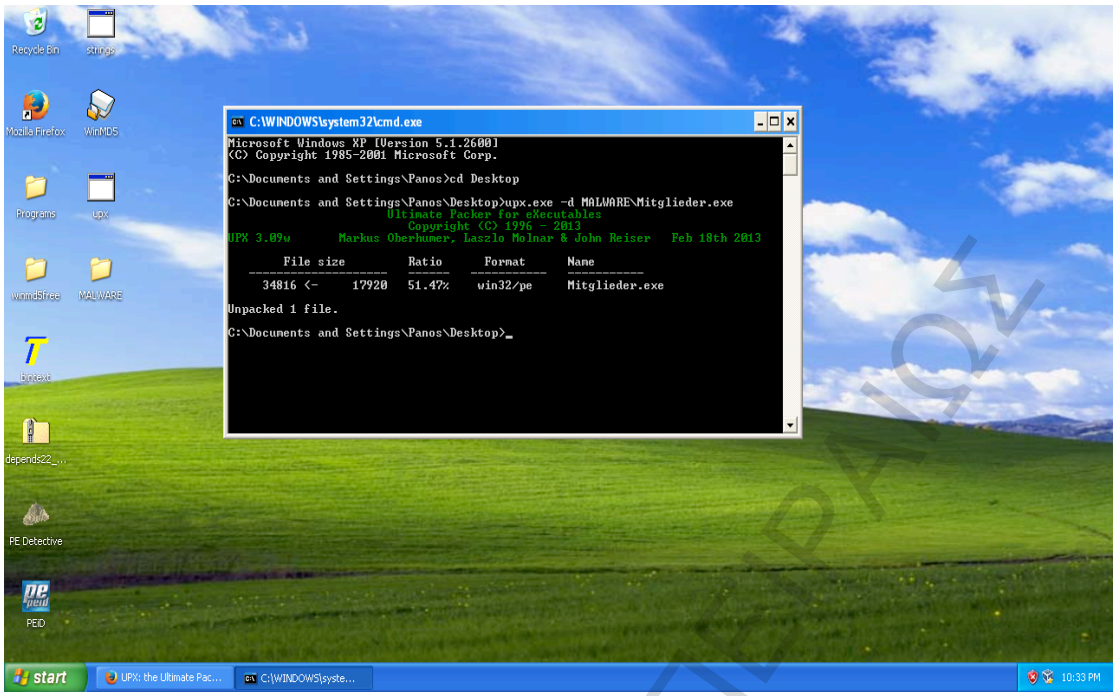


*Σχήμα 4.4: Το **PE Detective** μας επιβεβαιώνει και αυτό με τη σειρά του ότι ο προγραμματιστής του Trojan horse έχει χρησιμοποιείσει το UPX, με σκοπό να μας δυσκολέψει στην ανάλυση.*

Για να τον αποσυμπιέσουμε ώστε να το αναλύσουμε σωστά, θα μεταβούμε στον δικτυακό τόπο όπου φιλοξενείται το UPX (http://upx.sourceforge.net) και να κατεβάσουμε το πρόγραμμα (επιλέγουμε το αρχείο upx391w.zip για Windows). Στη συνέχεια εξάγουμε τα αποτελέσματα του ZIP και ανοίγουμε την κονσόλα των Windows (Start->Run και στη γραμμή Open γράφουμε cmd και πατάμε [Enter]). Πηγαίνουμε στο κατάλογο που βρίσκεται το αρχείο upx.exe. Συνεχίζουμε εκτελόντας το με τη παράμετρο –d (για decompress) και δίνουμε το path του αρχείου πάνω στο οποίο θα εφαρμοστεί. Στη περίπτωσή μας δίνουμε το εξής:

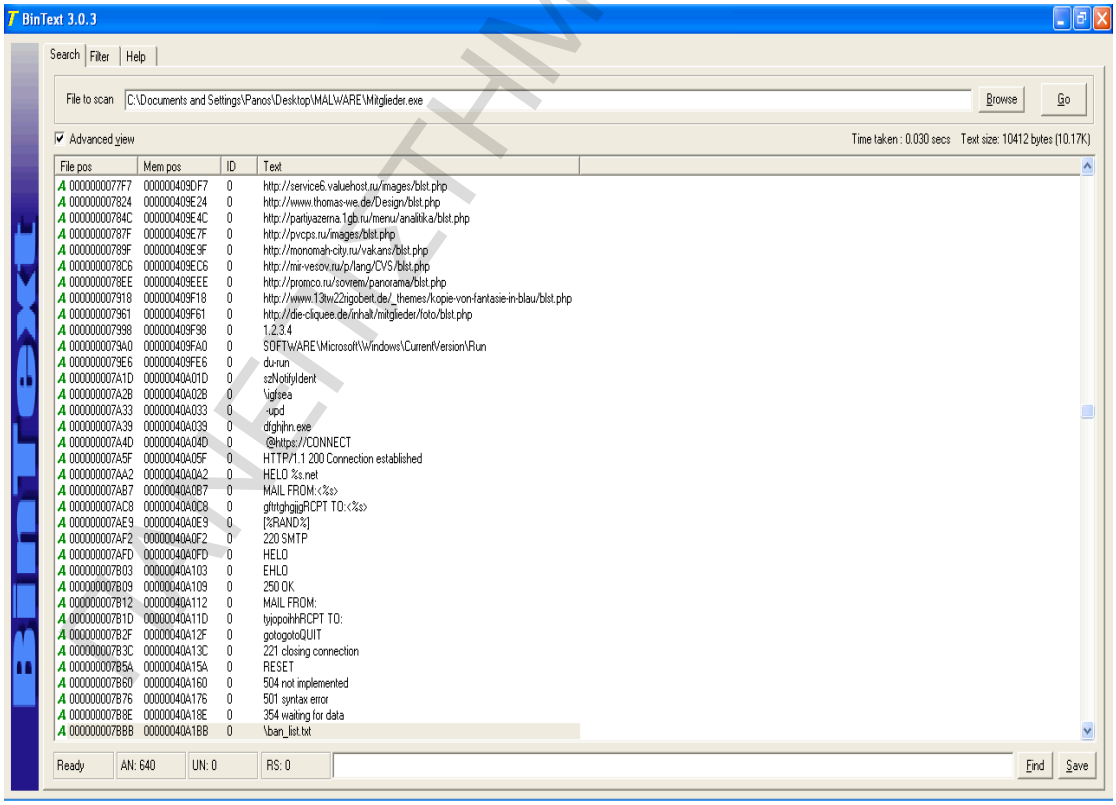
```
C:\Documents and Settings\Panos>cd Desktop

C:\Documents and Settings\Panos\Desktop>upx.exe –d MALWARE\Mitglieder.exe
```



Σχήμα 4.5: Το κακόβουλο εκτελέσιμο μόλις αποσυμπιέστηκε

Έχοντας αποσυμπιέσει το εκτελέσιμο θα δοκιμάσουμε να εξάγουμε για ακόμα μια φορά τα strings του προγράμματος.



Εικόνα 4.6: Μετά την αποσυμπίεση του εκτελέσιμου βλέπουμε ορισμένα ενδιαφέροντα string. Ορισμένα από αυτά είνα διευθύνσεις, ονόματα αρχείων, ηλεκτρονική αλληλογραφία κ.α

Τα αποτελέσματα που λαμβάνουμε αυτή τη φορά είναι εντελώς διαφορετικά. Ορισμένα από αυτά είναι συναρτήσεις συστήματος που καλεί το πρόγραμμα, ονόματα αρχείων που διαβάζει ή δημιουργεί (π.χ ban_list.txt) διαδρομές στο registry που μάλλον τροποποιεί κίολας (π.Χ SOFTWARE\Microsoft\Windows\CurrentVersion\Run –μάλλον προσθέτει τον εαυτό του για να εκκινείται αυτόματα), strings που φανερώνουν τη δυνατότητα αποστολής e-mail (π.χ MAIL FROM:, SMTP HELO κ.λπ). Τέλος να πούμε ότι δεν υπάρχει μια συγκεκριμένη μεθοδολογία με ποια σειρά πραγματοποιούμε την ανάλυση. Η διαδικασία της ανάλυσης πραγματοποιείται λίγο πολύ στα τυφλά τουλάχιστον στα πρώτα στάδια.

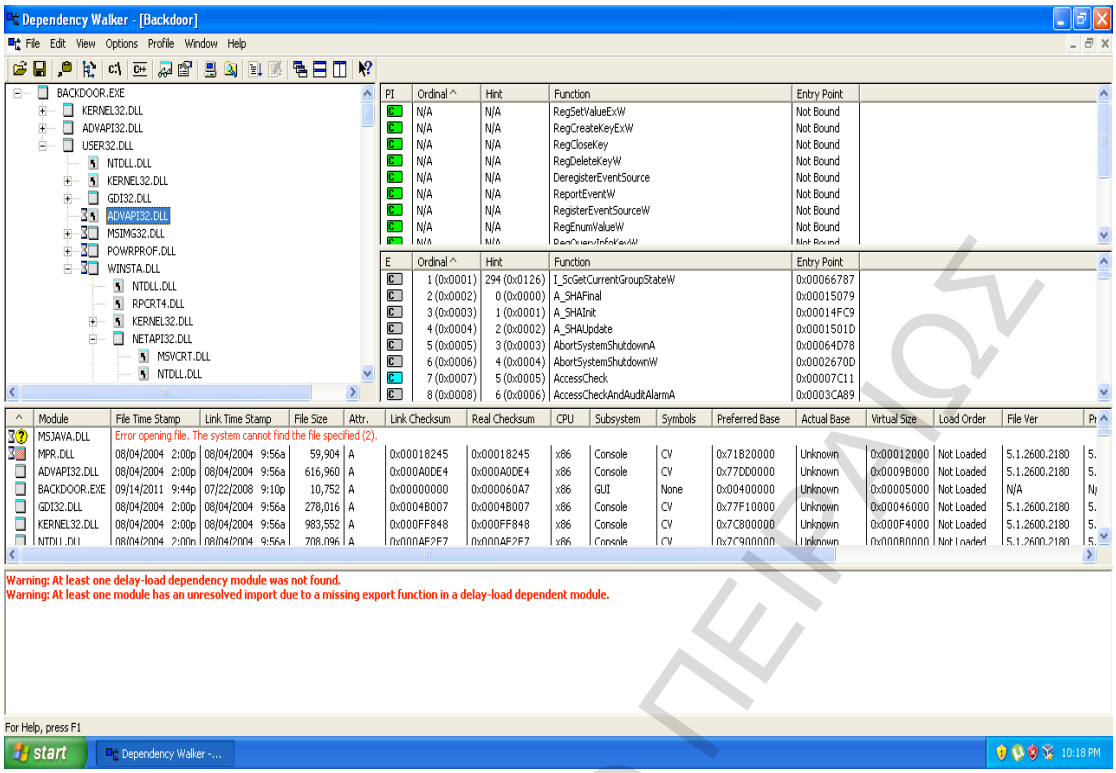
Βήμα 5. Ανάκτηση βιβλιοθηκών και συναρτήσεων συστήματος

Το τελευταίο βήμα με το οποίο θα ασχοληθούμε στη στατική ανάλυση είναι η ανάκτηση των βιβλιοθηκών (libraries) που χρησιμοποιεί το κακόβουλο πρόγραμμα. Κάτα κανόνα, την ώρα της εκτέλεσής του θα χρησιμοποιεί κάποιες συναρτήσεις από το σύστημα, οι οποίες προσφέρουν ειδικές λειτουργίες όπως π.χ είναι η δημιουργία κάποιου αρχείου. Αυτές οι συναρτήσεις βρίσκονται συνήθως σε ειδικά αρχεία βιβλιοθηκών, με την κατάληξη .DLL (Dynamic Link Library). Οι βιβλιοθήκες αυτές είναι όπως προσδίδει και το όνομα τους, δυναμικής σύνδεσης. Αυτό πολύ απλά σημαίνει ότι το λειτουργικό σύστημα ψάχνει να βρει τα κατάλληλα DLLs δυναμικά, δηλαδή κατά τη φόρτωση ενός προγράμματος που χρησιμοποιεί συναρτήσεις οι οποίες εμπεριέχονται σε αυτά. Όταν ακολούθως το πρόγραμμα κάνει χρήση της εκάστοτε συνάρτησης, αυτή εκτελείται μέσω της αντίστοιχης βιβλιοθήκης. Για να δούμε ποια DLLs φορτώνονται από το κακόβουλο πρόγραμμά μας, θα χρησιμοποιήσουμε την εφαρμογή Dependency Walker που έχουμε ήδη εγκαταστήσει στην εικονική μηχανή.

Δυστυχώς για το δικό μας παράδειγμα το αρχείο που έχουμε δεν είναι το κατάλληλο: Τα DLLs του συστήματος δεν εμφανίζονται. Για αυτό το λόγω θα δοκιμάσουμε ένα άλλο εκτελέσιμο αρχείο ώστε να δούμε τα DLLs που χρησιμοποιεί. Επισκεπτόμαστε πάλι το OffensiveComputing.net και γράφουμε τη λέξη «Trojan» στη μπάρα αναζήτησης. Κατεβάζουμε το δεύτερο κατά σειρά και βρίσκουμε ότι έχει συμπιεστεί και αυτό με τον UPX Packer και μέσα από τη κονσόλα των Windows εφαρμόζουμε την αντίστροφη διαδικασία.

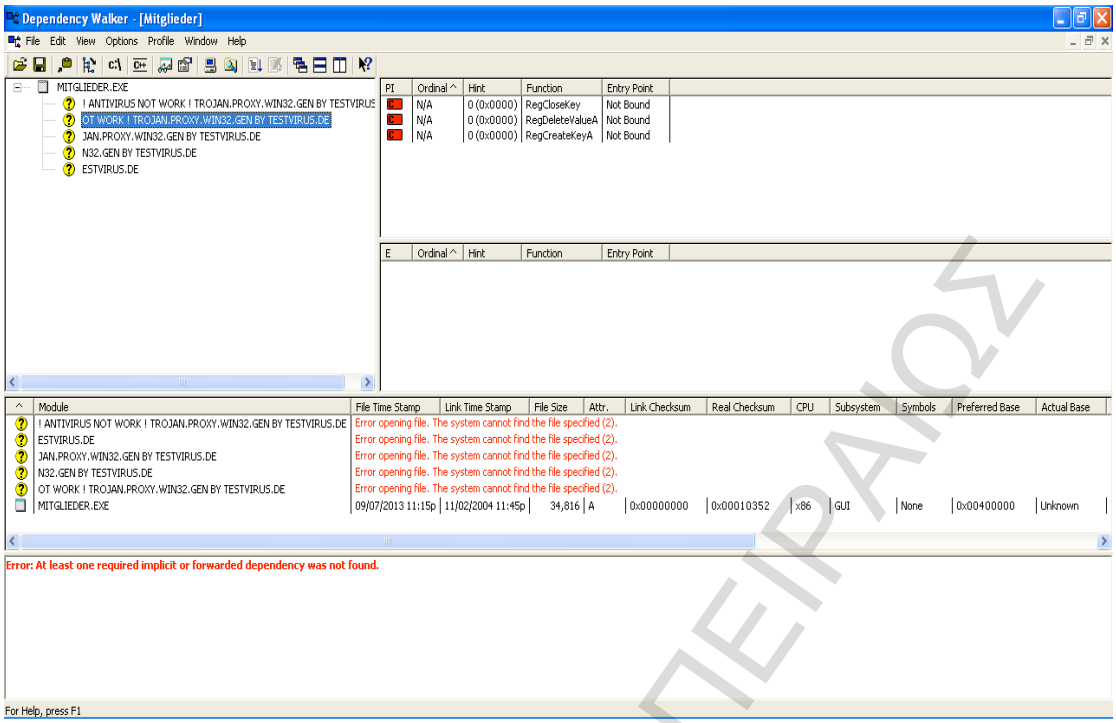
MD5:	efc33e74e0e53ee36c14c7b67f02331e
SHA1:	9ef218ec33df9a5e333d8dafdfb7515ca8c7357c
SHA256:	7e8ce771c4a504ce525a9fc88e5f0fc7d467485fd6f842c32d9b32bd013c052f
OCID:	21303
Original Filename:	Trojan-Downloader.Win32.Zlob.spg
Added:	2011-09-14 21:44:21.831057
AV Results:	**F-Prot:** W32/Backdoor2.CCCD (exact) **ClamAV:** Trojan.Zlob-7293

Στη συνέχεια το ανοίγουμε με την εφαρμογή Dependency Walker.



Εικόνα 4.7: Ορισμένα από τα DLLs που βλέπουμε να εμφανίζονται είναι τα Kernel32.DLL, ADVAPI32.DLL και USER32.DLL .

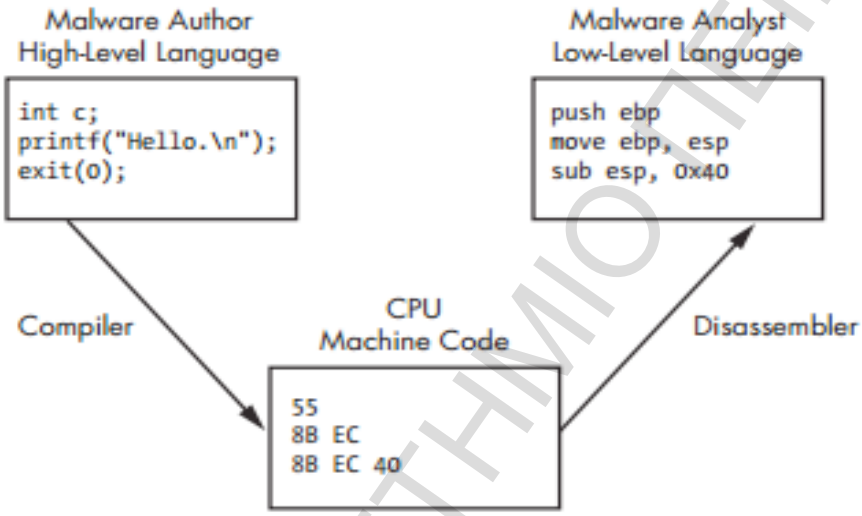
Επιστρέφουμε πάλι στο Mitglieder (το αρχικό μας malware sample) και πατάμε πάνω στα μηνύματα που εμφανίζονται στη λίστα αριστερά. Κάτω από το όνομα του κακόβουλου αρχείου, βλέπουμε στο παράθυρο πάνω δεξιά τις συναρτήσεις συστήματος που χρησιμοποιεί.



Εικόνα 4.8: Βλέπουμε πλέον ότι το κακόβουλο πρόγραμμα χρησιμοποεί ορισμένες συναρτήσεις τροποποίησης του registry (RegCreateKeyA, RegDeleteValueA).

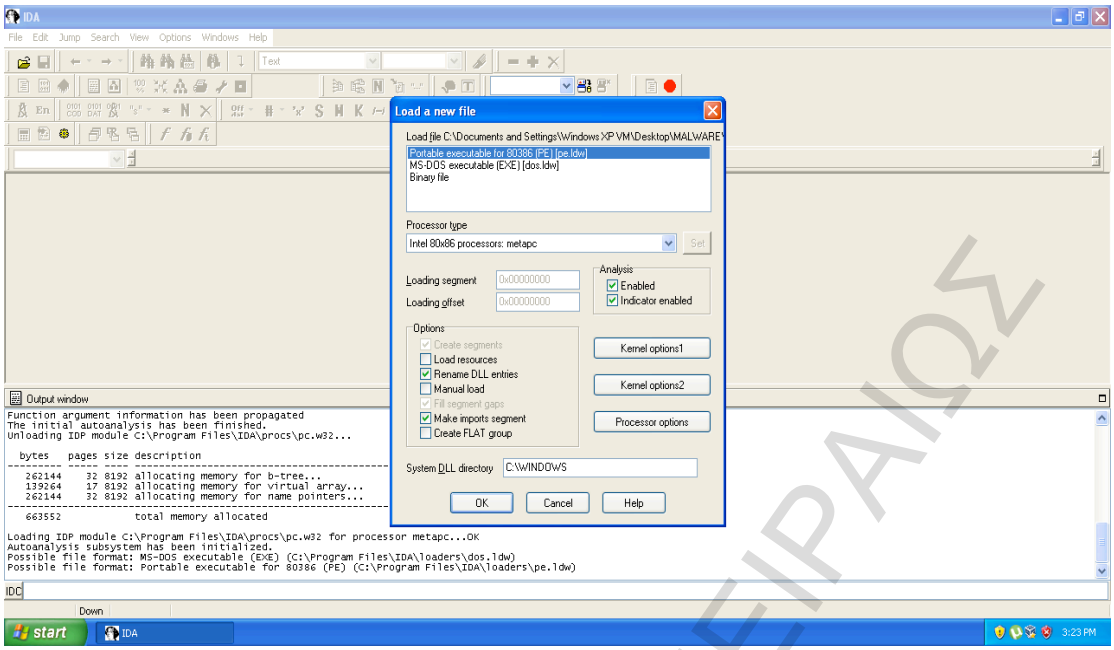
5. Προχωρημένη στατική ανάλυση malware

Όπως είδαμε προηγουμένως η βασική ανάλυση είναι χρήσιμη γιατί μας παρέχει πληροφορίες σχετικά με το κακόβουλο πρόγραμμα αλλά δεν μας παρέχει αρκετές πληροφορίες ώστε να αναλύσουμε το malware λεπτομερώς. Αυτή τη δυνατότητα μας τη δίνει η προχωρημένη ανάλυση. Η προχωρημένη στατική ανάλυση αφορά στην εισαγωγή του εκτελέσιμου αρχείου σε ένα disassembler, όπου παρατηρούμε και εξετάζουμε το κώδικά του σε συμβολική γλώσσα Assembly. Οι αναλυτές malware αναπτύσουν δραστηριότητες σε γλώσσες χαμηλού επιπέδου για να καταλάβουν καλύτερα πώς ένα πρόγραμμα λειτουργεί.



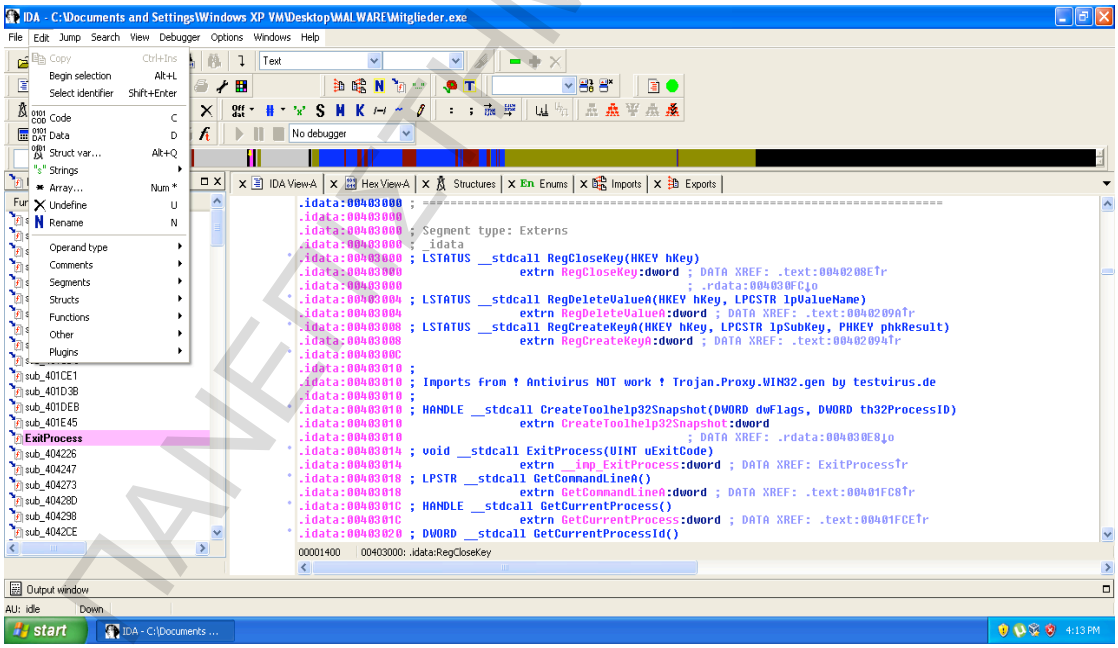
Στο παραπάνω σχήμα βλέπουμε ότι η ανάλυσή του κακόβουλου αρχείου γίνεται σε γλώσσα χαμηλού επιπέδου για να έχουμε καλή εικόνα του τι κάνει αυτό το εκτελέσιμο.

Για την προχωρημένη στατική ανάλυση θα χρησιμοποιήσουμε τον IDA Pro (Interactive Disassembler Professional) έναν εξαιρετικά ισχυρό disassembler που διανέμεται από την Hex-Rays. Ο IDA Pro χρησιμοποιήται από πολλούς αναλυτές malware, αναλυτές ευπαθειών καθώς και από πολλούς μελετητές που ασχολούνται με το reverse engineering. Επίσης υποστηρίζει διάφορους τύπους αρχείων όπως π.χ τα Portable Executable (PE), τα Common Object File Format (COFF), τα Executables Linking Format (ELF) κ.λπ. Όταν φορτώνουμε το malware sample ο IDA Pro προσπαθεί να αναγνωρίσει τον τύπο του αρχείου καθώς και την αρχιτεκτονική του επεξεργαστή (χ86, χ64 κ.λπ). Στην περίπτωσή μας το αρχείο μας εμφανίζεται να είναι PE αρχείο με x86 αρχιτεκτονική. Τώρα όταν φορτώνουμε ένα αρχείο στον IDA Pro (PE στην περίπτωσή μας) το πρόγραμμα μας αναλύει το κακόβουλο αρχείο όπως ακριβώς θα φορτωνόταν από το λειτουργικό σύστημα.

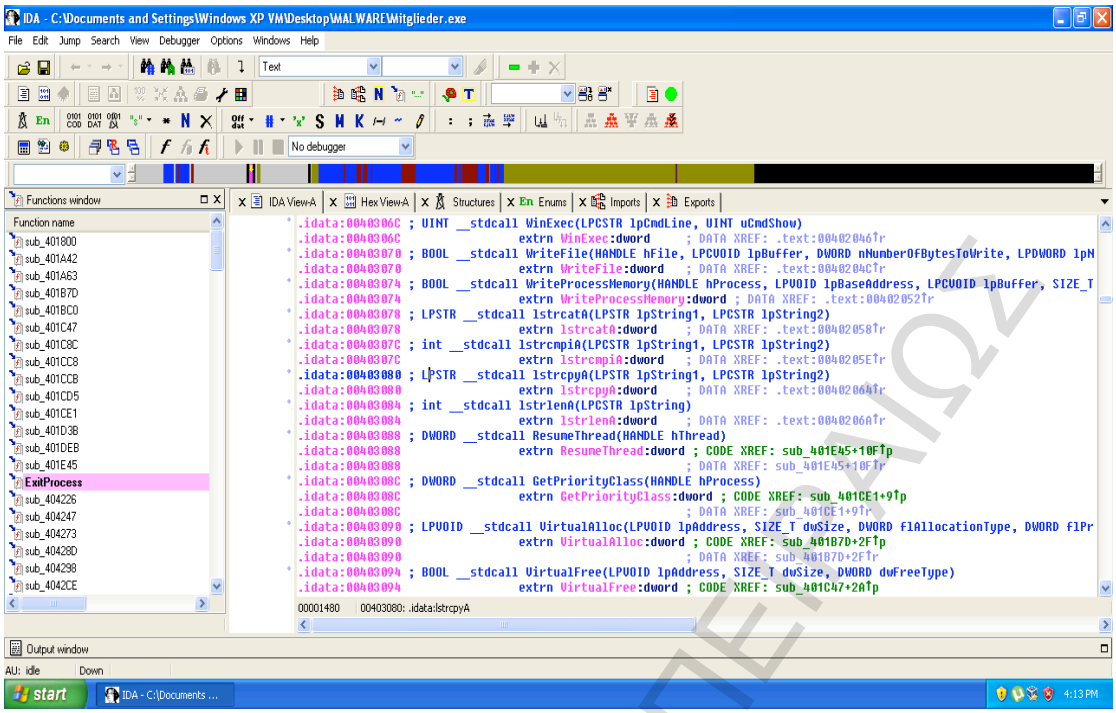


Εικόνα 5: Φόρτωση αρχείου στον IDA Pro

Ο IDA Pro δεν περιλαμβάνει την κεφαλίδα (header) PE ή τα τμήματα των πόρων (όπου συχνά το malware κρύβει το κακόβουλο κώδικά του). Για να φορτώσουμε κάθε τμήμα ένα προς ένα όπως είναι τα PE αρχεία ώστε αυτά να μη ξεφύγουν της ανάλυσης θα πρέπει να το φορτώσουμε χειροκίνητα.



Εικόνα 5.1



Εικόνα 5.2

Για τις δύο παραπάνω εικόνες μπορούμε να πούμε ότι ιδιαίτερα ενδιαφέρουσες είναι οι συναρτήσεις χειρισμού αρχείων (WriteFile, CreateFileA, CopyFileA), χειρισμού strings (lstrcpyA, lstlenA), και τροποποίησης του registry (RegCreateKeyA, RegDeleteValueA). Όπως φαίνεται και από τα προηγούμενα βήματα που εφαρμόσαμε συνδυάζουμε αυτή τη γνώση με τα προηγούμενα στοχεία που ανακαλύψαμε (όπως είναι τα strings) για να αποκτήσουμε μια πιο εμπεριστατωμένη εικόνα γύρω από τη συμπεριφορά του υπό ανάλυση malware. Πλέον ξέρουμε ότι το Trojan δημιουργεί ένα νέο κλειδί (key) στο registry και συγκεκριμένα στη διαδρομή που είδαμε προηγουμένως, στο BinText.

5.1 Συμπεράσματα

Μέχρις στιγμής πήραμε μια αρκετά καλή εικόνα από τη διαδικασία ανάλυσης του κακόβουλου λογισμικού. Είδαμε το βασικό αλλά και το προχωρημένο στάδιο της στατικής ανάλυσης καθώς και τα εργαλεία που συνήθως χρησιμοποιούνται σε αυτό. Η αλήθεια είναι πως μάθαμε αρκέτα για το malware sample που εξετάσαμε και το πιο σημαντικό από όλα αυτά είναι πως όλα αυτά τα μάθαμε χωρίς καν να το τρέξουμε! Στη συνέχεια λοιπόν θα δούμε τι μπορούμε να μάθουμε και να καταλάβουμε για τη λειτουργία του όταν το εκτελέσουμε.

6. Βασική δυναμική ανάλυση

Όπως αναφέραμε προηγουμένως, η στατική ανάλυση περιλαμβάνει όλες τις εργασίες που πραγματοποιούμε για την εξέταση ενός προγράμματος χωρίς όμως να το εκτελέσουμε. Η δυναμική ανάλυση αποτελεί τα δεύτερο στάδιο στη διαδικασία ανάλυσης του κακόβουλου λογισμικού. Πραγματοποιείται μετά τη βασική ανάλυση και όταν αυτή έχει φτάσει σε αδιέξοδο λόγω του packing, ή του obfuscation ή λόγου του ότι ο αναλυτής έχει εξαντλήσει όλες τις διαθέσιμες βασικές τεχνικές ανάλυσης. Κατά τη διαδικασία της δυναμικής ανάλυσης εξετάζουμε τη συμπεριφορά ενός κακόβουλου προγράμματος, την ώρα που αυτό εκτελείται. Η δυναμική ανάλυση μπορεί να περιλαμβάνει τον έλεγχο του malware καθώς τρέχει ή εξετάζοντας το σύστημα αφότου έχει εκτελεστεί το malware.

Σε αντίθεση με τη στατική, η δυναμική ανάλυση επιτρέπει να παρατηρήσουμε και να καταγράψουμε τη λειτουργικότητα ενός malware με βεβαιότητα. Η στατική ανάλυση δε οδηγεί πάντα σε ξεκάθαρα συμπεράσματα για το πώς λειτουργεί ένα malware και μας επιτρέπει μόνο να κάνουμε μερικές υποθέσεις. Για παράδειγμα εάν το malware μας είναι ένα keylogger η δυναμική ανάλυση μπορεί να μας επιτρέψει να εντοπίσουμε το log file του keylogger στο σύστημα, τα είδη αρχείων που διατηρεί καθώς και το πού στέλνει τις πληροφορίες. Αυτό θα ήταν δύσκολο να γίνει μέσω της στατικής ανάλυσης και επίσης θα απαιτούσε πάρα πολύ χρόνο.

Όπως βλέπουμε η δυναμική ανάλυση είναι πιο ουσιώδης και μας επιτρέπει να βγάζουμε πρακτικά συμπεράσματα για τη λειτουργία ενός προγράμματος. Αυτό όμως σε καμία περίπτωση δεν σημαίνει πως πρέπει να παραβλέπεται η στατική ανάλυση. Η δυναμική ανάλυση πρέπει να πραγματοποιείται πάντοτε σε συνδυασμό με τη στατική και μάλιστα αμέσως μετά (τη στατική).

Η δυναμική ανάλυση, εφόσον περιλαμβάνει την εκτέλεση του malware μπορεί να θέσει το δίκτυο και το σύστημά μας σε κίνδυνο. Επομένως πριν το εκτελέσουμε, θα ήταν καλό να έχουμε τουλάχιστον μια ιδέα για το τι πρόκειται να κάνει. Εξάλλου η δυναμική ανάλυση έχει και αυτή τους περιορισμούς της επειδή δεν μπορεί να εκτελεστεί όλως ο κώδικας όταν ένα μέρος του malware τρέχει. Για παράδειγμα ένα κακόβουλο πρόγραμμα θα μπορούσε να δέχεται παραμέτρους όταν π.χ εκτελείται από τη γραμμή εντολών (command line), οι οποίες και θα ενεργοποιούσαν νέες διαφορετικές λειτουργίες. Για αυτό το λόγω χωρίς τη στατική ανάλυση θα ήταν πολύ δύσκολο να μαντέψουμε ποιες είναι αυτές οι παράμετροι και το τι περίπου κάνει η καθεμία.

6.1 Ανάλυση συμπεριφοράς

Το malware sample εκτελείται αρκετές φορές και μάλιστα μέσα σε ένα ειδικά διαμορφωμένο σύστημα (π.χ εντός μιας εικονικής μηχανής). Κάθε φορά που εκτελούμαι το πρόγραμμα συγκεντρώνουμαι τη προσοχή μας σε μια από τις πολλές παραμέτρους της ανάλυσης συμπεριφοράς. Ένα malware μπορεί να πραγματοποιεί από πολύ μικρές έως και πολύ μεγάλες αλλαγές στο σύστημα. Θα πρέπει λοιπόν να έχουμε υπόψη μας όλους τους πιθανούς τομείς της δράσης του και να μελετήσουμε το καθένα συστηματικά και ξεχωριστά. Έτσι θα μπορέσουμε να συγκεντρώσουμε όλες τις απαιτούμενες πληροφορίες, ώστε να βγάλουμε στο τέλος χρήσιμα συμπεράσματα. Κατά την ανάλυση συμπεριφοράς που πραγματοποιείται μέσω της δυναμικής ανάλυσης η μελέτη που θα πραγματοποιήσουμε εκτείνεται στα ακόλουθα δύο πεδία:

1. **Σύστημα.** Τυπικά ένα κακόβουλο πρόγραμμα μεταβάλλει διάφορα στοιχεία στο σύστημα όπου αυτό εκτελείται. Η διερεύνηση αυτών των αλλαγών αποτελεί κατά κανόνα το πρώτο στάδιο της δυναμικής ανάλυσης. Πιο συγκεκριμένα ιδιαίτερο ενδιαφέρον έχουν τα ακόλουθα:
    - **Ανάλυση προσωρινής μνήμης RAM.** Το malware είναι αρκετά πιθανόν να υπερχειλίζει προσωρινές θέσεις μνήμης (buffer overflow) ή και να χειρίζεται τη μνήμη RAM με διάφορους ανορθόδοξους τρόπους, με σκοπό να αποκτήσει πρόσβαση σε διαβαθμισμένες λειτουργίες του συστήματος. Η καταγραφή της κατάστασης της μνήμης (memory dump) πριν αλλά και μετά την εκτέλεση του malware θα μας επιτρέψει να δούμε αν και πώς το malware χρησιμοποιεί τη μνήμη RAM για να ολοκληρώσει το σκοπό του.
    - **Αλλαγές σε ρυθμίσεις του συστήματος.** Συνήθως, αυτές πραγματοποιούνται στο registry. Εξετάζοντας την κατάσταση του registry πριν και μετά την εκτέλεση του malware, μπορούμε να δούμε αν και ποιες αλλαγές πραγματοποιήθηκαν, καθώς και το σκοπό που αυτές εξυπηρετούν.
    - **Ενέργειες σε αρχεία:** Ένα malware μπορεί να δημιουργεί, να τροποποιεί και να διαγράφει αρχεία του συστήματος. Ιδιαίτερα σημαντικό εδώ είναι να παρατηρήσουμε τις αλλαγές που πραγματοποιούνται κατά την εκτέλεση του malware. Πρέπει να είμαστε σε θέση να μπορούμε να δούμε ποια αρχεία προστέθηκαν, ποια τροποποιήθηκαν και ποια σβήστηκαν. Σε αυτή τη διαδικασία ιδιαίτερα σημαντική είναι η τεχνική του hashing, για την οποία μιλήσαμε στη βασική στατική ανάλυση. Ως γνωστόν ακόμη και ένα byte να αλλάξει σε κάποιο αρχείο, η συνάρτηση κατακερματισμού (hashing function) θα μας δώσει ένα εντελώς διαφορετικό αποτέλεσμα και η όποια τροποποίηση του αρχείου θα γίνει αμέσως αντιληπτή.

• **Διεργασίες και υπηρεσίες του συστήματος:** Σε αυτό το στάδιο εξετάζουμε αν εκκινήθηκαν νέες διεργασίες στο σύστημα, όπως επίσης και το αν άλλαξε η κατάσταση σε υπηρεσίες που εκτελούνταν ήδη. Για παράδειγμα είναι πολύ λογικό ένα malware να θέλει να σταματήσει τα προγράμματα antivirus ή να θέλει να αποκρύψει με κάποιον τρόπο τη δική του διεργασία (process). Και σε αυτή τη περίπτωση, καταγράφουμε την κατάσταση του περιβάλλοντος πριν και μετά την εκτέλεση του κακόβουλου αρχείου.

2. **Δίκτυο.** Αφού αποκτήσουμε μια εικόνα των όσων επιχειρεί το εκάστοτε malware στο ίδιο το σύστημα, προχωράμε στη μελέτη της δικτυακής συμπερφοράς του. Κατά κανόνα, ένα malware θα δοκιμάσει να συνδεθεί σε διάφορες τοποθεσίες στο Διαδίκτυο (domains ή διευθύνσεις IP), από τις οποίες θα περιμένει να δεχθεί εντολές για τη συνέχεια ή θα αποστείλει τα αποτελέσματα της δράσης του τα οποία μπορεί να είναι κωδικοί πιστωτικών καρτών που υπέκλεψε, λογαριαμοί e-mail κ.λπ. Επομένως η ανάλυση των δικτυακών κινήσεων είναι απαραίτητη για να κατανοήσουμε ποια ακριβώς είναι η αποστολή του. Όπως και προηγουμένως, εκτελούμε το malware sample αρκετές φορές και εστιάζουμε στα ακόλουθα:

   • **Δικτυακή κίνηση (network traffic).** Σε αυτό το στάδιο πρέπει να παρακολουθούμε και να καταγράφουμε τη δικτυακή κίνηση που σχετίζεται με το malware. Αυτή την διαδικασία την πραγματοποιούμαι με ειδικά εργαλεία που βλέπουν όλα τα πακέτα δεδομένων που έρχονται και φεύγουν από το σύστημα.

   • **Αναζήτηση ύποπτων προορισμών.** Εφόσον έχουμε στην κατοχή μας ένα αντίγραφο της δικτυακής κίνησης που προκλήθηκε από την εκτέλεση του malware, ξεκινάμε τη διαδικασία αναζήτησης ύποπτων στοιχείων. Ως τέτοια θα μπορούσαν να θεωρηθούν οι διευθύνσεις IP από γνωστά κακόβουλα websites, από παραβιασμένους υπολογιστές κ.λπ. Σε αυτή τη προσπάθεια μπορεί να βοηθήσει η γνωστή ιστοσελίδα του VirusTotal.

   • **Ανάλυση κίνησης.** Αφού γίνει η διαλογή και ο εντοπισμός των πιθανά ύποπτων προορισμών, μπορούμε να προχωρήσουμε στη μελέτη των δεδομένων που μεταφέρθηκαν. Με αυτό το τρόπο είναι αρκετά πιθανόν να πληροφορηθούμε για τα στοιχεία που συλλέγει το malware καθώς επίσης και με ποιο τρόπο επικοινωνεί μαζί του ο δημιουργός του.

Πρωτού τώρα περάσουμε στην ανάλυση θα πρέπει να θυμηθούμε ποιος είναι ο τελικός σκοπός της ανάλυσης ενός κακόβουλου προγράμματος: Ο σκοπός είναι να δημιουργήσουμε ένα αναγνωριστικό-προφίλ, το οποίο όμως θα περιγράφει με ακρίβεια διάφορα χαρακτηριστικά γνωρίσματα του

malware. Αυτό το προφίλ θα λειτουργεί κατά κάποιο τρόπο ως αναγνωριστικό – ταυτότητα και θα προσφέρει πληροφορίες για τη δράση του malware, τις διευθύνσεις με τις οποίες προσπαθεί να επικοινωνήσει κ.λπ. Τέλος με αυτές τις πληροφορίες που απαρτίζουν το προφίλ του κακόβουλου προγράμματος μπορούμε να φτιάξουμε την υπογραφή του εκάστοτε malware ώστε στο μέλλον να είναι εύκολη η μελλοντική ανίχνευση και αντιμετώπιση του υπό εξέταση προγράμματος.

6.2 Τα νέα μας εργαλεία

Τα στοιχεία που εξετάζουμε κατά τη δυναμική ανάλυση είναι πάρα πολλά. Για να καταγράψουμε όλα τα σχετικά δεδομένα και για να είμαστε σε θέση να παρακολουθήσουμε ακριβώς τη δράση του κακόβουλου προγράμματος, θα χρειαστούμε κάποια εργαλεία. Για να εγκαταστήσουμε αυτά τα εργαλεία θα χρειαστούμε και πάλι μια εικονική μηχανή την οποία στήνουμε με το VirtualBox. Στην εικονική μηχανή εγκαθιστούμε τα Windows XP SP3 και στη συνέχεια προσθέτουμε τα ακόλουθα εργαλεία.

- **Autoruns for Windows v11.70**

  http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

- **Process Hacker 2.33**

  http://freewarefiles.com/downloads_counter.php?programid=46335

- **Regshot 1.9.0**

  http://sourceforge.net/projects/regshot/

- **Process Monitor v3.05**

  http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

- **Wireshark 1.10.2**

  http://www.wireshark.org

• **Fakenet 1.0**

http://practicalmalwareanalysis.com/fakenet/

• **OllyDbg v1.10**

http://www.ollydbg.de/download.htm

Το επόμενο βήμα είναι να βρούμε ένα malware sample. Για ακόμη μια φορά θα καταφύγουμε στα γνωστά αποθετήρια (repositories) κακόβουλου λογισμικού και θα χρησιμοποιήσουμε την πλατφόρμα Open Malware. Ανοίγουμε λοιπόν τη σελίδα του OffensiveComputing.net και αναζητούμε το rxbot (πληκρολογούμε το όνομά του στη μπάρα πάνω αριστερά και πατάμε Search). Αμέσως θα μας επιστραφούν γύρω στα 20 αποτελέσματα. Εμείς επιλέξαμε να κατεβάσουμε το τελευταίο της λίστας και για να είμαστε ακριβείς, το αρχείο με MD5 hash **ba670bdb28bf451021de04ddf5f192ca.**

D5: ba670bdb28bf451021de04ddf5f192ca

HA1: 6d6f87e40332a5b6aa0d3295736e148c4559d0be

HA256: 49ddf550a906ffc5894e2f64f092484d647e992b3f13a9810f6318ce4e6e08

CID: 174225188

iginal lename:	ba670bdb28bf451021de04ddf5f192ca
lded:	2007-11-18 20:27:37.2498
/ Results:	**ClamAV:** Trojan.Poebot-32 **BitDefender:** Win32.Worm.Rxbot.AB **AVG:** Win32/Virut.A
:her formation:	Virustotal ThreatExpert

Το κατεβάζουμε λοιπόν εντός της εικονικής μηχανής και το αποσυμπιέζουμε χρησιμοποιώντας τον κωδικό «infected». Στη συνέχεια θα εκτελέσουμε το κακόβουλο πρόγραμμα και η εικονική μηχανή θα μολυνθεί. Για αυτό το λόγο παίρνουμε ένα snapshot από την καθαρή (τουλάχιστον για την ώρα) εικονική μηχανή. Αυτό γίνεται ακολουθώντας τη διαδρομή Machine ->Take Snapshot και δίνουμε ένα περιγραφικό όνομα όπως σαν το «Dynamic Malware Analysis Lab - clean» και προχωράμε στη δημιουργία του snapshot.

6.3 Βήματα βασικής δυναμικής ανάλυσης

Όπως και με τη στατική ανάλυση, έτσι και με τη δυναμική έχουμε δύο επίπεδα: Το βασικό και το προχωρημένο. Το δεύτερο περιλαμβάνει τη φόρτωση και την εκτέλεση του μεταγλωττισμένου αρχείου σε έναν debugger. Για αυτό το κομμάτι θα μιλήσουμε στη συνέχεια πιο αναλυτικά. Τώρα θα ασχοληθούμε όμως με το βασικό στάδιο, από το οποίο θα πάρουμε απαντήσεις για όλα τα κρίσιμα ερωτήματα που περιγράψαμε νωρίτερα. Ας αρχίσουμε λοιπόν να εξετάζουμε βήμα προς βήμα τη δράση του malware sample που κατεβάσαμε.

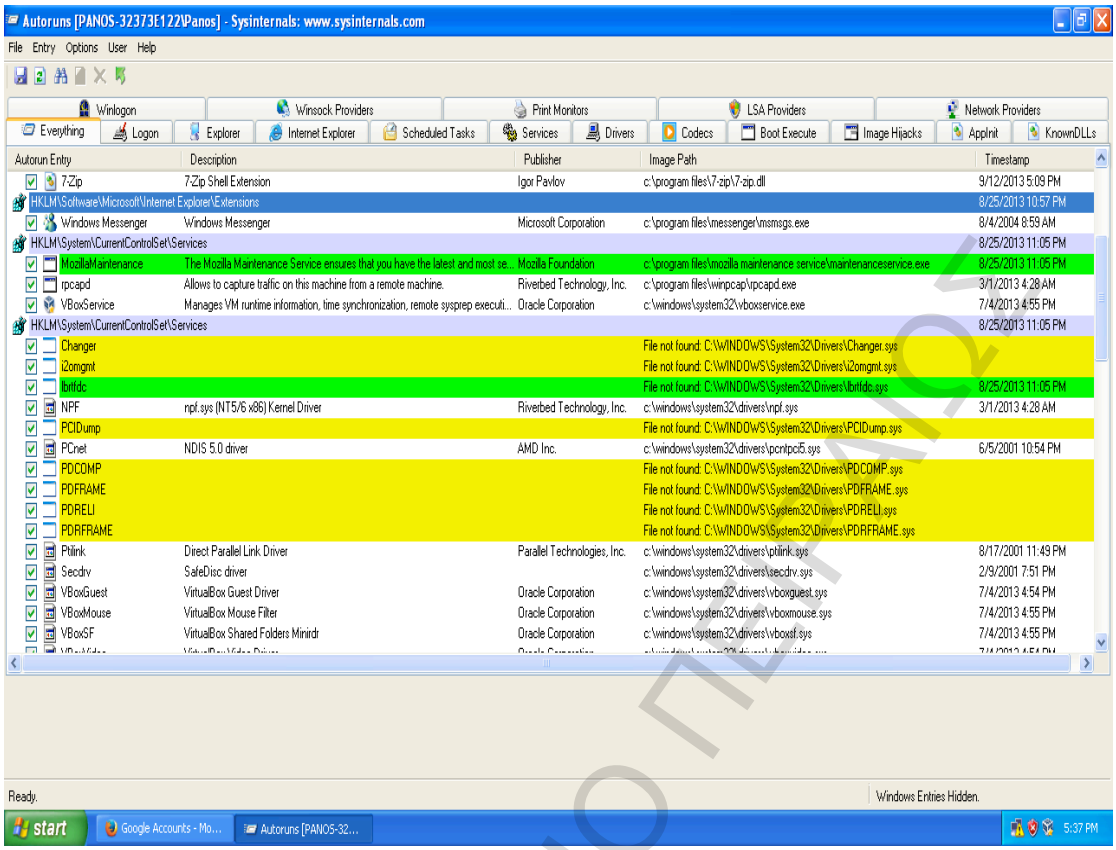
1ο Βήμα: Έλεγχος των προγραμμάτων στην αυτόματη εκκίνηση

Δεν είναι λίγες οι φορές που τα Windows απαιτούν αρκετό χρόνο για να εκκινήσουν τη λειτουργία τους. Η πιο πιθανή αιτία που συμβαίνει αυτό είναι το μεγάλο πλήθος των εφαρμογών εκκίνησης του υπολογιστή μας. Επίσης, η κατάσταση γίνεται χειρότερη όταν υπάρχει μια προβληματική εφαρμογή εκκίνησης. Μια προβληματική εφαρμογή εκκίνησης μπορεί να παρουσιαστεί, όταν προσπαθήσουμε να απεγκαταστήσουμε κάποιο πρόγραμμα στον υπολογιστή μας ενώ αυτό ήταν σε λειτουργία τη τρέχουσα στιγμή, με συνέπεια να έχουν παραμείνει κατάλοιπα του προγράμματος αυτού στη registry του συστήματος μας. Η λύση σε αυτή τη περίπτωση μπορεί πολύ εύκολα να δωθεί με το msconfig του λειτουργικού συστήματος των Windows. Για να το επιτύχουμε αυτό ακολουθούμε τη διαδρομή Start-> Run. Εκεί πληκτρολογούμε την εντολή «msconfig» και πατάμε το πλήκτρο «OK». Στο παράθυρο που θα ανοίξει επιλέγουμε την καρτέλα “Startup” και εκεί μπορούμε να παρατηρήσουμε όλες τις εφαρμογές που τρέχουν στον υπολογιστή κατά την εκκίνηση των Windows. Μπορούμε να επιλέξουμε ποιες από τις εφαρμογές αυτές δεν χρειαζόμαστε και να τις σταματήσουμε. Θα πρέπει όμως να είμαστε ιδιαίτερα προσεκτικοί με τις αλλαγές που θα κάνουμε γιατί μπορεί να προξενήσουμε κακό αντί για καλό. Μπορούμε για παράδειγμα να επιλέξουμε να αφαιρέσουμε μια διεργασία του λειτουργικού συστήματος η οποία θα έχει αρνητικές συνέπειες στη λειτουργία των Windows.

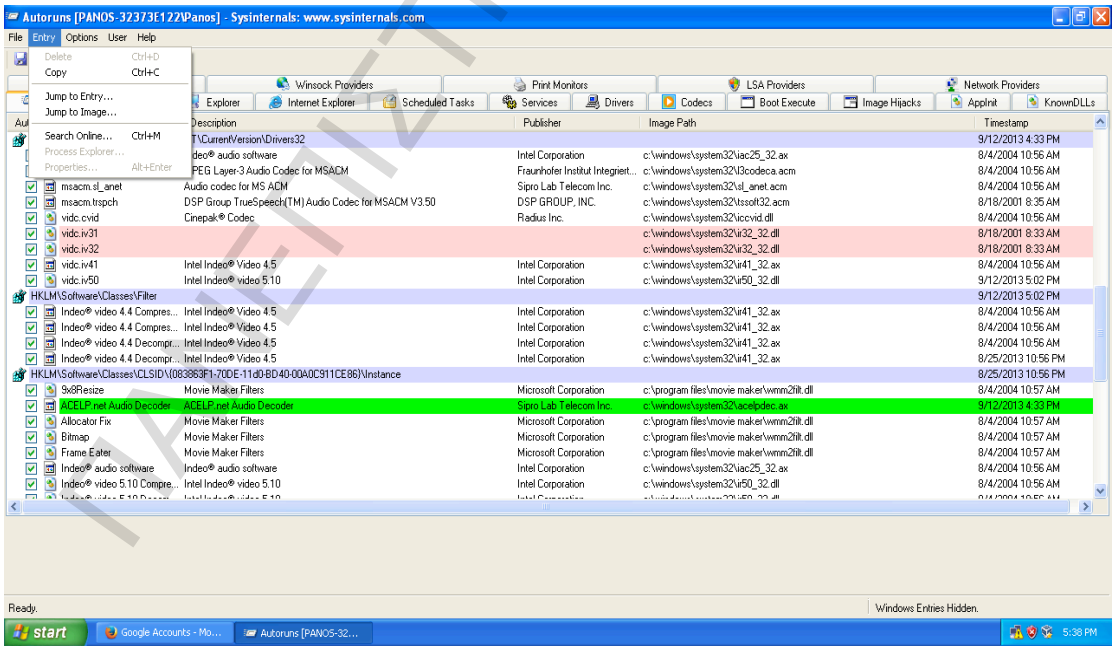
Ως γνωστόν, τα Windows διαθέτουν διάφορους μηχανισμούς για την αυτόματη εκκίνηση προγραμμάτων, κάθε φορά που ξεκινά το λειτουργικό σύστημα. Κατά συνέπεια είναι απόλυτα φυσιολογικό ο δημιουργός ενός malware να θέλει να εντάξει το πρόγραμμά του σε κάποιον από τους σχετικούς μηχανισμούς. Για να είμαστε σε θέση να ανακαλύψουμε αν πράγματι συμβαίνει κάτι τέτοιο αλλά και με ποιον ακριβώς τρόπο επιτυγχάνεται, θα χρησιμοποιήσουμε το Autoruns. Το Autoruns είναι ένα πρόγραμμα που δημιούργησε αρχικά η γνωστή εταιρεία

κατασκευής εργαλείων για Windows, με το όνομα Wininternals. Η Microsoft αγόρασε την εν λόγω εταιρεία το 2006 και από τότε τα εργαλεία της αποτελούν την σουίτα Windows Sysinternals. Το Autoruns τσεκάρει όλους τους μηχανισμούς αυτόματης εκκίνησης, εξετάζει τους σχετικούς καταλόγους του συστήματος, ελέγχει τις περιοχές του registry, μας επιτρέπει να τρέξουμε μια ανάλυση για κάθε εφαρμογή εκκίνησης και μπορεί να επιταχύνει αισθητά την εκκίνηση των Windows. Πρόκειται για ένα αρχείο .zip το οποίο περιλαμβάνει δύο εκδόσεις του Autoruns, εμείς όμως θα χρησιμοποιήσουμε το αρχείο "autoruns.exe" το οποίο χρησιμοποιεί γραφικό περιβάλλον. Συνεπώς δεν απαιτείται κάποια εγκατάσταση για τη λειτουργία του προγράμματος διότι είναι εκτελέσιμο αρχείο (.exe). Επομένως κάνουμε διπλό κλικ στοα αρχείο και ανοίγει αμέσως το πρόγραμμα. Έτσι όταν τρέχουμε το πρόγραμμα στο πρώτα δευτερόλεπτα εμφανίζονται όλα τα προγράμματα που ξεκινούν μαζί με το σύστημα και ο υπολογιστής μας εντοπίζει όλες τις διεργασίες που τρέχουν σε αυτό και μάλιστα με τη σειρά εκτέλεσής τους.

Μια ιδιαίτερα χρήσιμη δυνατότητα που παρέχει το πρόγραμμα είναι η αποθήκευση αυτής της λίστας σε αρχείο. Ένας αναλυτής malware μπορεί να αποθηκεύσει τη συγκεκριμένη λίστα μια φορά πριν από την εκτέλεση του κακόβουλου προγράμματος και άλλη μια μετά από την εκτέλεση. Ακολούθως συγκρίνοντας τις δύο λίστες θα είμαστε σε θέση να τσεκάρουμε αν προστέθηκαν προγράμματα σε κάποιον μηχανισμό αυτόματης εκκίνησης, σε ποια θέση του συστήματος αρχείων τοποθετήθηκαν, αν τροποποιήθηκε το registry, κ.λπ. Ανοίγουμε λοιπόν το εργαλείο και περιμένουμε να ολοκληρώσει τους ελέγχους του. Μόλις τελειώσει επιλέγουμε το Save από το μενού File και αποθηκεύουμαι το αποτέλεσμα σε κάποιο αρχείο. Αμέσως μετά εκτελούμε το malware sample! Στην οθόνη του υπολογιστή μας δεν θα δούμε τίποτα αφού η λειτουργία του είναι παρασκηνιακή. Εξάλλου, μετά από λίγο θα εξαφανιστεί και το ίδιο το αρχείο που μόλις εκτελέσαμε. Αυτό είναι απολύτως λογικό αφού το να υπάρχει μπροστά στα μάτια μας το εκτελέσιμο δεν είναι κάτι το οποίο θα ήθελε ο δημιουργός του. Για αρχή θα πρέπει να δούμε αν είχαμε κάποια αλλαγή στους μηχανισμούς αυτόματης εκκίνησης. Επισκεπτόματε λοιπόν το ανοικτό παράθυρο του Autoruns, ακολουθούμε τη διαδρομή File-> Refresh και αφήνουμε το πρόγραμμα να κάνει τους ελέγχους του. Για να είμαστε πιο σίγουροι για τις αλλαγές, επιλέγουμε το Compare από το μενού File και δίνουμε στο πρόγραμμα το αρχείο που αποθηκεύσαμε νωρίτερα. Το πρόγραμμα θα συγκρίνει τα περιεχόμενα του αρχείου με την τρέχουσα κατάσταση του συστήματος. Ένα πολύ καλό στοιχείο που ενσωματώνει το συγκεκριμένο πρόγραμμα είναι ότι επισημαίνει όλες τις διαφορές με πράσινο χρώμα. Στις δύο παρακάτω εικόνες μπορούμε να δούμε τα αποτελέσματα του πειράματος που πραγματοποιήσαμε.



Εικόνα 6: Το κακόβουλο πρόγραμμα πρόσθεσε ένα νέο πρόγραμμα για αυτόματη εκκίνηση



Εικόνα 6.1: Το κακόβουλο πρόγραμμα πρόσθεσε ένα νέο πρόγραμμα για αυτόματη εκκίνηση

Το συγκεκριμένο κακόβουλο πρόγραμμα δημιουργεί κάθε φορά ένα αρχείο με διαφορετικό όνομα. Αυτό που κάνει, είναι να επιλέγει τυχαία κάποιο από τα βασικά αρχεία του λειτουργικού συστήματος και να δημιουργεί κάθε φορά ένα αρχείο με παρόμοιο όνομα. Στη δική μας περίπτωση δημιούργησε το αρχείο με όνομα MozillaMaintenance (όπως φαίνεται στη παραπάνω εικόνα) το οποίο παραπέμπει στο γνωστό browser της Mozilla. Με αυτό το τρόπο λοιπόν το κακόβουλο πρόγραμμα προσπαθεί να περνά απαρατήρητο από τα μάτια του χρήστη. Επίσης με το να χρησιμοποπειεί κάθε φορά διαφορετικά ονόματα αρχείων, δυσκολεύει και τον εντοπισμό του.

Σε αυτό το σημείο θα πραγματοποιήσουμε και τη πρώτη μας επιστροφή στο καθαρό Snapshot που έχουμε κρατήσει από την εικονική μηχανή. Για το σκοπό αυτό πατάμε στο κουμπί κλεισίματος (X) του παραθύρου της μηχανής και από το μενού που μας εμφανίζει επιλέγουμε την τρίτη επιλογή (power off the machine). Αμέσως μετά μαρκάρουμε το “restore current snapshot” και πατάμε [Enter]. Με αυτόν τον τρόπο, μέσα σε λίγο χρόνο η μηχανή θα είναι έτοιμη για να την εκκινήσουμε πάλι.

2ο Βήμα: Επισκόπηση και καταγραφή κακόβουλων διεργασιών

Στους υπολογιστές, η οντότητα που ονομάζουμε διεργασία (process) είναι μια περίπτωση ενός προγράμματος υπολογιστών το οποίο εκτελείται. Περιέχει τον κώδικα προγράμματος, την τρέχουσα δραστηριότητά του και σχετικές τιμές που έχουν αποθηκευτεί στη μνήμη RAM. Όταν εκτελούμε ένα πρόγραμμα, το λειτουργικό σύστημα δημιουργεί αυτόματα την αντίστοιχη διεργασία. Ανάλογα με το λειτουργικό σύστημα (OS), μια διαδικασία μπορεί να αποτελείται από πολλαπλά νήματα εκτέλεσης που εκτελούν τις εντολές ταυτόχρονα.

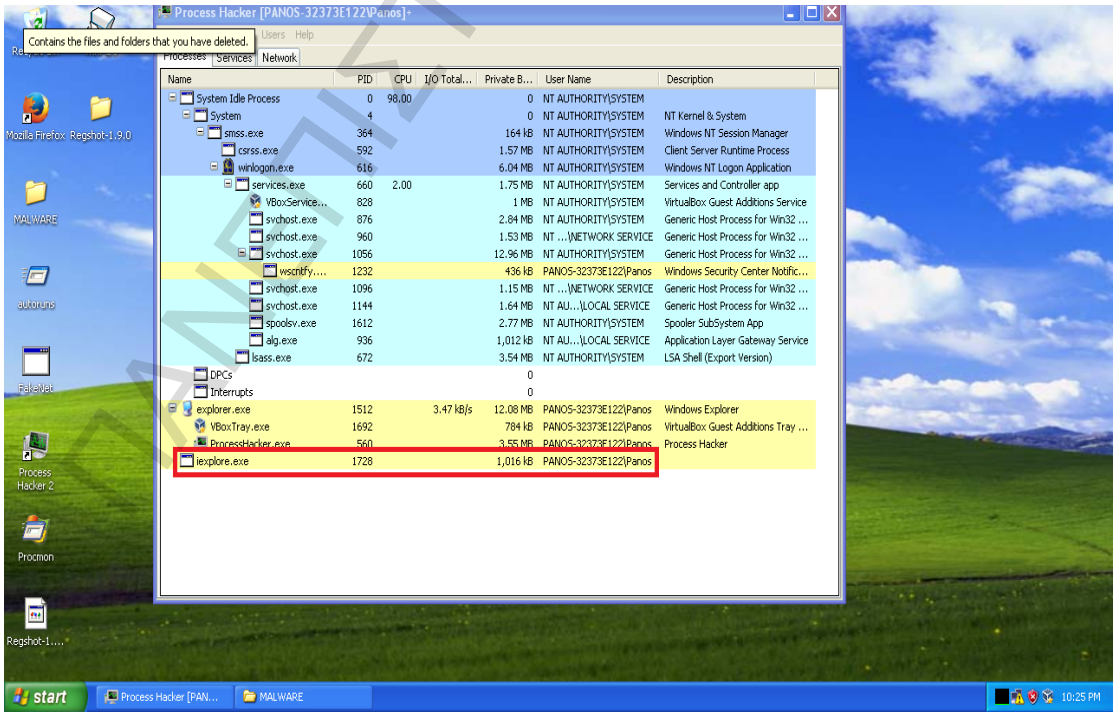
Ένα πρόγραμμα υπολογιστών είναι μια παθητική συλλογή εντολών ενώ μια διεργασία είναι η πραγματική εκτέλεση των εντολών. Διαφορετικές διεργασιες τώρα μπορούν να συσχετιθούν με το ίδιο πρόγραμμα παραδείγματος χάριν το άνοιγμα διάφορων περιπτώσεων του ίδιου προγράμματος σημαίνει συχνά ότι περισσότερες από μια διεργασίες εκτελούνται.

Ένα πρόγραμμα μπορεί να σχετίζεται με παραπάνω από μια διεργασία (parent process), ενώ μια διεργασία μπορεί να κατασκευάζει πρόσθετες διεργασίες που ονομάζονται υποδιεργασίες (child process). Μια υποδιεργασία κληρονομεί τις περισσότερες από τις ιδιότητές της από τη γονική. Στο Unix μια υποδιεργασία δημιουργείται τυπικά ως αντίγραφο της γονικής χρησιμοποιώντας τη συνάρτηση

fork(). Κάθε διεργασία μπορεί να δημιουργήσει πολλές υποδιεργασίες αλλά θα έχει το πολύ μια γονική διεργασία. Εάν μια διεργασία δεν έχει γονική αυτό συνήθως δείχνει ότι δημιουργήθηκε άμεσα από τον πυρήνα. Σε μερικά συστήματα, που βασίζονται στο Unix όπως είναι τα Linux, η πρωταρχική διεργασία (init) αρχίζει από τον πυρήνα κατά την εκκίνηση του λειτουργικού συστήματος. Ένας άλλος τρόπος για να καταλήξει μια διεργασία χωρίς γονική είναι εάν η γονική πεθάνει, αφήνοντας μια διεργασία ορφανή, αλλά σε αυτή τη περίπτωση θα υιοθετηθεί σύντομα από την init. Όταν μια υποδιεργασία ολοκληρωθεί πρωτού η γονική καλέσει τη συνάρτηση wait() ο πυρήνας διατηρεί κάποιες πληροφορίες για τη διεργασία όπως είναι η θέση εξόδου του, για να επιτρέψει στη γονική του να καλέσει αργότερα την call().

Αφού είδαμε κάποια θεωρητικά πράγματα ας εξετάσουμε όλα όσα συμβαίνουν κατά την εκτέλεση του κακόβουλου προγράμματος. Αυτή τη φορά θα χρησιμοποιήσουμε το εργαλείο Process Hacker. Με αυτό μπορούμε να παρακολουθούμε εύκολα όλες τις αλλαγές στις ενεργές διεργασίες του συστήματος.

Ανοίγουμε λοιπόν το Process Hacker και το αφήνουμε να τρέξει για μερικά δευτερόλεπτα. Στο παράθυρο του προγράμματος θα εμφανιστεί σχεδόν αμέσως μια λίστα με τις διεργασίες του συστήματος. Η λίστα αυτή λοιπόν, προβάλλεται με μια ιεραρχημένη δομή. Έτσι εκτός από τα ονόματα των διεργασιών βλέπουμε αμέσως ποιες διεργασίες δημιουργήθηκαν από το σύστημα ή από άλλες διεργασίες και αντίστοιχα ποιες διεργασίες είναι εκείνες που έχουν δημιουργήσει άλλες υποδιεργασίες. Για παράδειγμα όπως βλέπουμε και στη παρακάτω εικόνα το



Εικόνα 6.2: Το bot παρουσιάζεται με το όνομα ieexplore.exe

explorer.exe έχει δημιουργήσει τις διεργασίες VboxTray.exe, ProcessHacker.exe κ.λπ. Βέβαια το δυνατό σημείο αυτού του εργαλείου δεν είναι οι διεργασίες που τρέχει. Αυτό που μας ενδιαφέρει και θέλουμε να δούμε είναι με ποιον τρόπο το κακόβουλο πρόγραμμα κρύβει τη δική του διεργασία από τα μάτια του χρήστη. Με το παράθυρο του Process Hacker ανοικτό, εκτελούμε το κακόβουλο πρόγραμμα. Όπως φαίνεται και από την παραπάνω εικόνα το πρόγραμμα θα χρησιμοποιεί και πάλι ένα όνομα υπεράνω κάθε υποψίας. Στη δικιά μας περίπτωση η διεργασία του κακόβουλου προγράμματος ονομαζόταν iexplore.exe.

3ο Βήμα: Επισκόπηση αλλαγών σε registry και συστήματα αρχείων

Το registry αποτελεί μια ιεραρχική βάση δεδομένων που αποθηκεύει τις ρυθμίσεις και τις επιλογές από όλα τα λειτουργικά συστήματα της οικογένειας των Windows. Αυτή η βάση περιέχει πληροφορίες και ρυθμίσεις για τα χαμηλού επιπέδου λειτουργικά συστήματα και για τις εφαρμογές που τρέχουν στη πλατφόρμα που έχουν επιλεγεί για να χρησιμοποιούν τη registry. Ο πυρήνας, τα drivers των συσκευών, οι υπηρεσίες, οι εφαρμογές μπορούν να χρησιμοποιούν τη registry. Το registry επίσης παρέχει ένα τρόπο να προσεγγιστούν οι μετρητές για την σκιαγράφηση της απόδοσης συστήματος. Το registry περιέχει δύο βασικά στοιχεία: τα κλειδιά και τις τιμές (keys και values αντίστοιχα). Τα κλειδιά περιέχουν αντικείμενα στα οποία αποθηκεύονται διάφορες τιμές, όπως ακριβώς γίνεται και με τους φακέλους και συντάσσοντα παρόμοια με τα ονόματα διαδρομών (path name) των Windows χρησιμοποιώντας το backslash για να δείξουν τα επίπεδα ιεραρχίας. Οι τιμές είναι τα στοιχεία, τα δεδομένα ή κάποια χαρακτηριστικά που αποθηκεύονται εντός των κλειδιών όπως είναι για παράδειγμα τα αρχεία. Έτσι λοιπόν ένα κλειδί μπορεί να περιέχει τιμές αλλά και πρόσθετα (υπο)κλειδιά, τα οποία με τη σειρά τους μπορούν να περιέχουν τιμές ή πρόσθετα κλειδιά κ.ο.κ. Το registry είναι αποθηκευμένο σε διάφορες θέσεις του σκληρού δίσκου, ανάλογα με την έκδοση του λειτουργικού. Μια ιδιαίτερα συνηθυσμένη διαδρομή για την αποθήκευση των αρχείων του registry είναι η:

C:\Windows\System32\config

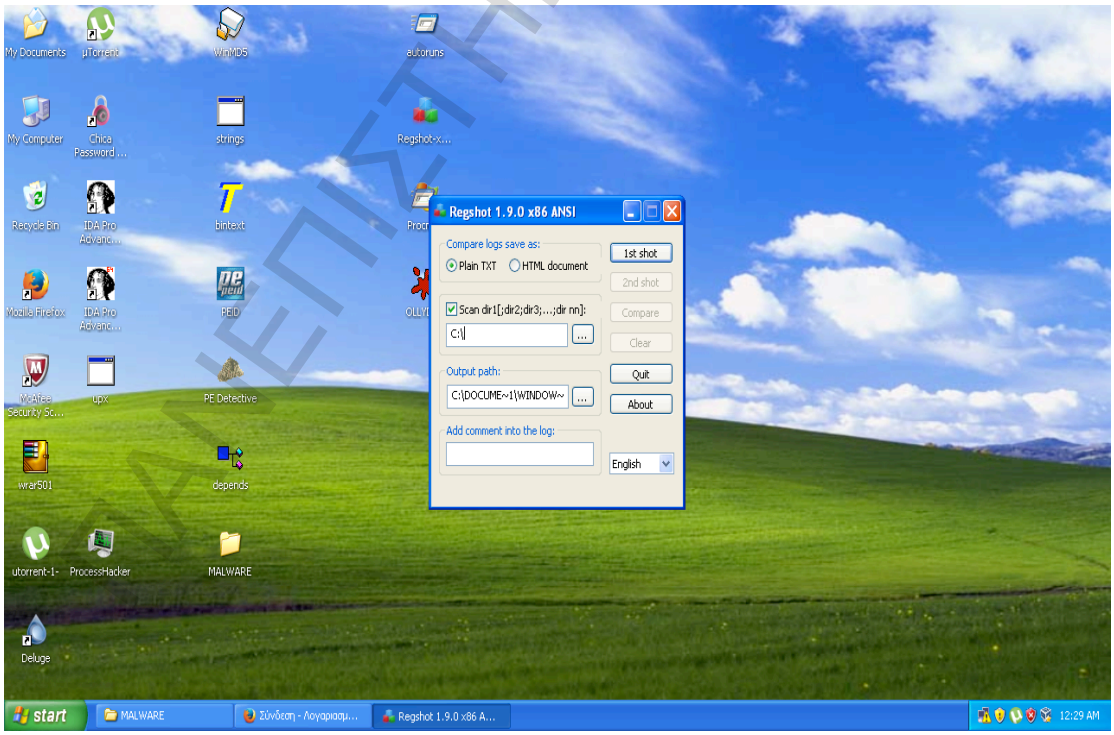
Επίσης, οι ρυθμίσεις που αφορούν σε ένα συγκεκριμένο χρήστη του υπολογιστή βρίσκονται στο αρχείο:

C:\Users\όνομα_χρήστη\System32\ntuser.dat

Τα περισσότερα malware επιθυμούν να επέμβουν στο registry του συστήματος. Για παράδειγμα, ένα malware μπορεί να θέλει να αποθηκεύσει εκεί ορισμένες ρυθμίσεις του ή και να προσθέσει τον εαυτό του σε κάποιο κλειδί του συστήματος (π.χ σε κάποιο κλειδί που περιέχει προγράμματα τα οποία εκτελούνται αυτόματα). Πέρα από το registry ένα malware ενδέχεται να τροποποιεί διάφορα άλλα αρχεία, όπως επίσης να διαγράφει ή και να δημιουργεί καινούργια.

Για να δούμε όλες αυτές τις αλλαγές στο σύστημα θα πρέπει να χρησιμοποιήσουμε τα εργαλεία Regshot και Process Monitor. Για αρχή όμως θα πρέπει πρώτα να επιστρέψουμε πάλι στο καθαρό snapshot της εικονικής. Ξεκινάμε λοιπόν με το εργαλείο Regshot, το οποίο αποτελεί μια ιδιαίτερα χρήσιμη εφαρμογή. Το συγκεκριμένο εργαλείο μας επιτρέπει να κρατήσουμε στιγμιότυπα του registry, αλλά και ολόκληρου του συστήματος αρχείων, αλλά και να συγκρίνουμε τις αλλαγές που έχουν γίνει στο Windows registry πριν και μετά την εκτέλεση του κακόβουλου προγράμματος και να συγκρίνουμε και να αναλύσουμε τα στιγμιότυπα μεταξύ τους.

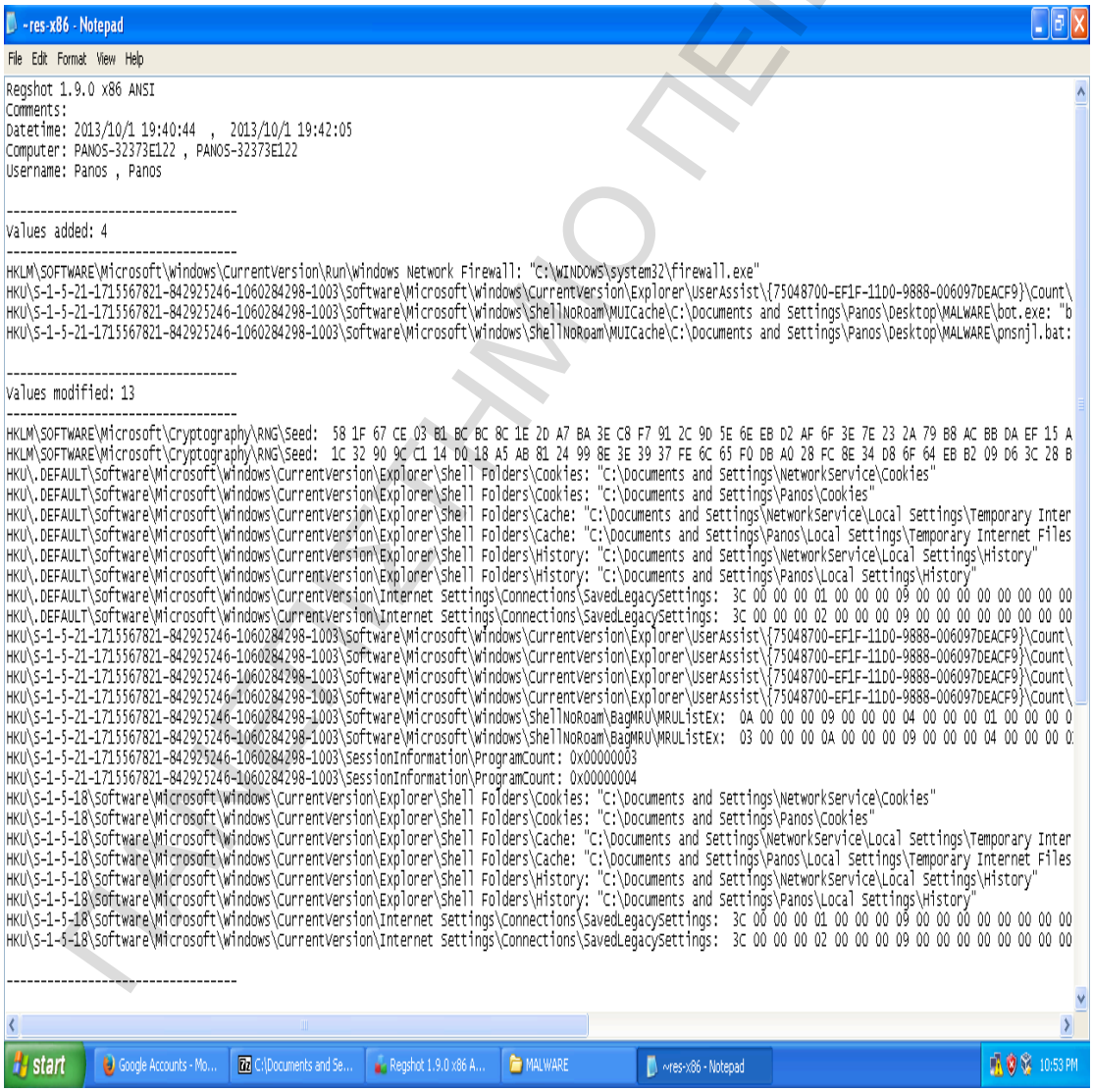
Αυτό που θα κάνομε λοιπόν είναι να κρατήσουμε ένα στιγμιότυπο πριν και ένα στιγμιότυπο μετά την εκτέλεση του κακόβουλου προγράμματος ώστε να δούμε αν και ποιες αλλαγές έγιναν στο σύστημα. Ανοίγουμε λοιπόν το Regshot και πατάμε στο κουμπάκι scan dir1. Από κάτω γράφουμε “C:\ ” για να δηλώσουμε στο πρόγραμμα τι θέλουμε να κρατήσει. Ακολούθως πατάμε στο 1st shot-> Shot (πάνω δεξιά).



Εικόνα 6.3: *Αρχική οθόνη του Regshot*

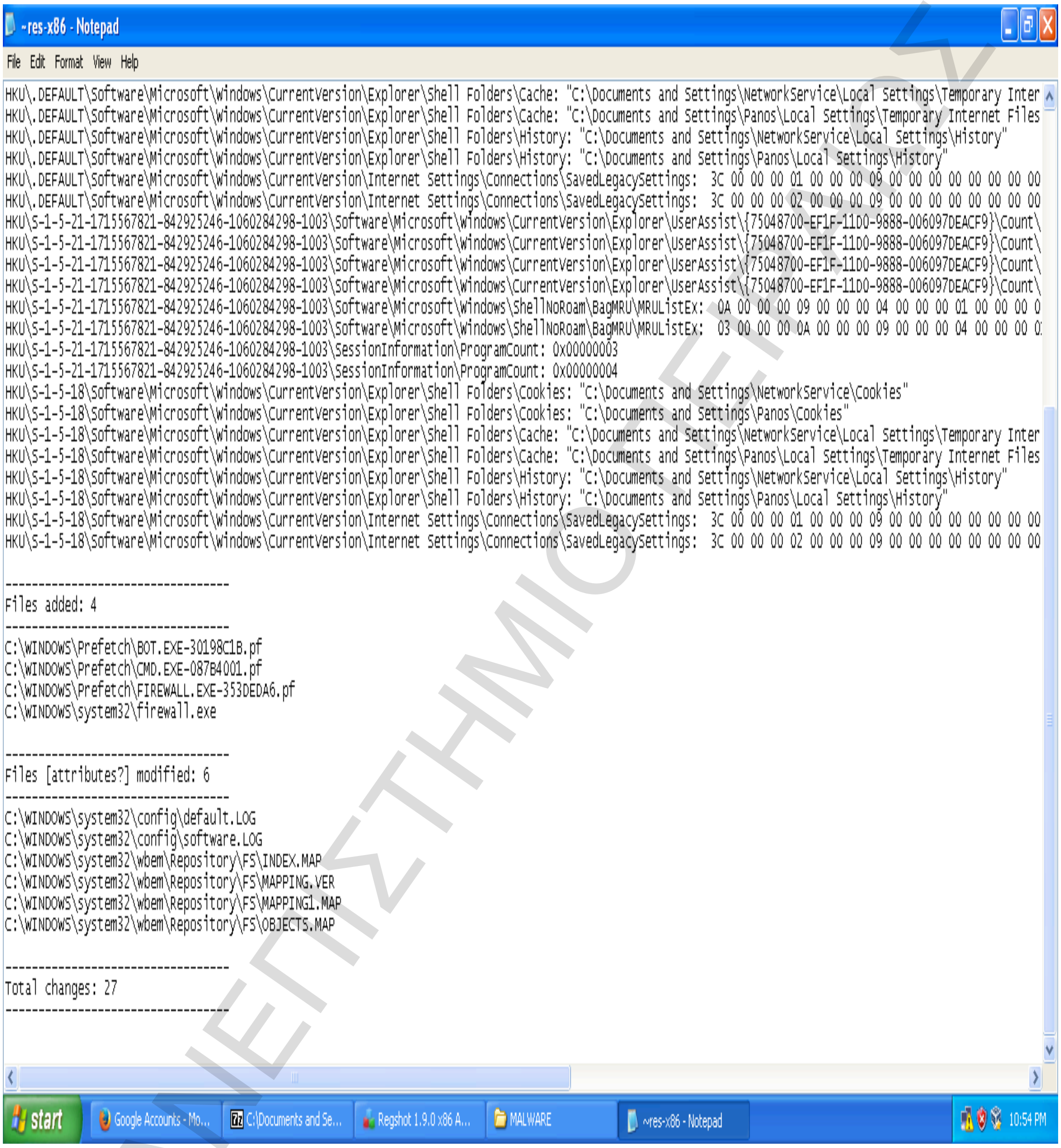
Μετά από λίγα δευτερόλεπτα το πρόγραμμα θα έχει αποθηκεύσει επιτυχώς την τρέχουσα κατάσταση του δίσκου. Σε αυτό το σημείο θα εκτελέσουμε το κακόβουλο πρόγραμμα και κατόπιν θα πάρουμε το δεύτερο στιγμιότυπο, πατώντας στο κουμπί 2nd shot. Θα πρέπει όμως να περιμένουμε λίγο (ένα με δύο λεπτά) ώστε το κακόβουλο πρόγραμμα να ολοκληρώσει την όποια δραστηριότητά του. Τέλος αφού έχουμε πάρει και το δεύτερο στιγμιότυπο, πατάμε στο κουμπί Compare, ώστε να πραγματοποιηθεί μια σύγκριση μεταξύ των δύο στιγμιοτύπων.

Το Regshot μας δίνει τη δυνατότητα να σώσουμε τα αποτελέσματα σε μορφή κειμένου ή σε μορφή HTML αρχείου με σκοπό να εντοπίσουμε γρηγορότερα τις αλλαγές που πραγματοποιήθηκαν.



Εικόνα 6.4 : Στην ενότητα values added η πρώτη εγγραφή δείχνει ότι το malware πρόσθεσε το δήθεν χρήσιμο αρχείο που δημιουργεί (firewall.exe) στη λίστα με τα προγράμματα που ξεκινούν αυτόματα.

Τα στοιχεία που βλέπουμε στη παρακάτω αλλά και στην παραπάνω εικόνα δεν αποτελούν κατά ανάγκη το αποτέλεσμα κάποιας κακόβουλης δραστηριότητας. Υπάρχουν αρκετά στοιχεία στο registry που μεταβάλλονται αρκετά συχνά.

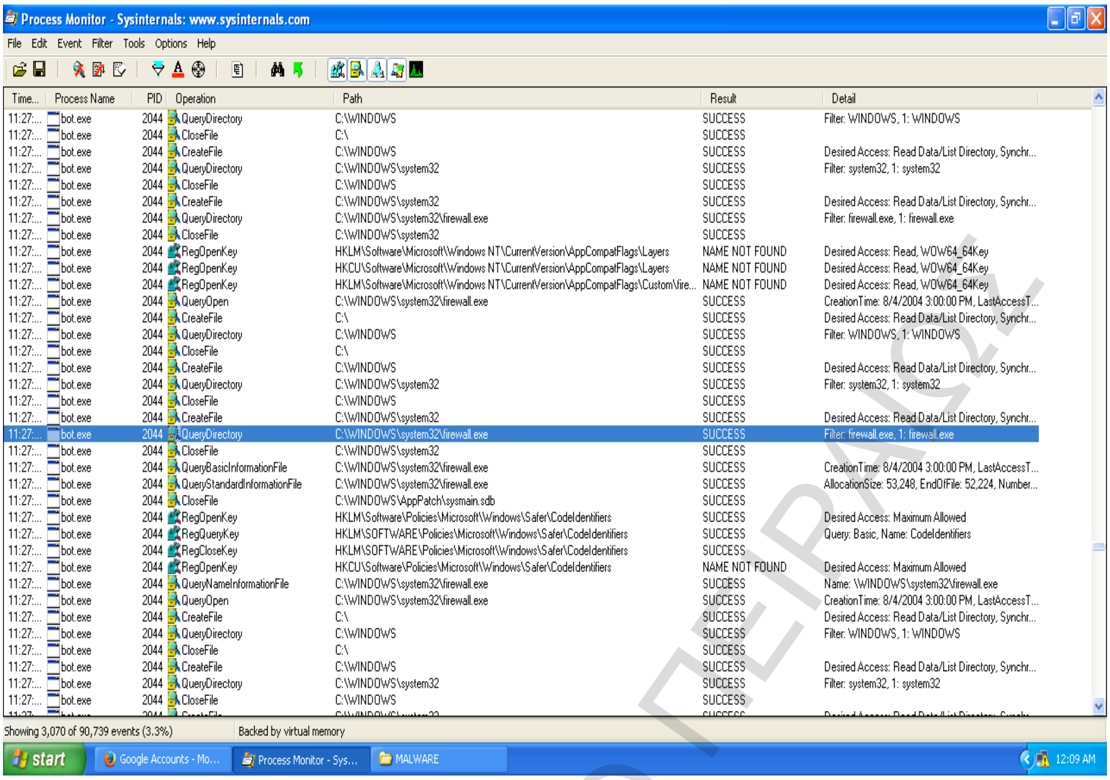


Εικόνα 6.5 : Επιπλέον στην ενότητα files added επιβεβαιώνουμε την ύπαρξη του αρχείου bot.exe μέσα στο φάκελο Windows\System32.

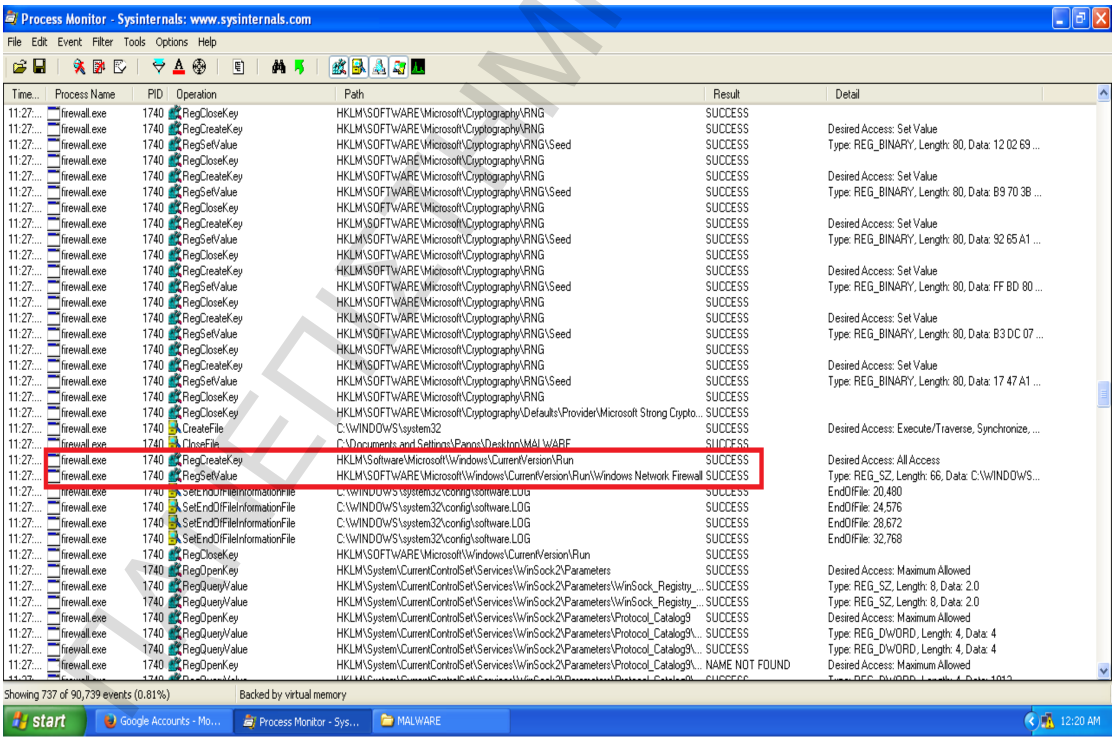
Το Process Monitor αποτελεί ένα προηγμένο εργαλείο ελέγχου για τα Windows που παρουσιάζει σε πραγματικό χρόνο το σύστημα αρχείων. Πρόκειται για ένα εργαλείο που φτιάχτηκε αρχικά από την εταιρεία Wininternals και πλέον ανήκει στη σουίτα Windows Sysinternals. Συνδυάζει τα δύο παλαιότερα εργαλεία FileMon και RegMon και χρησιμοποιείται επίσης στα computer forensics.

Επίσης το Process Monitor ελέγχει και καταγράφει όλες τις ενέργειες που γίνονται ενάντια του Registry των Windows. Μπορεί ακόμα να χρησιμοποιηθεί για να ανιχνεύσει τις αποτυχημένες προσπάθειες, να διαβάσει και να γράψει τα registry keys. Επιτρέπει επίσης το φιλτράρισμα στα συγκεκριμένα κλειδιά, τις διαδικασίες, τα IDs, και τις τιμές (values). Επιπλέον επιδεικνύει πώς οι εφαρμογές χρησιμοποιούν τα αρχεία, τα DLLs και ανιχνεύει ορισμένα κρίσιμα λάθη στα αρχεία συστημάτων κ.α. Με τα φίλτρα μπορούμε να δώσουμε οδηγίες στην εφαρμογή, ώστε να εμφανίσει μόνο τις ενέργειες που εκτελούνται από κάποιο συγκεκριμένο πρόγραμμα ή διεργασία.

Για μια ακόμα φορά πρέπει πρώτα να επαναφέρουμε την εικονική μηχανή στην αρχική κατάσταση χρησιμοποιώντας το καθαρό snapshot. Αφού το κάνουμε ανοίγουμε το Process Monitor και τρέχουμε το malware. Μετά από λίγο σταματάμε την καταγραφή πατώντας το τρίτο κουμπί από τα αριστερά (αυτό με τον μεγεθυντικό φακό). Αυτή τη στιγμή βλέπουμε μπροστά μας μια λίστα με εκατοντάδες γραμμές, καθεμιά από τις οποίες αντιστοιχεί σε κάποια ενέργεια, κάποιας διεργασίας. Για κάθε μια από αυτές τις διεργασίες βλέπουμε τη χρονική στιγμή που εκτελέστηκε, το ID, το είδος της ενέργειας που πραγματοποιήθηκε, την αντίστοιχη διαδρομή, το αρχείο καθώς και πολλές άλλες πληροφορίες.



Εικόνα 6.6: Σε αυτή την εκτέλεση το malware sample καμουφλάρεται ως firewall



Εικόνα 6.7: Το δήθεν firewall.exe πρόσθεσε τον εαυτό του στα προγράμματα που ξεκινούν αυτόματα τροποποιώντας κατάλληλα το registry του συστήματος.

Στο δικό μας παράδειγμα η νέα αυτή διεργασία έχει το όνομα firewall.exe. Όπως φαίνεται και στις παραπάνω δυο εικόνες, το κακόβουλο πρόγραμμα προσθέτει τον εαυτό του στα προγράμματα που ξεκινούν αυτόματα με το σύστημα.

4ο Βήμα: Καταγραφή και ανάλυση κακόβουλης δικτυακής κίνησης

Το τελευταίο βήμα στη δυναμική ανάλυση του κακόβουλου λογισμικού είναι η καταγραφή της δικτυακής δραστηριότητας. Αυτό που μας ενδιαφέρει να μάθουμε είναι πώς το κακόβουλο λογισμικό επικοινωνεί με το δημιουργό του, ώστε να είμαστε σε θέση να κατανοήσουμε την αποστολή του. Τα εργαλεία που θα χρησιμοποιήσουμε είναι το FakeNet και το Wireshark.

Το Wireshark πρόκειται για ένα κορυφαίο εργαλείο ανάλυσης της δικτυκής κίνησης. Είναι παρόμοιο με το πρόγραμμα tcpdump, όμως έχει γραφικό front-end και πολλές περισσότερες επιλογές ταξινόμησης και φιλτραρίσματος. Το tcpdump είναι ένα εργαλείο το οποίο γράφτηκε το 1987 από τον Van Jacobson, Craig Leres και τον Steven McCanne. Χρησιμοποιείται επίσης για να δούμε τις επικοινωνίες ενός άλλου χρήστη ή υπολογιστή. Η επίσημη ιστοσελίδα είναι η www.tcpdump.org . Μαζί με το tcpdump διατίθεται και η βιβλιοθήκη libpcap στην οποία βασίζεται το Wireshark.

Το Wireshark επιτρέπει στο χρήστη να παρακολουθήσει όλη την κίνηση που γίνεται στο δίκτυο. Το Wireshark αποτελεί ένα ελεύθερο και ανοιχτού κώδικα λογισμικό ανάλυσης πρωτοκόλλων δικτύου υπολογιστών. Χρησιμοποιείται για ανάλυση δικτύου, παρακολούθηση δικτύου, εντοπισμό και αντιμετώπιση προβλημάτων στα δίκτυα και για εκπαίδευση. Το αρχικό όνομα του προγράμματος ήταν Ethereal, το οποίο αναπτύχθηκε από ένα φοιτητή του πανεπιστημίου του Μιζούρη και τον Μάιο του 2006 άλλαξε σε Wireshark για λόγους εμπορικών σημάτων.

Είναι διαθέσιμο για όλα τα κύρια λειτουργικά συστήματα όπως τα Windows, Linux, Mac OS X, Solaris, BSD και χρησιμοποιεί το GTK+ για το γραφικό περιβάλλον και το Pcap για capture πακέτων. Το Wireshark εκδίδεται από την GNU GPL, πράγμα το οποίο σημαίνει ότι ο κώδικάς του είναι δωρεάν και ανοικτός για το καθένα.

Το FakeNet, από την άλλη δεν είναι και τόσο γνωστό. Δημιουργήθηκε από τον Andrew Honig (συγγραφέας του πολύ καλού βιβλιου Practical Malware Analysis) Το FakeNet, δημιουργεί ένα ψεύτικο δίκτυο με ψεύτικες υπηρεσίες (π.χ DNS, HTTP server, SSL κ.α). Το εργαλείο αυτό εξομειώνει ένα δίκτυο έτσι ώστε το κακόβουλο πρόγραμμα να αλληλεπιδράσει με ένα απομακρυσμένο χρήστη και να συνεχίσει να

τρέχει έτσι ώστε να επιτρέψει στον αναλυτή να παρακολουθήσει τη δραστηριότητα του δικτύου μέσα από ένα ασφαλές περιβάλλον. Ο στόχος του προγράμματος είναι:

• Να είναι εύκολο να εγκατασταθεί και να χρησιμοποιηθεί. Το εργαλείο αυτό τρέχει σε Windows .

• Υποστηρίζει τα πιο κοινά προτόκολλα που χρησιμοποιούν τα malware.

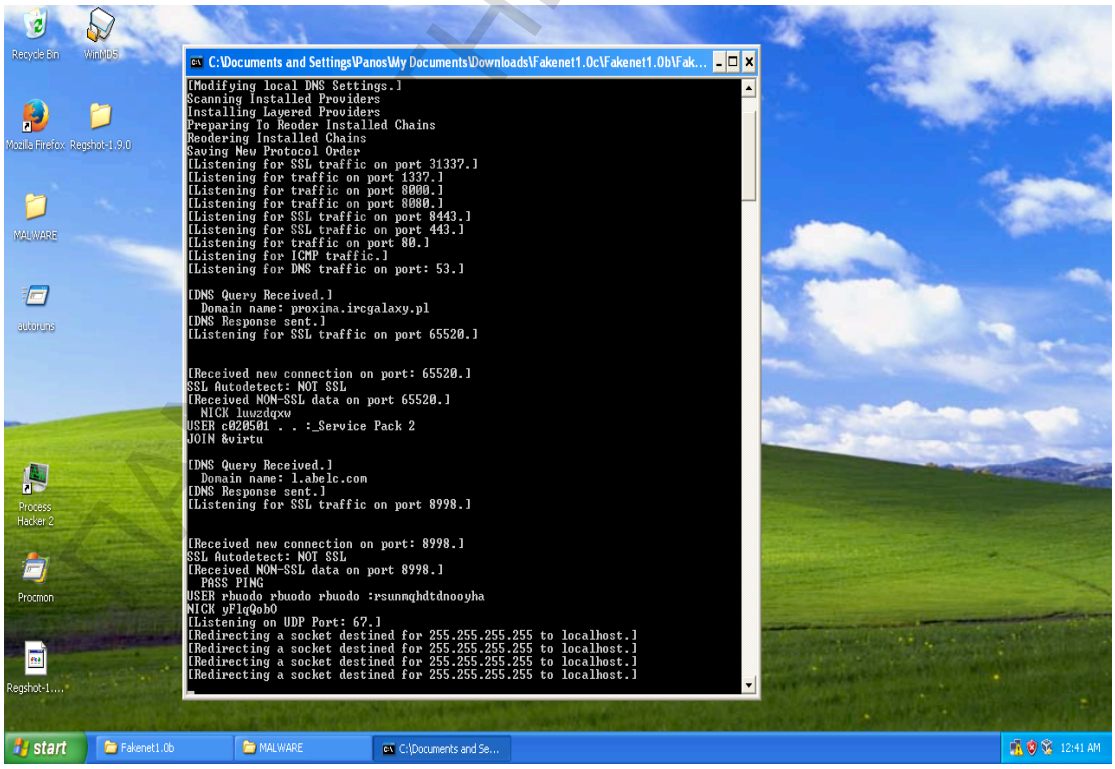
• Εκτελεί όλη τη δραστηριότητα στο τοπικό μηχάνημα για να αποφύγει την ανάγκη για ένα δεύτερο μηχάνημα σε εικονική μηχανή.

• Παρέχει επεκτάσεις της python για τη προσθήκη νέων ή custom προτοκόλλων.

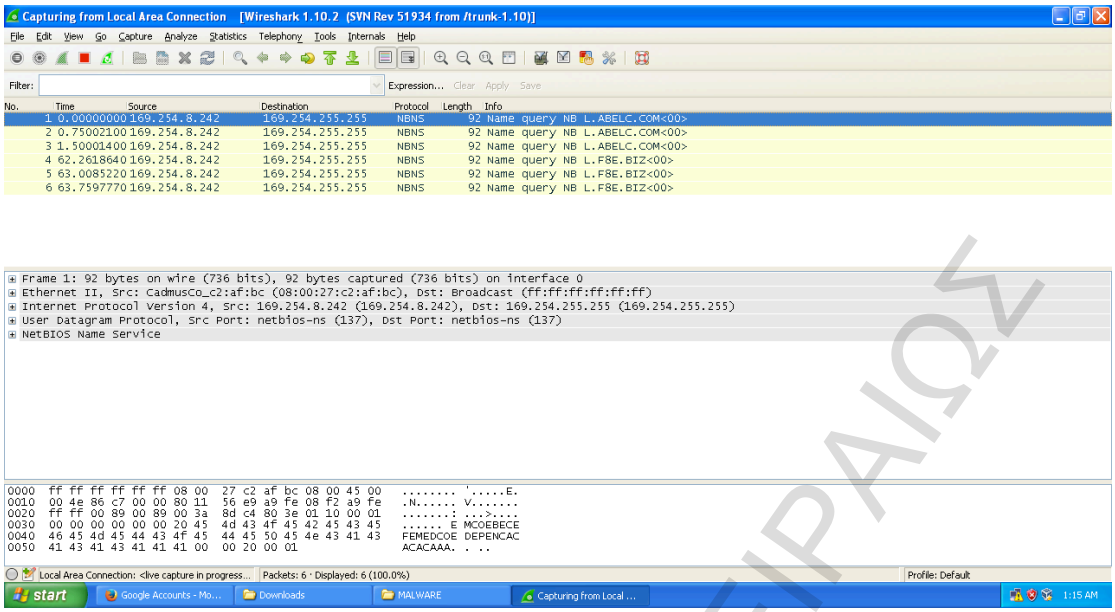
• Κρατάει το malware ενώ τρέχει έτσι ώστε να μπορούμε να παρατηρήσουμε όσες το δυνατόν περισσότερες λειτουργίες είναι δυνατόν.

• Διαθέτει ιδιαίτερα εύκαμπτη διαμόρφωση (configuration).

Κατά τα γνωστά τώρα ξεκινάμε φορτώνοντας το καθαρό snapshot της εικονικής μηχανής. Μετά ξεκινάμε το FakeNet και εκτελούμε το malware sample. Αυτόματα βλέπουμε να εμφανίζεται στο παράθυρο της εικόνας η δικτυακή δράση του αρχείου. Όπως φαίνεται και στην εικόνα 12, το bot αναζητά δυο domain names (proxima.ircgalaxy.pl και 1.abelc.com). Το FakeNet απαντάει και το bot συνδέεται σε διαφορετικά port σε κάθε server.



Εικόνα 6.8: Το FakeNet μόλις εντόπισε τις προσπάθειες του κακόβουλου αρχείου να συνδεθεί σε ένα IRC botnet.



Εικόνα 6.9: Το Wireshark αποτελεί και αυτό ένα πολύ καλό εργαλείο. Εδώ βλέπουμε την καταγραφή των NETBIOS request, για τα κακόβουλα domans που χρησιμοποιεί το bot.

Συμπεράσματα

Το κακόβουλο πρόγραμμα που εξετάσαμε προσπαθεί να συνδεθεί σε ένα IRC botnet. Επίσης από ότι βλέπουμε στην εικόνα 12 το κακόβουλο πρόγραμμα στέλνει και ορισμένες εντολές (JOIN, PART) οι οποίες αποτελούν εντολές του πρωτοκόλλου IRC.

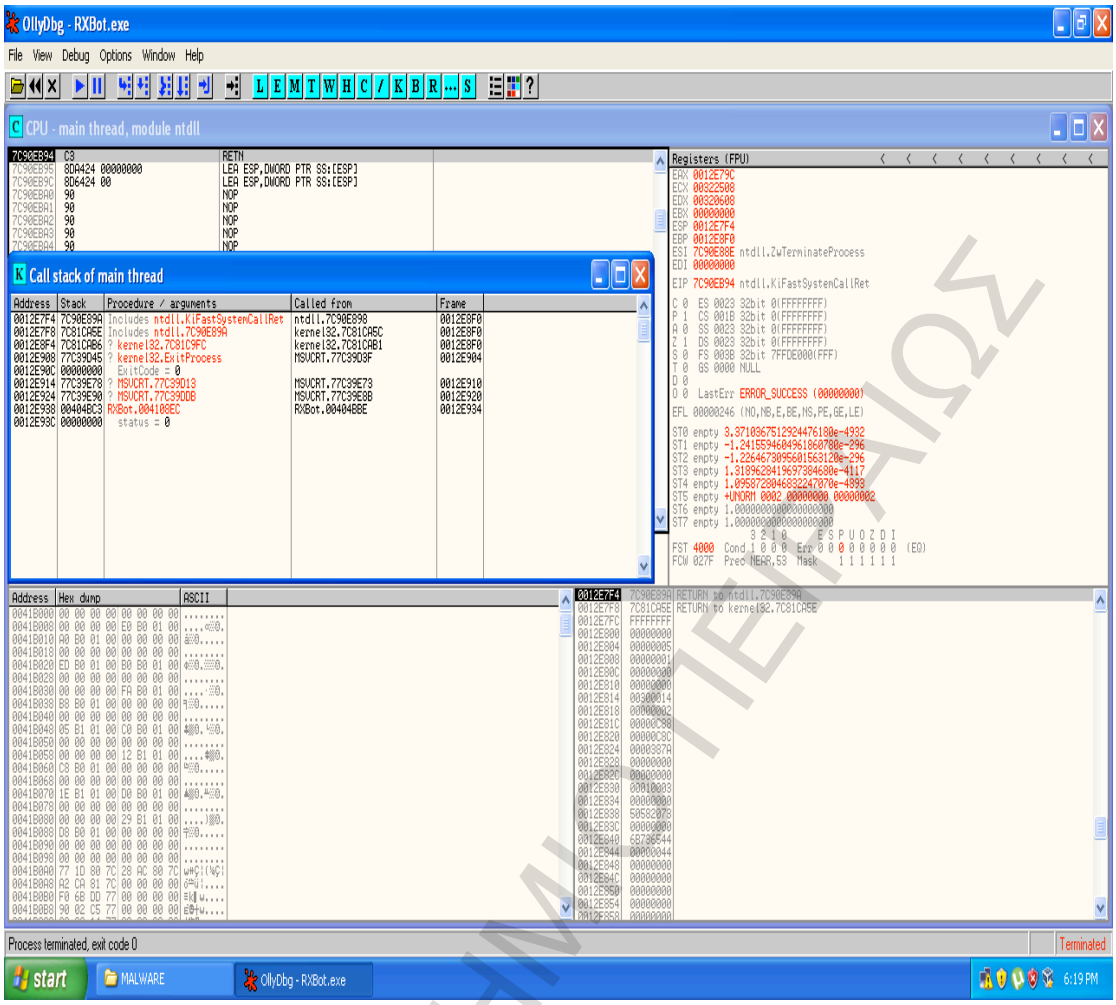
7. Προχωρημένη δυναμική ανάλυση malware

Το προχωρημένο στάδιο περιλαμβάνει τη φόρτωση και την εκτέλεση του μεταγλωττισμένου αρχείου σε ένα debugger. Ο debugger είναι ένα κομμάτι του λογισμικού ή του υλικού που χρησιμοποιείται για να εξετάσει την εκτέλεση ενός άλλου προγράμματος. Οι debuggers βοηθούν στο στάδιο της ανάπτυξης του λογισμικού, δεδομένου ότι τα προγράμματα έχουν συνήθως τα λάθη όταν γράφονται αρχικά. Επίσης οι debuggers μας επιτρέπουν να δούμε τη τιμή της κάθε μνήμης και του κάθε καταχωρητή σε κάθε συνάρτηση. Επιπλέον μας αφήνουν να αλλάξουμε οτιδήποτε σχετικό με την εκτέλεση του προγράμματος οποιαδήποτε χρονική στιγμή. Για παράδειγμα μπορούμε να αλλάξουμε την τιμή μιας ενιαίας μεταβλητής σε οποιοδήποτε σημείο. Το μόνο που χρειαζόμαστε είναι κάποιες πληροφορίες για εκείνη τη μεταβλητή, συμπεριλαμβανομένης της θέσης του.

7.1 Olly debugger

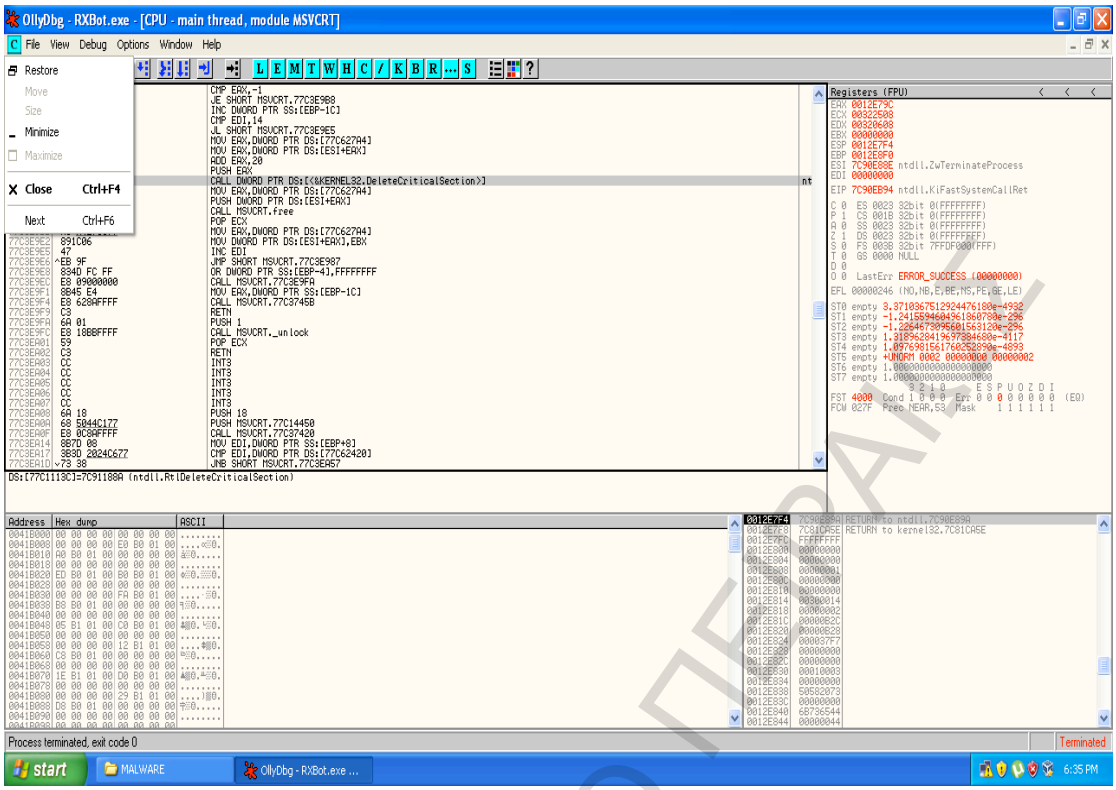
Ο OllyDbg είναι ένας debugger x86 αρχιτεκτονικής τον οποίο χρησιμοποιούμε όταν δεν έχουμε διαθέσιμο τον πηγαίο κώδικα (source code). Ο OllyDbg χρησιμοποιείται συχνά για το reverse engineering των προγραμμάτων από τους crackers ώστε να σπάσουν το λογισμικό που έχουν δημιουργήσει άλλοι developers. Πρόκειται για έναν από τους πιο γνωστούς debuggers λόγω της ευκολίας χρήσης του. Επίσης είναι ιδιαίτερα χρήσιμος και για τους προγραμματιστές ώστε να μπορέσουν να εξασφαλίσουν ότι το πρόγραμμά τους τρέχει όπως ακριβώς θα έπρεπε. Φυσικά χρησιμοποιείται και από τους αναλυτές malware.

Αφού επιστρέψουμε για μια ακόμα φορά στο καθαρό snapshot τρέχουμε τον OllyDbg ώστε να μελετήσουμε τον αλγόριθμο του κακόβουλου προγράμματος. Αφού το αφήσουμε για λίγο να τρέξει επιλέγουμε το ανοιχτό παράθυρο του debugger και πατάμε [F12] (pause), ώστε να διακόψουμε την εκτέλεση του προγράμματος. Αμέσως μετά μπαίνουμε στο παράθυρο με το όνομα «Call stack of main thread».



Εικόνα 7: Παρατηρούμε ότι πριν την εκτέλεση του προγράμματός μας είχε μεταβεί στη διεύθυνση MSVCRT.77C39DDB.

Στην εικόνα 7 βλέπουμε ότι πριν τη διακοπή το πρόγραμμα είχε μεταβεί στη διεύθυνση *MSVCRT.77C39DDB.* Εδώ ιδιαίτερα σημαντικές είναι οι στήλες Procedure και Called From. Η πρώτη μας δείχνει το όνομα της συνάρτησης που καλείται και η δεύτερη δείχνει από ποια διαδικασία καλείται. Υπάρχουν διάφορες διεργασίες όπως είναι οι kernel32 και η Rxbot. Η πρώτη διεργασία δεν μας ενδιαφέρει αφού είναι του λειτουργικού. Συνεπώς ασχολούμαστε με την Rxbot. Όπως φαίνεται και από το όνομά της η Rxbot αφορά στο malware μας. Αφού την ανοίξουμε με διπλό κλικ μεταφερόμαστε μέσα στο main thread της. Απο τη στιγμή που μεταφερόμαστε στο main thread ψάχνουμε τις εντολές για να δούμε τι θα μπορέσουμε να βρούμε.



Εικόνα 7.1: Εδώ βλέπουμε τη ακριβώς κάνει το κακόβουλο πρόγραμμά μας

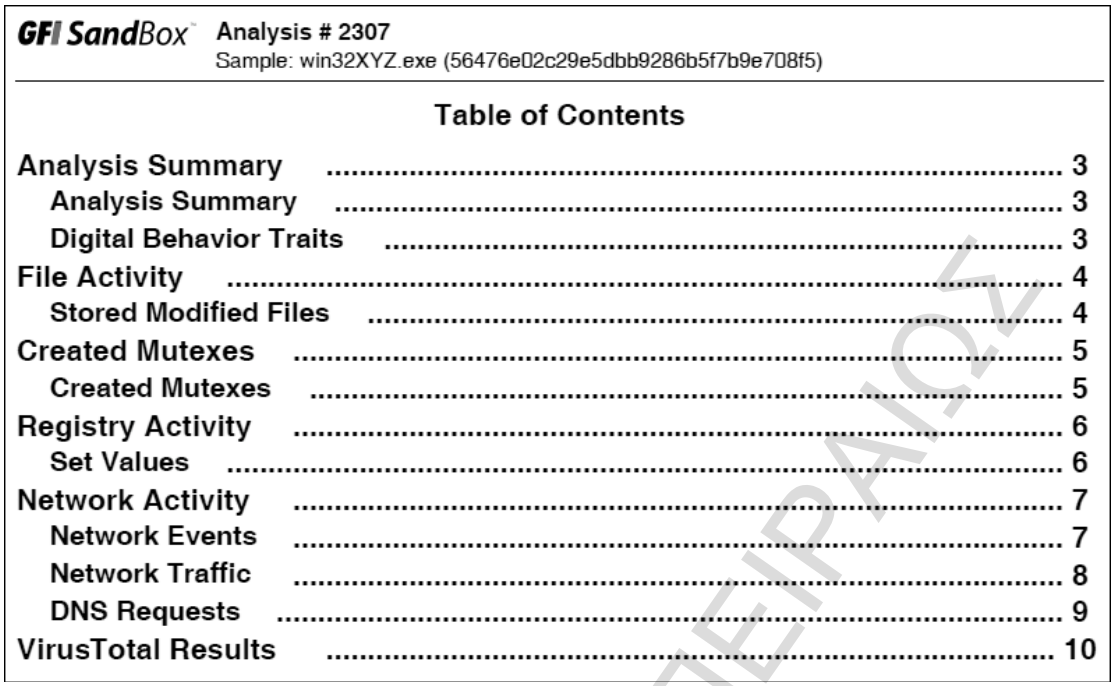
Στην εικόνα 7.1 βλέπουμε λοιπόν σε ποια από τις περιοχές του προγράμματος γίνεται η διαγραφή μιας κρίσιμης συνάρτησης του λειτουργικού. Αυτή λοιπόν είναι η περιοχή που είναι ζητούμενη για τους αναλυτές κακόβουλων προγραμμάτων.

8. Προστατευτικά κουτιά

Στο συγκεκριμένο κεφάλαιο θα ασχοληθούμε με τα προστατευτικά κουτιά (sandbox). Ένα sandbox στην ασφάλεια υπολογιστών είναι ένας μηχανισμός ασφάλειας για την εκτέλεση προγραμμάτων για τα οποία δεν γνωρίζουμε την προέλευσή τους. Χρησιμοποιούνται συχνά για να εκτελέσουμε μη δοκιμασμένο κώδικα, ή μη έμπιστα προγράμματα από τρίτους, ή από μη έμπιστους προμηθευτές, ή από μη εμπιστους χρήστες ή ακόμα και από μη έμπιστους ιστοχώρους. Σε ένα sandbox μπορούν να προστεθούν λειτουργίες όπως για παράδειγμα είναι η προσομοίωση ορισμένων δικτυακών υπηρεσιών με σκοπό να ξεγελάσουμε το κακόβουλο πρόγραμμα και να νομίσει ότι εκτελείται σε ένα κανονικό υπολογιστή.

Υπάρχουν πολλά έτοιμα malware sandboxes, που υποστηρίζονται από εταιρείες ανάλυσης malware. Γνωστά παραδείγματα αποτελούν τα Norman SandBox, GFI Sandbox, Anubis, Joe Sandbox, ThreatExpert, BitBlaze, καθώς και το Comodo Instant Malware Analysis. Το Norman SandBox καθώς και το GFI Sandbox (το παλιό CWS Sandbox) είναι τα πιο δημοφιλή προστατευτικά κουτιά. Η έξοδος που παρέχουν αυτά τα Sandboxes είναι σε ιδιαίτερα κατανοητή μορφή. Η ανάλυση που γίνεται από αυτές τις υπηρεσίες αποτελεί μια πολύ καλή αφετηρία για να υποβάλουμε το malware προς ανάλυση. Ακόμα και αν τα sandboxes είναι αυτοματοποιημένοι μηχανισμοί μπορούμε να επιλέξουμε να μην υποβάλουμε malware που περιέχει πληροφορίες επιχείρησεις σε ένα δημόσιο ιστότοπο.

Τα περισσότερα sandboxes δουλεύουν με σχεδόν παρόμοιο τρόπο. Στη συνέχεια βλέπουμε ένα παράδειγμα του πώς φαίνονται τα αποτελέσματα στο GFI Sandbox. Η αναφορά περιλαμβάνει ποικίλες λεπτομέρειες για το malware, όπως είναι η δραστηριότητα του δικτύου, τα αρχεία που δημιουργεί, τα αποτελέσματα που σκάναρε με το VirusTotal, κ.λπ.



GFI SandBox Analysis # 2307
Sample: win32XYZ.exe (56476e02c29e5dbb9286b5f7b9e708f5)

Table of Contents

Εικόνα 27: Αποτελέσματα ενός malware όπως μας το εμφανίζει το GFI Sandbox

8.1 Μειονεκτήματα των sandboxes

Δυστυχώς η χρήση των sandboxes έχει και ορισμένα μειονεκτήματα. Για παράδειγμα το sandbox απλά τρέχει το κακόβουλο πρόγραμμα όπως του το δίνουμε, χωρίς τα command-line options που είναι πιθανό να δέχεται. Συνεπώς αν αυτά τα command-line options δεν υπάρχουν δεν θα εκτελεστούν ποτέ. Επιπλέον, εάν το προς υποβολή κακόβουλο πρόγραμμα περιμένει ένα command-and-control πακέτο να επιστραφεί πρωτού εκτελεσθεί το backdoor, το backdoor δεν θα εκτελεσθεί εντός του sandbox.

Ένα άλλο μειονέκτημα είναι ότι το sandbox μπορεί να μην καταγράψει όλα τα γεγονότα και αυτό επειδή ούτε εμείς ούτε το sandbox μπορούμε να περιμένουμε αρκετό καιρό. Αυτό είναι απολύτως λογικό γιατί αν θέλουμε να εξετάσουμε μερικές δεκάδες ή εκατοντάδες δείγματα δεν μπορούμε να δίνουμε στο καθένα μεγάλο χρονικό διάστημα. Για παράδειγμα το κακόβουλο πρόγραμμα μπορεί να έχει τεθεί να κοιμηθεί για μια μέρα (να παραμείνει ανενεργό) προτού εκτελέσει τη κακόβουλη δραστηριότητά του. Συνεπώς δεν μπορούμε να δεσμεύσουμε ένα sandbox για μεγάλο χρονικό διάστημα.
Άλλα πιθανά μειονεκτήματα είναι και τα ακόλουθα:

• Τα κακόβουλα προγράμματα αντιλαμβάνονται συχνά αν εκτελούνται μέσα σε εικονική μηχανή ή σε πραγματικό μηχάνημα. Αν το κακόβουλο προγράμμα καταλάβει ότι βρίσκεται σε εικονική μηχανή τότε το κακόβουλο προγράμμα πιθανώς να σταματήσει να εκτελείται ή μπορεί να αρχίσει να συμπεριφέρται διαφορετικά με σκοπό να δυσκολέψει κατά τη διαδικασία της ανάλυσής του.

• Κάποια κακόβουλα προγράμματα για να δράσουν απαιτούν την ύπαρξη ορισμένων κλειδιών ή αρχείων στο σύστημα και αυτά τα αρχεία είναι πολύ πιθανόν να μην υπάρχουν στο sandbox. Τέτοιοι στόχοι μπορεί να είναι κλειδιά του registry.

• Εάν το κακόβουλο προόγραμμα είναι ένα DLL, ορισμένες λειτουργίες του δεν θα καλεσθούν κατάλληλα επειδή ένα DLL δεν θα τρέξει τόσο εύκολα όσο το εκτελέσιμο.

• Το περιβάλλον του sandbox αλλά και το λειτουργικό σύστημα μπορεί να μην είναι κατάλληλο για την εκτέλεση του κακόβουλου προγράμματος. Για παράδειγμα το κακόβουλου πρόγραμμα μπορεί να κρασάρει στα Windows XP ενώ στα Windows 7 να τρέχει κανονικά.

• Ένα sandbox δεν μπορεί να μας πει τι ακριβώς κάνει το κακόβουλο πρόγραμμα. Μπορεί να μας πει ποια είναι η βασική του λειτουργία αλλά δεν μπορεί να μας δώσει ποτέ ολοκληρωμένες πληροφορίες. Για παράδειγμα δεν μπορεί να μας πει αν αυτό που αναλύει είναι Trojan horse ή rootkit. Αυτά είναι συμπεράσματα που θα πρέπει να είμαστε σε θέση να ανακαλύψουμε μόνοι μας.

8.2 Cuckoo Sandbox

Το Cuckoo Sandbox είναι αρκετά γνωστό στο χώρο των αναλυτών κακόβουλου λογισμικού. Επισήμως αποτελεί ένα ανοικτού κώδικα σύστημα αυτοματοποιημένης ανάλυσης malware. Χρησιμοποιείται για την αυτόματη εκτέλεση, την ανάλυση διαφόρων αρχείων καθώς και για τη συλλογή και δημιουργία περιεκτικών αποτελεσμάτων ανάλυσης που περιγράφουν τι ακριβώς κάνει το malware όταν τρέχει σε ένα απομονωμένο λειτουργικό σύστημα όπως είναι τα Windows. Το Cuckoo μπορεί να ανακτήσει και να καταγράψει τους ακόλουθους τύπους αποτελεσμάτων:

• Τις κλήσεις συστήματος (win 32API) που πραγματοποιούνται από όλες τις διεργασίες που δημιούργησε το κακόβουλο πρόγραμμα.

• Όλα τα αρχεία που δημιουργήθηκαν, που διαγράφηκαν, που λήφθηκαν από το κακόβουλο πρόγραμμα κατά την εκτέλεσή του.

• Στιγμιότυπα της μνήμης (memory dump) RAM των διεργασιών του κακόβουλου λογισμικού.

• Τη δικτυακή κίνηση που καταγράφηκε η οποία αποθηκεύεται σε αρχείο PCAP (packet capture).

• Επίσης κατά τη διάρκεια εκτέλεσης του κακόβουλου προγράμματος λαμβάνονται screenshots από την επιφάνεια εργασίας.

• Πλήρη στιγμιότυπα της μνήμης RAM των μηχανημάτων που εκτελούν το κακόβουλο προγραμμα.

Το Cuckoo Sandbox ξεκίνησε ως project της Google το 2010 μαζί με το Honeynet. Σχεδιάστηκε και αναπτύχθηκε από τον Claudio nex Guarnieri, ο οποίος είναι ακόμα ο κύριος υπεύθυνος για την ανάπτυξη και αυτός που συντονίζει όλες τις προσπάθειες που γίνονται από τους υποστηρικτές και τους developers. Μετά από αυτή την αρχική εργασία η πρώτη beta έκδοση ανακοινώνεται στις 5 Φεβρουαρίου 2011 και διανέμεται για πρώτη φορά. Το Μάρτιο του 2012 κερδίζει το πρώτο γύρο του προγράμματος Magnificent 7 που οργανώθηκε από τη Rapid7 (ιδιοκτήτρια του Metasploit) η οποία έχει χρηματοδοτήσει το Cuckoo Sandbox. Έκτοτε έχουν ανακοινωθεί πολλές εκδόσεις και η τρέχουσα έκδοση (1.0) παρουσιάστηκε στης 9 Ιανουαρίου 2014.

Το Cuckoo Sandbox έχει σχεδιαστεί ώστε να μπορεί να χρησιμοποιηθεί και ως αυτόνομη εφαρμογή καθώς επίσης και να μπορεί να ενσωματωθεί σε μεγαλύτερα δίκτυα και όλα αυτά χάριν τις επεκτασιμότητας που υποστηρίζει. Μπορεί να χρησιμοποιηθεί για να αναλύσει διάφορα είδη αρχείων όπως:

• Εκτελέσιμα αρχεία εφαρμογών Windows

• Αρχεία DLL

• Αρχεία PDF και Microsoft Office

• Αρχεία URL και HTML

• PHP scripts

• Αρχεία CPL (Control Panel)

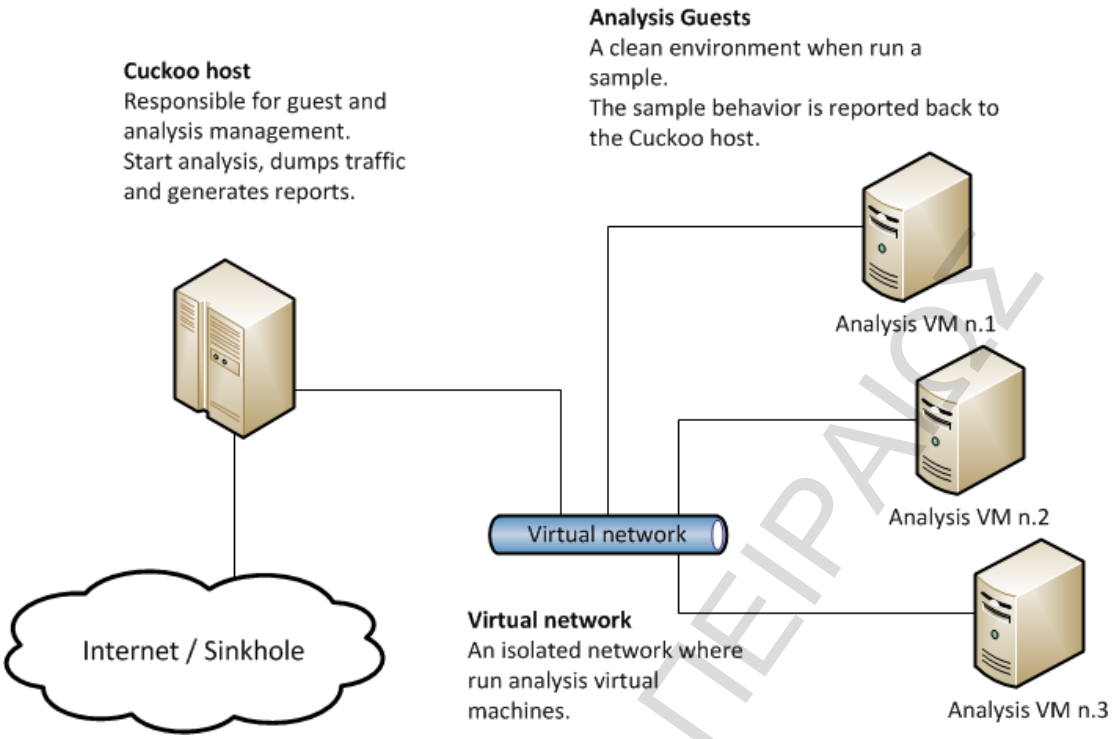
• Visual Basic (VB) scripts

• Zip αρχεία

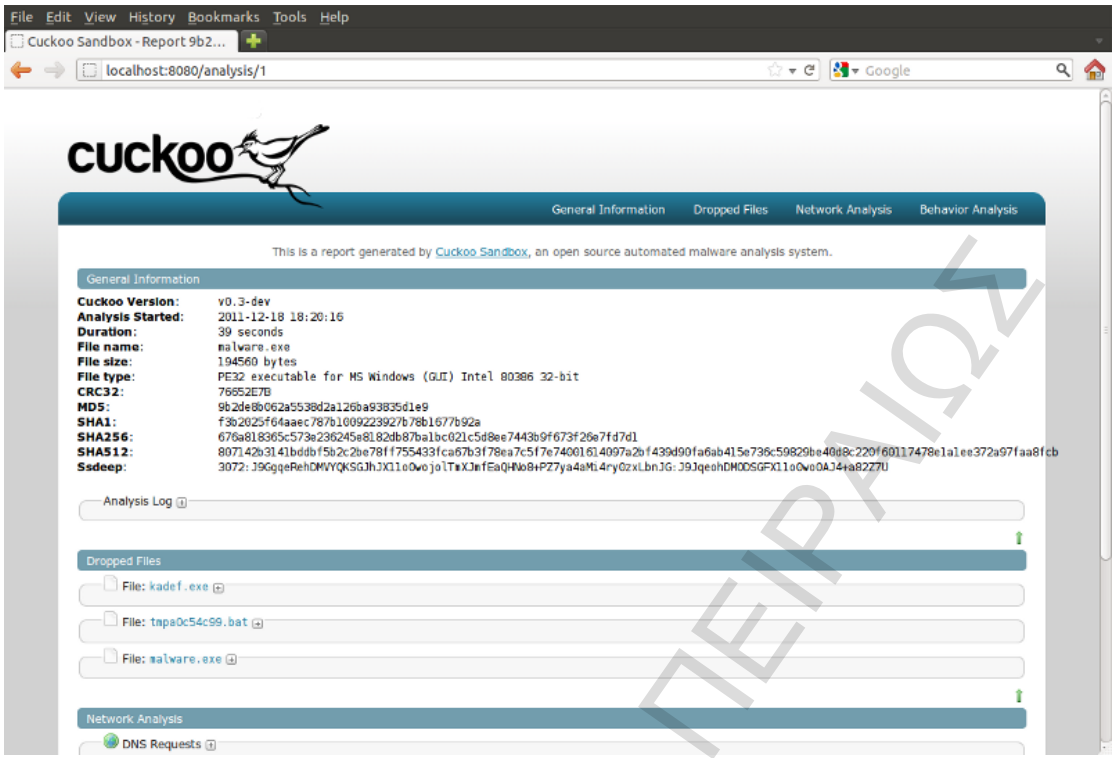
• Java Jar

• και σχεδόν οτιδήποτε άλλο μπορεί να εκτελεστεί.

Το Cuckoo Sandbox αποτελείται από ένα κεντρικό διοικητικό λογισμικό που χειρίζεται την εκτέλεση και ανάλυση δειγμάτων. Κάθε ανάλυση πραγματοποιείται σε ένα απομονωμένο μηχάνημα εικονικής μηχανής. Το Cuckoo χωρίζεται σε δύο τμήματα: Στον Cuckoo host και στους Cuckoo guests. Τα Cuckoo guests αποτελούν τις εικονικές μηχανές που γίνονται οι αναλύσεις των κακόβουλων προγραμμάτων. Το Cuckoo host είναι εκεί όπου ο αναλυτής στέλνει αυτά τα κακόβουλα προγράμματα. Η παρακάτω εικόνα δείχνει την αρχιτεκτονική του Cuckoo Sandbox.



Εικόνα 8: Η αρχιτεκτονική του Cuckoo Sandbox. Ένα Cuckoo host χειρίζεται διάφορα Cuckoo guests ως sandboxes, στα οποία γίνετα η ανάλυση των κακόβουλων προγραμμάτων.



Εικόνα 8.1: Ανάλυση κακόβουλου προγράμματος από το Cuckoo Sandbox. Όπως βλέπουμε καταγράφει αυτόματα ενδιαφέροντα στοιχεία του malware.

9. Περιοχή εγκατάστασης των malware

Όπως μπορούμε να καταλάβουμε και από τα παραπάνω η χρήση antivirus κρίνεται κάτι παραπάνω από απαραίτητη και μάλιστα ανεξαρτήτως λειτουργικού. Δυστυχώς όμως η χρήση του σε πολλές περιπτώσεις κρίνεται ανεπαρκής οπότε οφείλουμε να γνωρίζουμε τι πρέπει να κάνουμε σε τέτοιες περιπτώσεις .

Υπάρχουν κάποιοι συγκεκριμένοι προορισμοί που ένα κακόβουλο πρόγραμμα μπορεί να αποθηκευτεί. Αυτό βέβαια δεν σημαίνει ότι το κακόβουλο πρόγραμμα δεν μπορεί να αποθηκευτεί οπουδήποτε αλλού. Οι πιο συνηθισμένες περιοχές που μπορεί το κακόβουλο πρόγραμμα να κρυφτεί είναι οι εξής:

• Στο κατάλογο που βρίσκονται τα αρχεία του λειτουργικού συστήματος (**C: \Windows**).

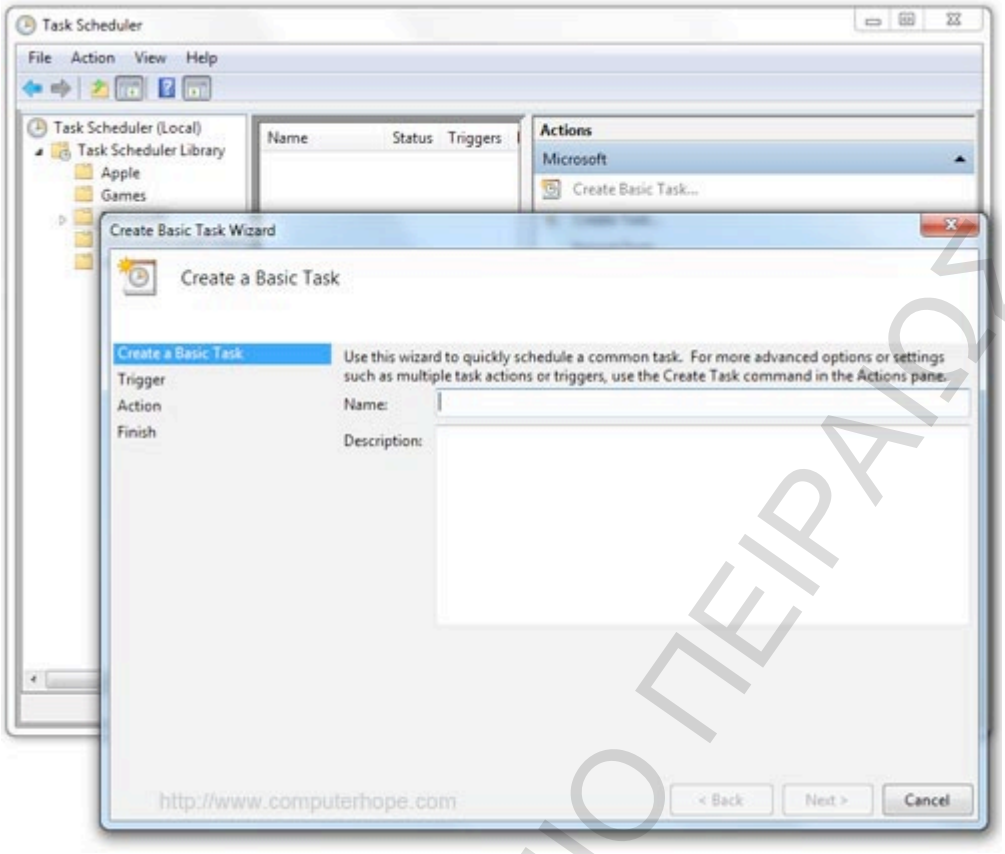
• Στο κατάλογο (**C:\Temp**) ο οποίος χρησιμοποιείται από το σύστημα για προσωρινή αποθήκευση. Βεβαια από την έκδοση των Vista και μετά η εγγραφή σε αυτό το κατάλογο απαγορεύεται χωρίς την έγκριση από το χρήστη. Για αυτόν το λόγο συνήθως τα κακόβουλα προγράμματα μπορούν να αποθηκευτούν στον κατάλογο

C:\Users<USER-NAME>\AppData\Local\Temp

Αυτός ο κατάλογος είναι αντίστοιχος του C:\Temp και η αντιγραφή μέσα σε αυτόν επιτρέπεται ελεύθερα.

9.1 Task Scheduler

Ο task scheduler είναι ένα πρόγραμμα το οποίο υπάρχει στα Windows και το οποίο επιτρέπει την προκαθορισμένη εκτέλεση προγραμμάτων. Για παράδειγμα εάν θέλουμε να εκτελέσουμε ένα πρόγραμμα κάθε Δευτέρα μπορούμε να ρυθμίσουμε τον scheduler task να τρέχει ο υπολογιστής μας αυτό το αρχείο κάθε Δευτέρα. Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα του πως είναι ο task scheduler.



Εικόνα 9.1: Αρχική εικόνα του task scheduler

Μπορούμε να βρούμε τον task scheduler και να ελέγξουμε τις προγραμματισμένες εργασίες του, ακολουθώντας την εξής διαδρομή.

- Πατάμε **Start**
- Στη συνέχεια πατάμε στο **Programs** ή **All Programms**
- Μετά πατάμε στο Accessories και εν συνεχεία στο **System Tools**
- Τέλος στο φάκελο του System Tools πατάμε πάνω στον **Task Scheduler**

Εναλλακτικά πατάμε στο Start και γράφουμε κατευθείαν στη μπάρα αναζήτησης τουTask Scheduler.

9.2 Cron Daemon

Οι εργαζόμενοι που εγκαθιστουν προγράμματα αλλά και αυτοί που πρέπει να διατηρήσουν κάποια προγράμματα χρησιμοποιούν τον Cron ώστε να μπορούν να εκτελούν εργασίες περιοδικά, είτε σε συγκεκριμένες ημερομηνίες και διαστήματα. Συνεπώς αυτό το εργαλείο είναι ιδιαίτερα σημαντικό σε περιπτώσεις όπου συνδεόμαστε στο Διαδίκτυο και κατεβάζουμε e-mail σε τακτά χρονικά διαστήματα. Ο Cron Daemon αποτελεί το ισοδύναμο του Task Scheduler αλλά για συστήματα Unix. Δυο πολύ σημαντικά αρχεία είναι τα ακόλουθα:

• **/etc/cron.allow.** Εάν αυτό το αρχείο υπάρχει πρέπει να περιέχει το username μας ώστε να μπορέσουμε να το χρησιμοποιήσουμε.

• **/etc/cron.deny.** Εάν αυτό το cron αρχείο δεν υπάρχει και υπάρχει το /etc/cron.deny τότε για να μπορέσουμε να χρησιμοποιήσουμε τις cron εργασίες δεν θα πρέπει να συμπεριλάβουμε τα αρχεία του /etc/cron.deny.

9.3 Προβολή συμβάντων (Event Viewer)

Στα Windows, το συμβάν είναι ένα σημαντικό περιστατικό στο σύστημα ή σε ένα πρόγραμμα, το οποίο απαιτεί να ειδοποιούνται οι χρήστες ή να προστίθεται μια καταχώρηση σε ένα αρχείο καταγραφής. Η υπηρεσία καταγραφής συμβάντων (Event Log Service) καταγράφει συμβάντα εφαρμογών, ασφαλείας και συστήματος στην προβολή συμβάντων (Event Viewer). Με τα αρχεία καταγραφής συμβάντων της προβολής συμβάντων (Event Viewer), μπορούμε να αποκτήσουμε πληροφορίες σχετικά με τα στοιχεία του υλικού, του λογισμικού και του συστήματος καθώς και να παρακολουθούμε συμβάντα ασφαλείας σε έναν τοπικό ή απομακρυσμένο υπολογιστή. Τα αρχεία καταγραφής συμβάντων μπορούν να μας βοηθήσουν στην αναγνώριση και τη διάγνωση της προέλευσης των τρεχόντων προβλημάτων συστήματος ή να μας βοηθήσουν στην πρόβλεψη πιθανών προβλημάτων συστήματος.

Τύποι αρχείων καταγραφής

Ένας υπολογιστής που βασίζεται στα Windows XP καταγράφει συμβάντα στα ακόλουθα τρία αρχεία καταγραφής:

- Αρχείο καταγραφής εφαρμογής (Application log). Το αρχείο καταγραφής εφαρμογής περιέχει συμβάντα που έχουν καταγραφεί από προγράμματα. Για παράδειγμα, ένα πρόγραμμα βάσης δεδομένων ενδέχεται να καταγράψει ένα σφάλμα αρχείου στο αρχείο καταγραφής εφαρμογής. Τα συμβάντα που καταγράφονται στο αρχείο καταγραφής εφαρμογής καθορίζονται από τους προγραμματιστές του προγράμματος λογισμικού.

- Αρχείο καταγραφής ασφαλείας (Security log). Το αρχείο καταγραφής ασφαλείας καταγράφει συμβάντα όπως έγκυρες και μη έγκυρες προσπάθειες σύνδεσης, καθώς και συμβάντα που σχετίζονται με τη χρήση πόρων, όπως είναι η δημιουργία, το άνοιγμα ή η διαγραφή αρχείων. Για παράδειγμα, όταν είναι ενεργοποιημένος ο έλεγχος σύνδεσης, καταγράφεται ένα συμβάν στο αρχείο καταγραφής συμβάντων, κάθε φορά που ο χρήστης προσπαθεί να συνδεθεί στον υπολογιστή. Για να ενεργοποιήσουμε, να χρησιμοποιήσουμε και να καθορίσουμε τα συμβάντα που καταγράφονται στο αρχείο καταγραφής ασφαλείας, πρέπει πρώτα να έχουμε συνδεθεί ως διαχειριστής ή ως μέλος της ομάδας Administrators.

- Αρχείο καταγραφής συστήματος (System log). Το αρχείο καταγραφής συστήματος περιέχει συμβάντα που έχουν καταγραφεί από τα στοιχεία συστήματος των Windows XP. Για παράδειγμα, όταν αποτύχει η φόρτωση ενός προγράμματος οδήγησης κατά την εκκίνηση, καταγράφεται ένα συμβάν στο αρχείο καταγραφής συστήματος. Τα Windows XP προκαθορίζουν τα συμβάντα που καταγράφονται από τα στοιχεία συστήματος.

Τρόπος προβολής αρχείων καταγραφής συμβάντων

Για να ανοίξουμε την προβολή συμβάντων (Event Viewer), ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Στη συνέχεια κάνουμε κλικ στην επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα προβολή συμβάντων (Event Viewer).

2. Στη δομή κονσόλας, κάνουμε κλικ στο στοιχείο **προβολή συμβάντων (Event Viewer)**.

Τα αρχεία καταγραφής εφαρμογής, ασφαλείας και συστήματος εμφανίζονται στο παράθυρο της προβολής συμβάντων (Event Viewer).

Τρόπος προβολής λεπτομερειών συμβάντων

Για να προβάλουμε τις λεπτομέρειες ενός συμβάντος, ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Συνεχίζουμε κάνοντας κλικ στην επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα Προβολή Συμβάντων (Event Viewer).
2. Στη δομή κονσόλας, αναπτύσουμε το στοιχείο **Προβολή Συμβάντων (Event Viewer)** και στη συνέχεια, κάνουμε κλικ στο αρχείο καταγραφής που περιέχει το συμβάν που θέλουμε να προβάλουμε.
3. Στο παράθυρο λεπτομερειών, κάνουμε διπλό κλικ στο συμβάν που θέλουμε να προβάλουμε. Στη συνέχεια εμφανίζεται το παράθυρο διαλόγου **Ιδιότητες συμβάντος (Event Properties)**, το οποίο περιέχει πληροφορίες κεφαλίδας και μια περιγραφή του συμβάντος. Για να αντιγράψουμε τις λεπτομέρειες του συμβάντος, κάνουμε κλικ στο κουμπί **Αντιγραφή (Copy)**, έπειτα ανοίγουμε ένα νέο έγγραφο με το πρόγραμμα στο οποίο θέλουμε να επικολλήσουμε το συμβάν (για παράδειγμα, στο Microsoft Word) και, στη συνέχεια, κάνουμε κλικ στην εντολή **Επικόλληση (Paste)** από το μενού **Επεξεργασία (Edit)**.

   Για να προβάλουμε την περιγραφή του προηγούμενου ή του επόμενου συμβάντος, κάνουμε κλικ στο επάνω ή στο κάτω βέλος.

Τρόπος ερμηνείας ενός συμβάντος

Κάθε καταχώρηση αρχείου καταγραφής ταξινομείται κατά τύπο και περιέχει πληροφορίες κεφαλίδας και μια περιγραφή του συμβάντος.

Κεφαλίδα συμβάντος

Η κεφαλίδα συμβάντος περιέχει τις ακόλουθες πληροφορίες σχετικά με το συμβάν:

- Ημερομηνία ή ημερομηνία εμφάνισης του συμβάντος.
- Ώρα
  Η ώρα εμφάνισης του συμβάντος.

- Χρήστης
  Το όνομα του χρήστη ο οποίος ήταν συνδεδεμένος κατά την εμφάνιση του συμβάντος.

- Υπολογιστής
  Το όνομα του υπολογιστή όπου εμφανίστηκε το συμβάν.

- Αναγνωριστικό συμβάντος
  Ένας αριθμός συμβάντος που προσδιορίζει τον τύπο του συμβάντος. Το αναγνωριστικό συμβάντος είναι δυνατό να χρησιμοποιηθεί από τους αντιπροσώπους υποστήριξης προϊόντων για να σας βοηθήσει να κατανοήσετε τι προέκυψε στο σύστημα.

- Προέλευση
  Η προέλευση του συμβάντος. Αυτή μπορεί να είναι το όνομα ενός προγράμματος, ένα στοιχείο συστήματος ή ένα μεμονωμένο στοιχείο ενός μεγάλου προγράμματος.

- Τύπος
  Ο τύπος του συμβάντος. Αυτός μπορεί να είναι ένας από τους ακόλουθους πέντε τύπους: Σφάλμα, Προειδοποίηση, Πληροφορία, Επιτυχημένος έλεγχος ή Αποτυχημένος έλεγχος.

- Κατηγορία:
  Η ταξινόμηση του συμβάντος κατά προέλευση συμβάντος. Χρησιμοποιείται κυρίως στο αρχείο καταγραφής ασφαλείας.

Τύποι συμβάντος

Η περιγραφή κάθε συμβάντος που καταγράφεται εξαρτάται από τον τύπο του συμβάντος. Κάθε συμβάν ενός αρχείου καταγραφής ταξινομείται σε έναν από τους ακόλουθους τύπους:

- Πληροφορίες
  Ένα συμβάν περιγράφει την επιτυχημένη λειτουργία μιας εργασίας, όπως μιας εφαρμογής, ενός προγράμματος οδήγησης ή μιας υπηρεσίας. Για παράδειγμα, ένα συμβάν πληροφοριών καταγράφεται όταν φορτωθεί σωστά ένα πρόγραμμα οδήγησης δικτύου.

- Προειδοποίηση
  Ένα συμβάν που δεν είναι απαραίτητα σημαντικό, ωστόσο, ενδέχεται να υποδηλώνει την πιθανή εμφάνιση ενός μελλοντικού προβλήματος. Για

παράδειγμα, ένα μήνυμα προειδοποίησης καταγράφεται όταν αρχίζει να εξαντλείται ο χώρος στο δίσκο.

- Σφάλμα
  Ένα συμβάν που περιγράφει ένα σημαντικό πρόβλημα, όπως την αποτυχία μιας κρίσιμης εργασίας. Τα συμβάντα σφάλματος ενδέχεται να συνεπάγονται απώλεια δεδομένων ή απώλεια λειτουργιών. Για παράδειγμα, ένα συμβάν σφάλματος καταγράφεται όταν μια υπηρεσία αποτυγχάνει να φορτωθεί στη διάρκεια της εκκίνησης.

- Επιτυχημένος έλεγχος (Αρχείο καταγραφής ασφαλείας)
  Ένα συμβάν που περιγράφει την επιτυχημένη ολοκλήρωση ενός ελεγχόμενου συμβάντος ασφαλείας. Για παράδειγμα, ένα συμβάν επιτυχημένου ελέγχου καταγράφεται κατά τη σύνδεση του χρήστη στον υπολογιστή.

- Αποτυχημένος έλεγχος (Αρχείο καταγραφής ασφαλείας)
  Ένα συμβάν που περιγράφει ένα ελεγχόμενο συμβάν ασφαλείας, το οποίο δεν ολοκληρώνεται με επιτυχία. Για παράδειγμα, αποτυχημένος έλεγχος μπορεί να καταγραφεί όταν κάποιος χρήστης δεν μπορέσει να αποκτήσει πρόσβαση σε μια μονάδα δίσκου δικτύου.

Τρόπος εύρεσης συμβάντων σε ένα αρχείο καταγραφής

Η προεπιλεγμένη προβολή των αρχείων καταγραφής συμβάντων είναι μια λίστα όλων των καταχωρήσεων. Εάν θελήσουμε να βρούμε ένα συγκεκριμένο συμβάν ή να προβάλουμε ένα υποσύνολο συμβάντων, μπορούμε να εκτελέσουμε αναζήτηση στο αρχείο καταγραφής ή να εφαρμόσουμε ένα φίλτρο στα δεδομένα καταγραφής.

Τρόπος αναζήτησης ενός συγκεκριμένου συμβάντος σε αρχείο καταγραφής

Για να αναζητήσουμε ένα συγκεκριμένο συμβάν σε ένα αρχείο καταγραφής, ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Στη συνέχεια κάνουμε κλικ στην επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα προβολή συμβάντων (Event Viewer).

2. Στη δομή κονσόλας, αναπτύσουμε το στοιχείο **Προβολή Συμβάντων (Event Viewer)** και στη συνέχεια, κάνουμε κλικ στο αρχείο καταγραφής που περιέχει το συμβάν που θέλουμε να προβάλουμε.
3. Στο μενού **Προβολή (View)**, κάνουμε κλικ στην εντολή **Εύρεση (Find)**.
4. Καθορίζουμε τις επιλογές για το συμβάν που θέλουμε να προβάλουμε στο παράθυρο διαλόγου **Εύρεση (Find)** και στη συνέχεια, κάνουμε κλικ στην επιλογή **Εύρεση επόμενου (Find Next)**.

Το συμβάν που ταιριάζει με τα κριτήρια αναζήτησης που έχουμε καθορίσει εμφανίζεται με επισήμανση στο παράθυρο λεπτομερειών. Στη συνέχεια κάνουμε κλικ στην επιλογή **Εύρεση επόμενου (Find Next)** για να εντοπίσουμε την επόμενη εμφάνιση ενός συμβάντος, όπως καθορίζεται από τα κριτήρια αναζήτησης.

Τρόπος φιλτραρίσματος συμβάντων αρχείου καταγραφής

Για να φιλτράρουμε συμβάντα αρχείων καταγραφής, ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και, στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Στη συνέχεια κάνουμε κλικ στην επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα Προβολή Συμβάντων (Event Viewer).
2. Στη δομή κονσόλας, αναπτύσσουμε το στοιχείο **Προβολή Συμβάντων (Event Viewer)** και στη συνέχεια, κάνουμε κλικ στο αρχείο καταγραφής που περιέχει το συμβάν που θέλουμε να προβάλουμε.
3. Στο μενού **Προβολή (View)**, κάνουμε κλικ στην εντολή **Φίλτρο (Filter)**.
4. Στην περίπτωση που δεν είναι ήδη επιλεγμένη κάνουμε κλικ στην καρτέλα **Φίλτρο (Filter)**.
5. Καθορίζουμε τις επιλογές φίλτρου που θέλουμε και κατόπιν κάνουμε κλικ στο κουμπί **ΟΚ**.

Μόνο τα συμβάντα που ταιριάζουν με τα κριτήρια φίλτρου που έχουμε καθορίσει θα εμφανιστούν στο παράθυρο λεπτομερειών.

Για να επιστρέψουμε στην προβολή όπου εμφανίζονται όλες οι καταχωρήσεις αρχείου καταγραφής, κάνουμε κλικ στην επιλογή **Φίλτρο (Filter)** του μενού **Προβολή (View)** και στη συνέχεια, κάνουμε κλικ στην επιλογή **Επαναφορά προεπιλογών (Restore Defaults)**.

Τρόπος διαχείρισης περιεχομένων αρχείου καταγραφής

Από προεπιλογή, το αρχικό μέγιστο μέγεθος ενός αρχείου καταγραφής είναι ορισμένο στα 512 KB και όταν αυτό το όριο καλυφθεί, τα νέα συμβάντα

αντικαθιστούν τα παλαιότερα, όπως απαιτείται. Ανάλογα με τις απαιτήσεις μας, μπορούμε να αλλάξουμε αυτές τις ρυθμίσεις ή και να καταργήσουμε τα περιεχόμενα ενός αρχείου καταγραφής.

Τρόπος ορισμού του μεγέθους ενός αρχείου καταγραφής και αντικατάσταση επιλογών

Για να καθορίσουμε το μέγεθος ενός αρχείου καταγραφής και να αντικαταστήσουμε τις επιλογές, ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και, στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Στη συνέχεια κάνουμε κλικ στην επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα Προβολή Συμβάντων (Event Viewer).
2. Στη δομή κονσόλας, αναπτύσουμε το στοιχείο **Προβολή Συμβάντων (Event Viewer)** και στη συνέχεια, κάνουμε κλικ με το δεξιό κουμπί του ποντικιού στο αρχείο καταγραφής, του οποίου θέλουμε να ορίσουμε το μέγεθος και να αντικαταστήσουμε τις επιλογές του.
3. Στην περιοχή **Μέγεθος αρχείου καταγραφής (Log size)**, πληκτρολογούμε το μέγεθος που θέλουμε στο πλαίσιο **Μέγιστο μέγεθος αρχείου καταγραφής (Maximum log size)**.
4. Στην περιοχή **Όταν καλυφθεί το μέγιστο μέγεθος του αρχείου καταγραφής (When maximum log size is reached)**, κάνουμε κλικ στην επιλογή αντικατάστασης που θέλουμε.
5. Εάν θέλουμε να καταργήσουμε τα περιεχόμενα του αρχείου καταγραφής, κάνουμε κλικ στην επιλογή **Εκκαθάριση αρχείου καταγραφής (Clear Log)**.
6. Τέλος κάνουμε κλικ στο κουμπί **OK**.

Τρόπος αρχειοθέτησης ενός αρχείου καταγραφής

Εάν θέλουμε να αποθηκεύσουμε τα δεδομένα του αρχείου καταγραφής, μπορούμε να αρχειοθετήσουμε τα αρχεία καταγραφής συμβάντων σε οποιαδήποτε από τις ακόλουθες μορφές:

- Μορφή αρχείου καταγραφής (.evt)
- Μορφή αρχείου κειμένου (.txt)
- Μορφή αρχείου κειμένου οριοθετημένου με κόμματα (.csv)

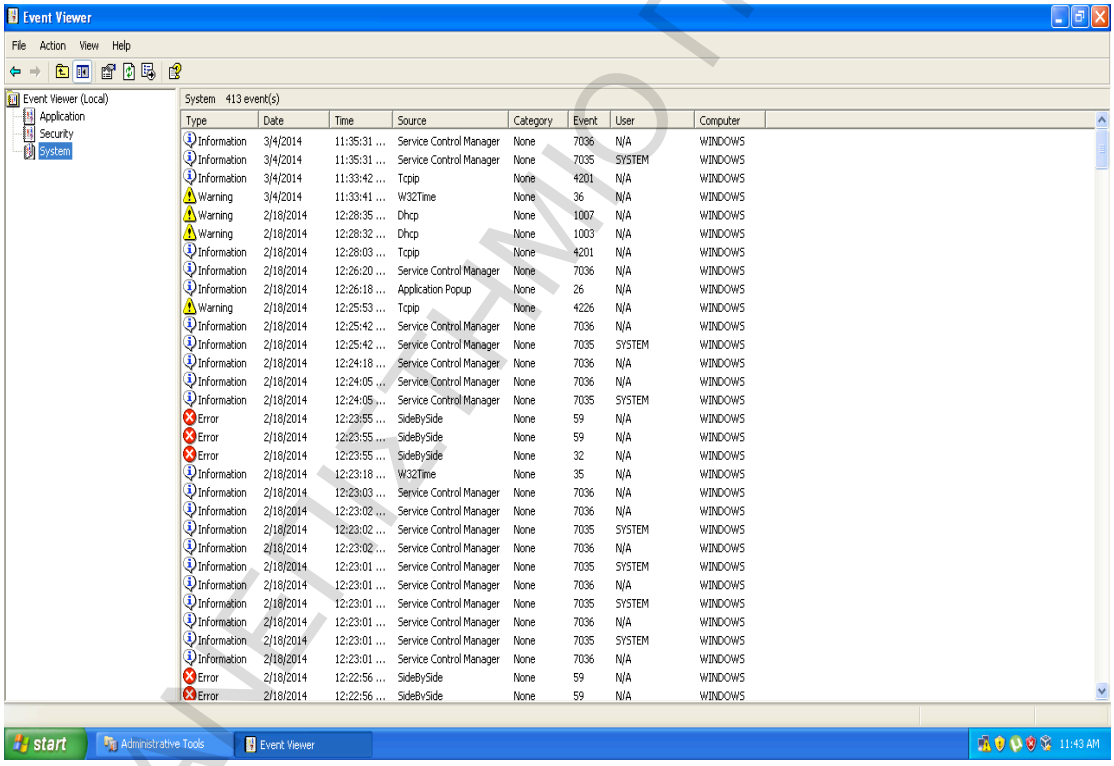
Για να αρχειοθετήσουμε ένα αρχείο καταγραφής, ακολουθούμε τα εξής βήματα:

1. Κάνουμε κλικ στο μενού **Έναρξη (Start)** και, στη συνέχεια, κάνουμε κλικ στην επιλογή **Πίνακας Ελέγχου (Control Panel)**. Στη συνέχεια κάνουμε κλικ στην

επιλογή **Επιδόσεις και Συντήρηση (Performance and Maintenance)**, έπειτα στην επιλογή **Εργαλεία διαχείρισης (Administrative Tools)** και τέλος, κάνουμε διπλό κλικ στην επιλογή **Διαχείριση Υπολογιστή (Computer Management)**. Εναλλακτικά, μπορούμε να ανοίξουμε το MMC που περιέχει το συμπληρωματικό πρόγραμμα Προβολή Συμβάντων (Event Viewer).

2. Στη δομή κονσόλας, αναπτύσουμε το στοιχείο **Προβολή Συμβάντων (Event Viewer)** και στη συνέχεια, κάνουμε κλικ με το δεξιό κουμπί του ποντικιού στο αρχείο καταγραφής στο οποίο θέλουμε να αρχειοθετήσουμε και συνεχίζουμε, κάνοντας κλικ στην επιλογή **Αποθήκευση αρχείου καταγραφής ως (Save Log File As)**.
3. Έπειτα καθορίζουμε το όνομα του αρχείου καθώς και τη θέση στην οποία θέλουμε να αποθηκεύσουμε το αρχείο. Στο πλαίσιο **Αποθήκευση ως τύπου (Save as type)**, κάνουμε κλικ στη μορφή που θέλουμε και στη συνέχεια, κάνουμε κλικ στο κουμπί **Αποθήκευση (Save)**.

Το αρχείο καταγραφής αποθηκεύεται με τη μορφή που έχουμε καθορίσει.



Εικόνα 9.2: Ο Event Viewer προσφέρει πρόσβαση σε όλα τα μηνύματα του συστήματος. Θεωρείται δύσκολο αλλά είναι πιθανόν να μας βοηθήσει να εντοπίσουμε τα ίχνη κάποιου κακόβουλου προγράμματος.

Στην κατηγορία security αναφέρονται όλες οι παρασκηνιακές κινήσει που γίνονται στα Windows και οι οποίες σχετίζονται με την ασφάλεια. Για παράδειγμα εδώ μπορούμε να δούμε τις συνδέσεις που πραγματοποιούνται στο σύστημα (τα login).

Τέλος η κατηγορία Application είναι η πιο σημαντική. Εδώ καλό είναι να ασχολούμαστε με όλα τα Errors και Warnings που εμφανίζονται. Αν και αποτελεί μια βαρετή διαδικασία κάποιες φορές μπορεί να είναι απαραίτητη και να μας γλιτώσει από φασαρίες.

9.4 Task Manager

Ένα άλλο πολύτιμο εργαλείο για την αντιμετώπιση του κακόβουλου λογισμικού είναι και ο Task Manager. Από αυτόν μπορούμε να δούμε όλες τις εργασίες που εκτελούνται στο παρασκήνιο του συστήματος. Για να τον εμφανίσουμε θα πρέπει να πατήσουμε ταυτόχρονα τα πλήκτρα [CTRL + Shift + ESC].

Εκτός αυτού, το Task Manager προβάλλει την πληρότητα του επεξεργαστή και της μνήμης RAM, και επιτρέπει τον υποχρεωτικό τερματισμό («κολλημένων») προγραμμάτων.

Το Taskmanager διαχειρίζεται τη λειτουργία του υπολογιστή. Το πρόγραμμα συστήματος των Windows βοηθά σε περίπτωση κατάρρευσης κάποιου προγράμματος.

Επίσης το Taskmanager χρησιμοποιείται σε περίπτωση που θέλουμε να αποκτήσουμε μια γρήγορη επισκόπηση των λειτουργιών ενός υπολογιστή με λειτουργικό σύστημα Windows που βρίσκονται σε εξέλιξη, αρκεί να πιέσουμε τα πλήκτρα [CTRL+ Shift + ESC]. Τότε το Taskmanager (που στα Αγγλικά σημαίνει «διαχείριση εργασιών) δείχνει τις εφαρμογές και διαδικασίες που βρίσκονται σε λειτουργία, καθώς και την πληρότητα του CPU και της μνήμης RAM, τα χρησιμοποιούμενα δίκτυα και το χρήστη. Εναλλακτικά, μπορούμε να ανοίξουμε το Taskmanager κάνοντας διπλό κλικ στη γραμμή εργασιών.Το Taskmanager προσφέρει γρήγορη επισκόπηση σε περίπτωση προβλημάτων του υπολογιστή.

Μέσω του Taskmanager μπορεί κανείς να τερματίσει αμέσως τα καταγεγραμμένα προγράμματα, όταν π.χ. «κολάνε». Όταν ένα πρόγραμμα δεν αποκρίνεται στις εντολές, συνιστάται αρχικά ο έλεγχός του από το Taskmanager. Η δήλωση κατάστασης «δεν αποκρίνεται» υποδεικνύει π.χ. ότι το πρόγραμμα είναι πολύ φορτωμένο ή έχει καταρρεύσει.

Εκτός αυτού, το Taskmanager προσφέρει γρήγορη βοήθεια σε περίπτωση προβλημάτων απόδοσης. Εάν ο υπολογιστής είναι πολύ αργός, μπορεί κανείς να ελέγξει εάν αυτό οφείλεται στην πληρότητα του CPU, σε ένα πρόγραμμα με αυξημένες υπολογιστικές απαιτήσεις ή σε κακή σύνδεση δικτύου.
Το Taskmanager όπως αναφέραμε μπορεί να αποδειχθεί, επίσης, πολύ χρήσιμο κατά την αναζήτηση κακόβουλου λογισμικού. Όταν π.χ. είναι ενεργός κάποιος ιός, ενδέχεται το αντίστοιχο αρχείο «.exe» να εμφανιστεί στον κατάλογο των ενεργών διαδικασιών. Πριν τη διαγραφή κάποιου ύποπτου αρχείου, καλό είναι να σιγουρευτούμε αφού δεν είναι όλες οι άγνωστες διαδικασίες ιοί. Πολλές φορές πρόκειται για στοιχεία ενεργών προγραμμάτων που λειτουργούν στο παρασκήνιο.

9.5 Ιός της αστυνομίας

Όταν μιλάμε για τον ιό της αστυνομίας στην πραγματικότητα εννοούμε ένα worm. Απλά έχει επικρατήσει για χάρη ευκολίας όλα σχεδόν τα κακόβουλα προγράμματα να αναφέρονται ως ιοί. Πρόκειται λοιπόν για ένα πονηρό worm το οποίο εγκαθίσταται στη θέση C:\Windows και καλείται αυτόμα τα από το registry με το ακόλουθο κλειδί:

HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Εφόσον το worm ενεργοποιείται κάθε φορά που ξεκινά το σύστημα εμφανίζεται το σήμα δίωξης ηλεκτρονικού εγκλήματος μαζί με το ακόλουθο μήνυμα:

Εικόνα 9.3.a:Το worm της αστυνομίας

Προσοχή:

Αυτό το λειτουργικό σύστημα μπλοκάρεται λόγω παραβίασης των νόμων της Ελλάδος! Σημειώθηκαν οι ακόλουθες παραβιάσεις:

Η IP διεύθυνσή σας είναι...

Πρόκειται για ένα μήνυμα το οποίο ποντάρει στην άγνοια και στο φόβο ορισμένων χρηστών. Υπάρχουν αρκετά στοιχεία από τα οποία μπορούμε να καταλάβουμε ότι πρόκειται για απάτη.

1) Τα ελληνικά σε αυτό το μήνυμα είναι σαν να προέρχονται από το google translate. Δεν υπάρχει κανένας άνθρωπος της πληροφορικής που θα έγραφε κείμενο με τόσα λάθη.
2) Αν η δίωξη ηλεκτρονικού εγκλήματος εντόπιζε κάποια παράνομη δραστηριότητα το σίγουρο είναι ότι δεν θα κλείδωνε τον υπολογιστή του εγκληματία αλλά θα προτιμούσε να τον συλλάβει.

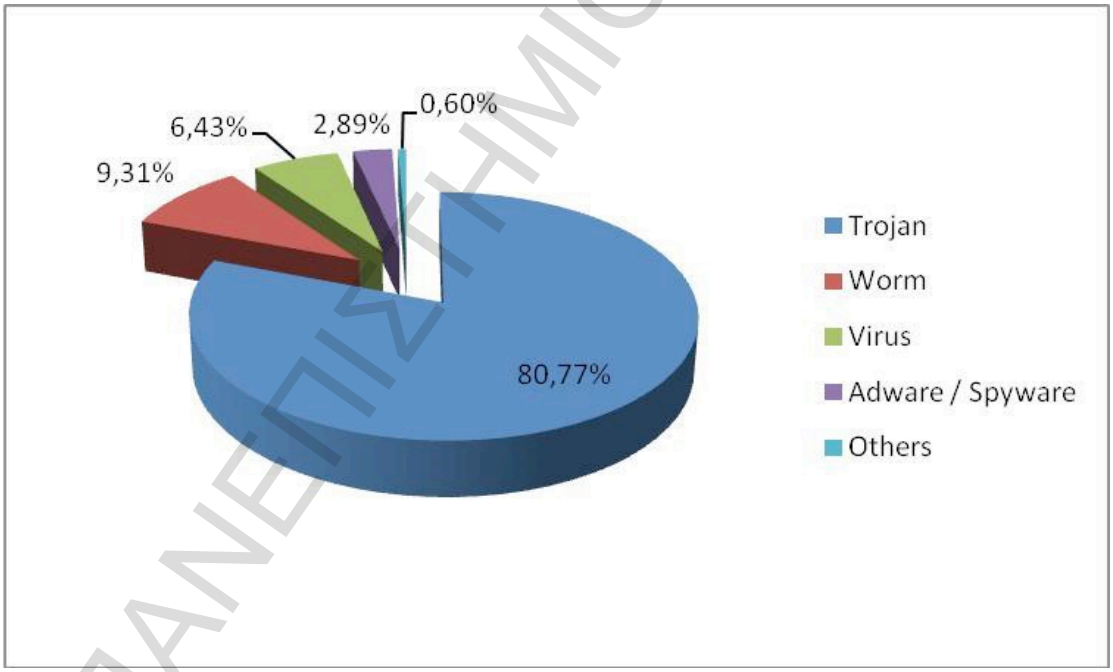
Εικόνα 9.3.b: Το ίδιο worm στα αγγλικά

Όπως καταλαβαίνουμε το worm αποτελεί μετάφραση κάποιου ξένου. Αυτός είναι και ο λόγος που το συγκεκριμένο worm θύμιζε τόσο έντονα μετάφραση από το google translate.

9.6 Στατιστικά των Malware

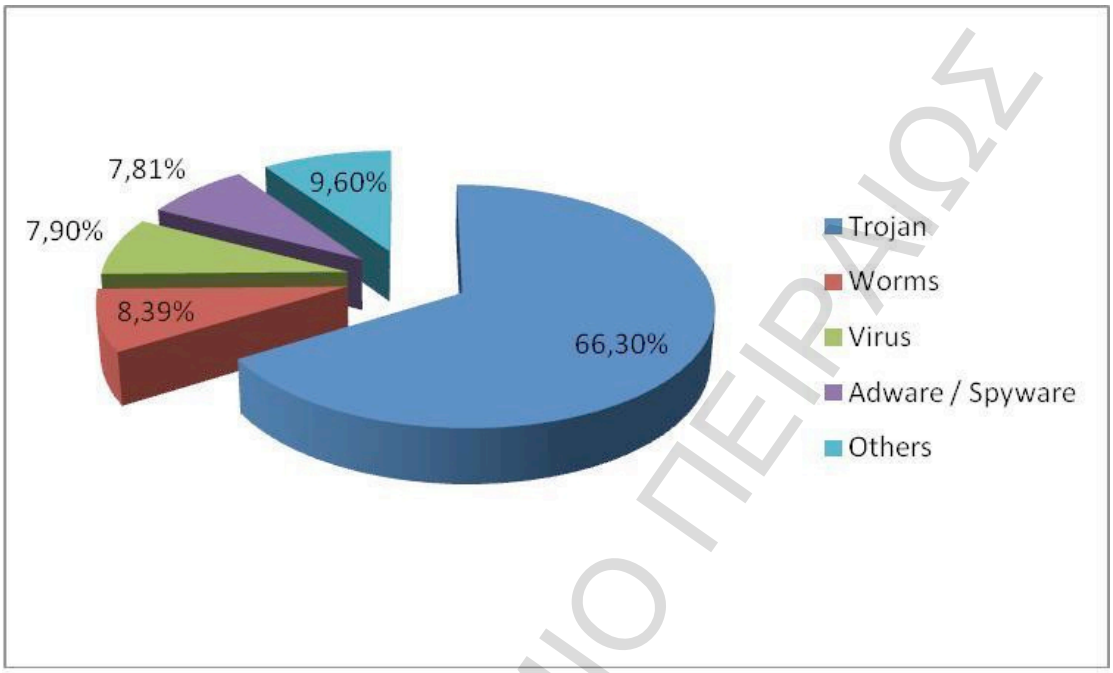
Τα Trojans καταγράφουν ρεκόρ, με ποσοστό 80% μεταξύ όλων των νέων malware, αφού από ότι φαίνεται αποτελούν την πιο δημοφιλή κατηγορία για τους κυβερνο-απατεώνες προκειμένου να εκμαιεύσουν πληροφορίες από τα υποψήφια θύματά τους. Να σημειώσουμε ότι το 2011, αποτελούσαν το 73% όλων των malware.

Στη δεύτερη θέση βρίσκονται τα Worms σε ποσοστό 9,3% ενώ ακολουθούν οι ιοί με ποσοστό 6,43%. Ενδιαφέρον είναι το γεγονός ότι οι δύο αυτές κατηγορίες κακόβουλου λογισμικού έχουν ανταλλάξει τις θέσεις με τα ποσοστά τους σε σύγκριση με την ετήσια έκθεση του 2011, όπου οι ιοί έφτασαν σε ποσοστό το 14,25% ενώ τα worms ήταν τρίτα με 8% του συνόλου του κακόβουλου λογισμικού.



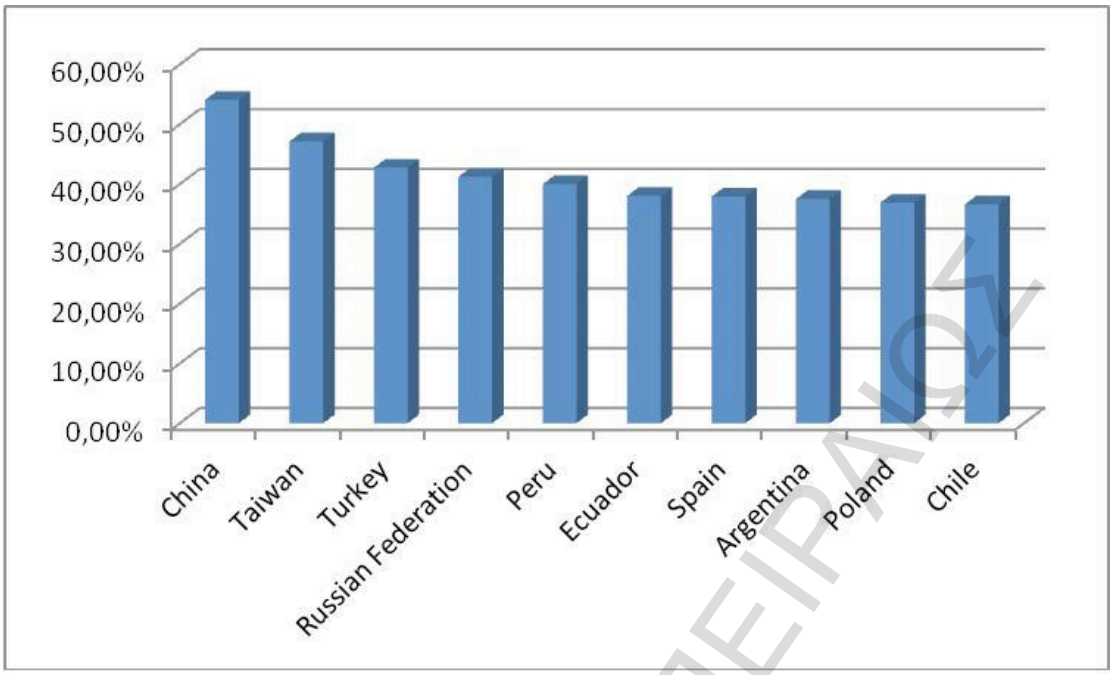
Όσον αφορά στον αριθμό των νέων μολύνσεων που προκλήθηκαν από κάθε κατηγορία malware, η κατάταξη συμπίπτει με εκείνη των νέων δειγμάτων που κυκλοφόρησαν. Τα Trojans, τα worms και οι ιοί κατέλαβαν και πάλι τις πρώτες τρεις θέσεις. Είναι ενδιαφέρον, ότι τα worms προκάλεσαν μόλις το 8% όλων των μολύνσεων παρά το γεγονός ότι αντιπροσωπεύουν πάνω από το 9% του νέου malware. Αυτό είναι αρκετά αξιοσημείωτο, καθώς τα worms προκαλούν συνήθως πολλές περισσότερες μολύνσεις σε υπολογιστές, χάρη στην ικανότητά τους να

προπαγανδίζουν με αυτοματοποιημένο τρόπο. Σε κάθε περίπτωση, τα στοιχεία επιβεβαιώνουν απλώς αυτό που είναι γνωστό: Οι μαζικές επιθέσεις από worms είναι πλέον παρελθόν και έχουν αντικατασταθεί από μία αυξανόμενη χιονοστιβάδα «σιωπηλών» Trojans.



Η Κίνα στην κορυφή των μολυσμένων χωρών

Ο μέσος αριθμός των μολυσμένων υπολογιστών σε όλο τον κόσμο ανέρχεται σε 35,51%, σχεδόν τρεις ποσοστιαίες μονάδες κάτω, σε σύγκριση με το 2011, σύμφωνα με τα στοιχεία που έχει συλλέξει η εταιρεία ασφάλειας Panda Security. Η Κίνα είναι και πάλι επικεφαλής αυτής της κατάταξης (54,25%), ακολουθούμενη από την Ταϊβάν και την Τουρκία. Ο κατάλογος των λιγότερο μολυσμένων χωρών, κυριαρχείται από τις ευρωπαϊκές χώρες, αφού οι εννέα από τις δέκα πρώτες θέσεις καταλαμβάνονται από αυτές. Η Ιαπωνία είναι η μόνη μη ευρωπαϊκή χώρα μεταξύ των πρώτων δέκα χωρών με ποσοστό κάτω από 30% σε υπολογιστές που μολύνθηκαν. Τις τρεις πρώτες θέσεις των λιγότερο μολυσμένων χωρών, καταλαμβάνουν η Σουηδία, η Ελβετία και η Νορβηγία.



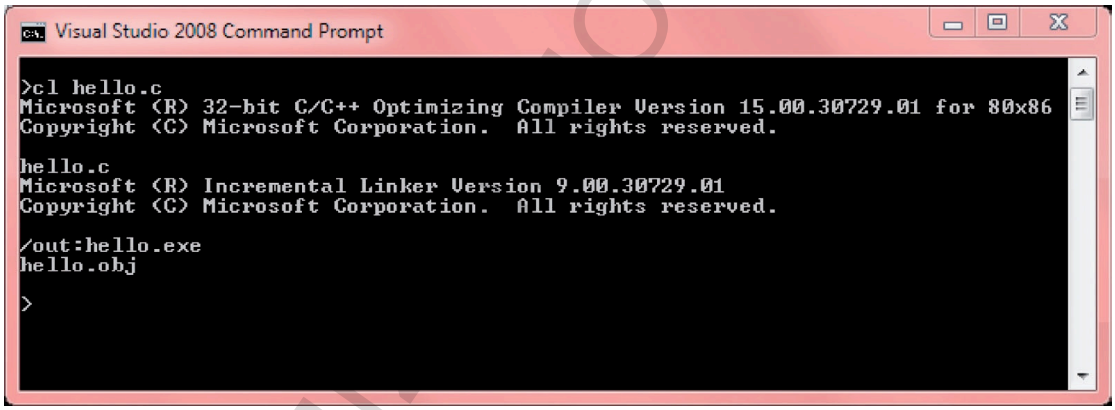
Εικόνα 9.4 Η κατάταξη των χωρών με τις περισσότερες μολύνσεις από malware:

10. Δημιουργία και εκτέλεση malware

Αρχικά θα κατασκευάσουμε ένα απλό πρόγραμμα σε C++, το οποίο δεν θα πραγματοποιεί καμία απολύτως κακόβουλη ενέργεια. Στη συνέχεια θα φτιάξουμε ένα Trojan Horse, το οποίο θα μεταφέρει αυτό το εκτελέσιμο. Όταν εκτελείται το Trojan Horse, θα φυτεύει στο δίσκο του υπολογιστή το απλό προγραμμά μας και θα το εκτελεί. Ξεκινάμε λοιπόν με ένα απλό κι αθώο πρόγραμμα:

```
// hello.c
//
#include <stdio.h>
void main(void)
{
 printf("Hello World!\n");
}
```

Το παραπάνω πρόγραμμα εμφανίζει στην οθόνη το γνωστό «Hello World». Για τη μεταγλώττισή του θα εργαστούμε με τον παραδοσιακό τρόπο, από τη γραμμή εντολών. Εμείς θα χρησιμοποιήσουμε το Microsoft Visual Studio 2008 Professional. Καλούμε λοιπόν το command line μέσω της C++ και παράγουμε το εκτελέσιμο.



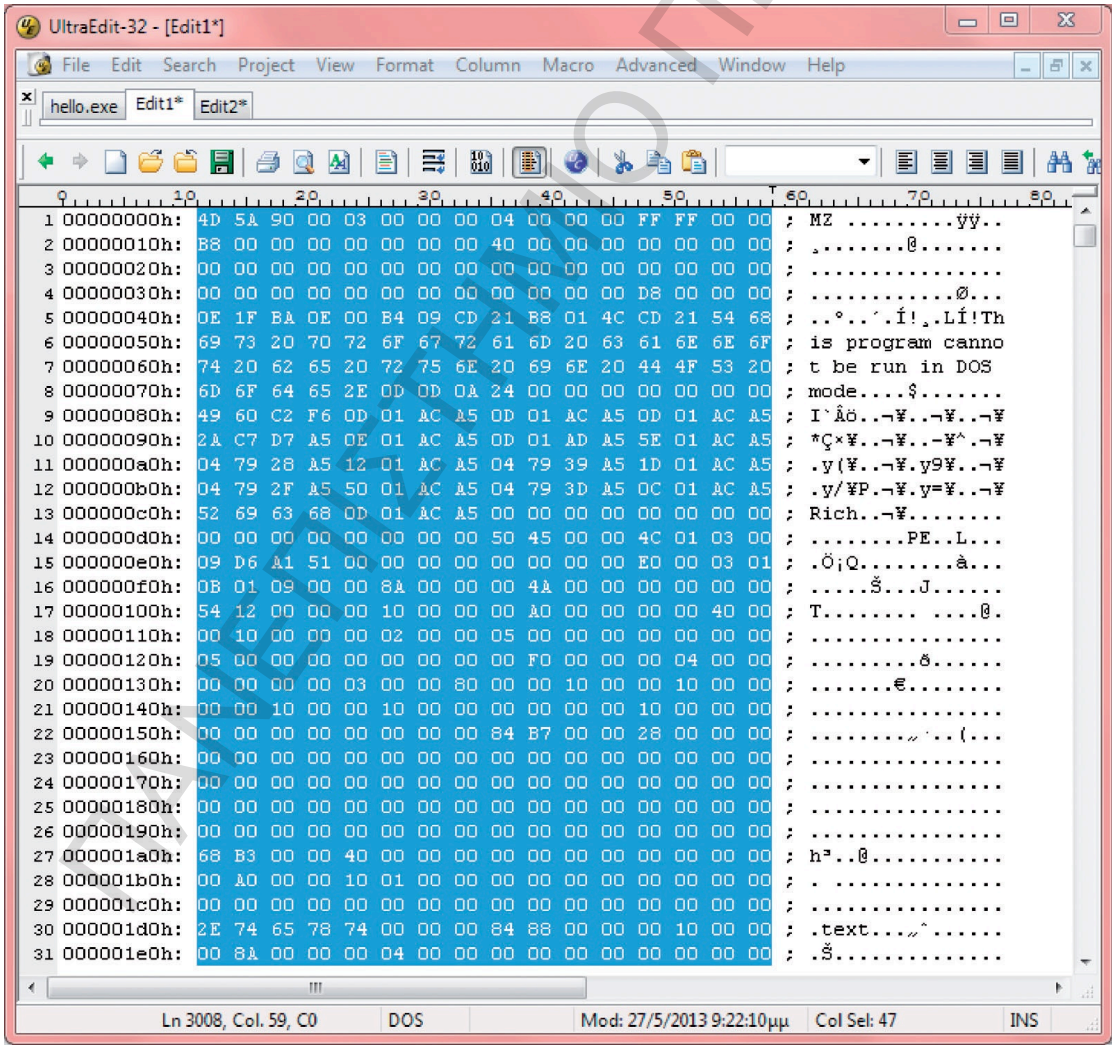
Εικόνα 10.1: Το κάνουμε Compile

Μετά την παραπάνω διαδικασία θα προκύψουν στο δίσκο μας δύο νέα αρχεία (Εικόνα 10.1): Το ένα θα έχει την κατάληξη OBJ (object file) και θα περιέχει τη δυαδική αναπαράσταση των εντολών του πηγαίου κώδικα, ενώ το άλλο θα έχει την κατάληξη .EXE και θα αποτελεί το εκτελέσιμο πρόγραμμα. Το hello.exe αποτελεί το προϊόν της μεταγλώττισης του hello.c. Συγκεκριμένα είναι το hello.obj, επαυξημένο με ορισμένες εντολές του λειτουργικού συστήματος οι οποίες φροντίζουν για την εκτέλεσή του (δέσμευση μνήμης, αίτηση για δημιουργία του process κ.ο.κ.).

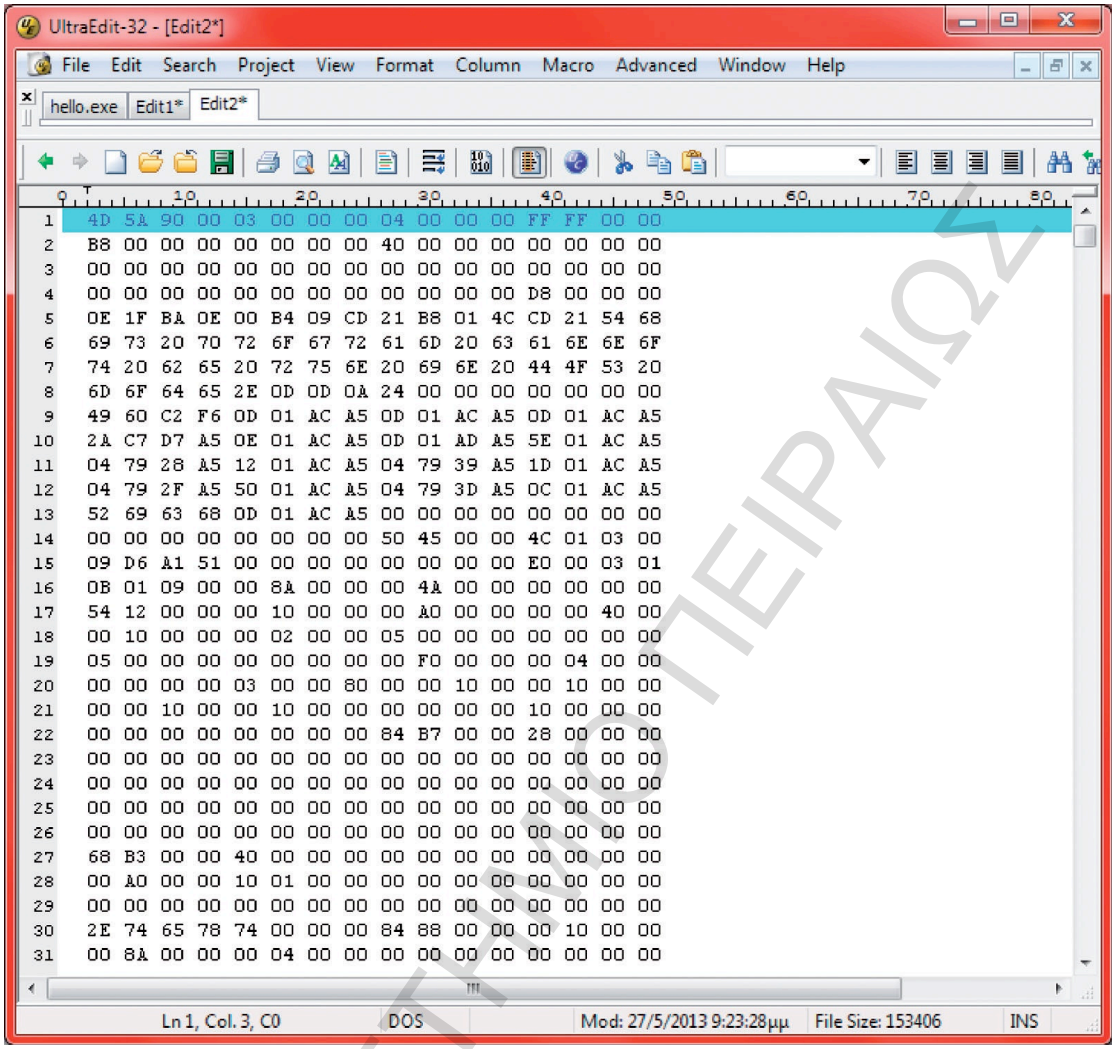
Στη συνέχεια θα πρέπει να αντιγράψουμε τον δυαδικό κώδικα του εκτελέσιμου μέσα σε έναν πίνακα του πηγαίου κώδικα. Για να το πετύχουμε αυτό θα πρέπει να χρησιμοποιήσουμε έναν editor. Θα χρησιμοποιήσουμε λοιπόν το γνωστό Ultraedit (http://www.ultraedit.com).

Αρχικά ανοίγουμε το εκτελέσιμο και κάνουμε Select All [Ctrl+A] κι αντιγράφουμε ολόκληρο τον κώδικα του εκτελέσιμου hello.exe στην δεκαεξαδική του μορφή, με την εντολή «Hex Copy Selected View».

Αυτή η εντολή θα αντιγράψει και τον κώδικα αλλά και την αναπαράσταση του σε απλό κείμενο. Εμείς όμως θέλουμε μόνο τον κώδικα για το λόγο αυτό, ανοίγουμε με τον editor ένα νέο αρχείο και κάνουμε Paste αυτά που μόλις αντιγράψαμε (εικόνα 10.2). Από αυτό το δεύτερο αρχείο αντιγράφουμε μόνο το 16δικό τμήμα και όχι την αναπαράστασή του. Στη συνέχεια ανοίγουμε κι ένα τρίτο αρχείο και κάνουμε πάλι paste (εικόνα 10.3).

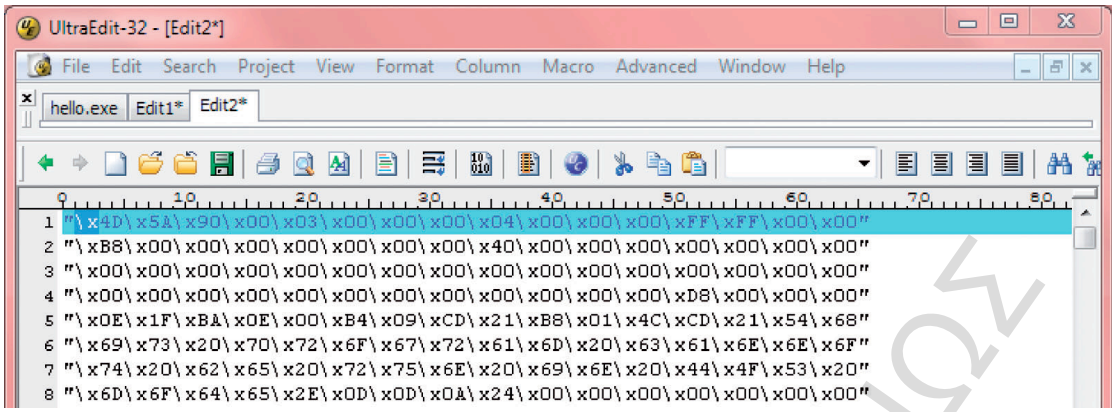


Εικόνα 10.2 *Τώρα αντιγράφουμε μόνο το 16δικό τμήμα και όχι την αναπαράσταση του σε εκτυπώσιμους χαρακτήρες (printable characters).*



Εικόνα 10.3: *Αυτοί είναι οι δεκαεξαδικοί αριθμοί που πρέπει να βάλουμε στον πίνακά μας.*

Στη συνέχεια θα πρέπει να μορφοποιήσουμε αυτούς τους 16δικούς αριθμούς κατά τέτοιον τρόπο, ώστε να μπορούν να εισαχθούν μέσα σε ένα character array. Βασικά, το ζητούμενο είναι να βάλουμε μπροστά από κάθε αριθμό τους χαρακτήρες \x που στη C++ αποτελούν την ένδειξη ότι ακολουθεί αριθμός στο δεκαεξαδικό. Τέλος, κλείνουμε όλα τα δεδομένα μέσα σε εισαγωγικά, για να μπορέσουμε να τα δηλώσουμε ως δεδομένα ενός πίνακα.



Εικόνα 10.4: *Για την εισαγωγή των αριθμών σε ένα string array, χρειάζεται να αλλάξουμε τη μορφή τους.*

Μετά από όλα αυτά, είμαστε πλέον έτοιμοι να γράψουμε τον κώδικα του New δηλαδή του Δούρειου Ίππου μας:

```
////////////////////////////////////
#include <stdio.h>
#include <windows.h>
char executable[] =
{
"\x4D\x5A\x90\x00\x03\x00\x00\x00\x04\x00\x00\x00\xFF\xFF\x00\x00"
"\xB8\x00\x00\x00\x00\x00\x00\x00\x40\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
...
...
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
};
int main(void)
{
  int i, len1 = sizeof(executable);
  char sName[100], sAns[10];

  FILE *ptr ;
 ptr = fopen(".\\TheNewExe.exe", "wb");

for (i=0; i<len1-1; i++)
   fprintf(ptr, "%c",executable[i]);

  fclose(ptr);
 WinExec("TheNewExe.exe", SW_SHOW);
  return 0;
}
```

Το πρώτο πράγμα που κάνει το πρόγραμμα είναι να γράφει τα περιεχόμενα ενός πίνακα στο δίσκο. Αυτός ο πίνακας όμως περιέχει τον κώδικα του εκτελέσιμου προγράμματός μας. Αυτό είναι κάτι που δεν γνωρίζει η C++. Τα περιεχόμενα του πίνακα γράφονται σε ένα αρχείο που ονομάζεται «TheNewExe.exe» και το οποίο είναι κατάλληλο για τη φιλοξενία binary data. Αμέσως μετά, το πρόγραμμά μας ενεργοποιεί το εκτελέσιμο που μόλις δημιούργησε, με τη βοήθεια της συνάρτησης WinExec(). Σε αυτήν τη συνάρτηση, ιδιαίτερη σημασία παρουσιάζει η παράμετρος SW_SHOW η οποία σηματοδοτεί την εκτέλεση του προγράμματος σε ένα ορατό για τον χρήστη παράθυρο. Υπάρχει βέβαια και η SW_HIDE. Η χρησιμότητα αυτής της συνάρτησης είναι σε περίπτωση που θα θέλαμε πραγματικά να κατασκευάσουμε έναν ύπουλο Δούρειο Ίππο. Σε μια τέτοια περίπτωση δεν θα θέλαμε να γίνεται αντιληπτή η εκτέλεση του κακόβουλου προγράμματος. Στη συνέχεια μεταγλωττίζουμε και τρέχουμε το New.

Μόλις εκτελεστεί το πρόγραμμα θα δούμε στην οθόνη μας αυτό που θα βλέπαμε αν εκτελούσαμε το hello.exe. Την ίδια στιγμή στο δίσκο μας θα έχει δημιουργηθεί ένα νέο εκτελέσιμο, με το όνομα TheNewExe.exe.

11. Επίλογος

Η ασφάλεια πληροφοριών δεν είναι ένας στατικός τομέας, και η απειλή από τα malware εξελίσσεται συνεχώς. Κατά συνέπεια η συνεχής ενημέρωση και γνώση κρίνεται αναγκαία για την εμπόδιση της εξάπλωσης του κακόβουλου λογισμικού. Η ανάλυση επιτρέπει στον αναλυτή να καταλάβει τη συμπεριφορά του κακόβουλου προγράμματος από μια διαφορετική σκοπιά. Υπάρχουν διάφορα εργαλεία και προσεγγίσεις για την ανάλυση κακόβουλου λογισμικού. Σε αυτή την εργασία προτείναμε ορισμένα εργαλεία και μια μεθοδολογία που μπορούμε να ακολουθήσουμε κατά την ανάλυση κακόβουλου λογισμικού. Δυστυχώς λόγο της φύσης του κακόβουλου λογισμικού αλλά και του ότι εξελίσσεται διαρκώς δεν μπορούμε να πούμε ότι υπάρχει μια συγκεκριμένη μεθοδολογία που μπορούμε να ακολουθήσουμε κατά τη διαδικασία της ανάλυσης. Η ανάλυση κακόβουλου λογισμικού αποτελεί ένα ιδιαίτερα δυσκολο τομέα στην ασφάλεια υπολογιστών, ο οποίος απαιτεί ιδιαίτερες γνώσεις αλλά και κόπο για να πραγματοποιηθεί σωστά.

Ένας ιδιαίτερα σημαντικός παράγοντας κατά την ανάλυση κακόβουλου λογισμικού είναι η συγκράτηση-διατήρηση του malware εντός του περιβάλλοντος ανάλυσης. Ο αναλυτής κακόβουλου λογισμικού συνήθως χρησιμοποιεί ένα ξεχωριστό σύστημα (είτε εικονικό είτε φυσικό) εντός ενός διαχωρισμένου δικτύου για να κρατήσει το malware απομονωμένο και να προστατεύσει το σύστημα. Επίσης το σύστημα που χρησιμοποιείται για την ανάλυση θα πρέπει να μην είναι ιδιαίτερα σύνθετο.

Υπάρχει ένα συνεχές παιχνίδι μεταξύ των δημιουργών και των αναλυτών κακόβουλου λογισμικού. Αυτό έχει οδηγήσει στα πιο σύνθετα κακόβουλα προγράμματα και συνεπώς και στις πιο προηγμένες τεχνικές ανίχνευσης και ανάλυσης, με συνέπεια ένα συνεχές κυνηγητό μεταξύ δημιουργών και αναλυτών κακόβουλου λογισμικού.

Τέλος η ανάλυση θα πρέπει να είναι λεπτομερής, καλά τεκμηριωμένη και πάντα μέσα σε λογικά χρονικά πλαίσια.

Βιβλιογραφία

1. http://en.wikipedia.org/wiki/Malware
2. T.C Sottek 19th June 2012 **US and Israel developed Flame malware to attack Iranian nuclear program** http://www.theverge.com/2012/6/19/3098080/us-israel-flame-malware-iran
3. Mike Rothman. Tuesday 12th June 2012 **Malware Analysis Quant** [Final Paper] https://securosis.com/blog/malware-analysis-quant-final-paper
4. 7th September 2010 Τι είναι ο ιός, trojan, worm, spyware, adware, malware; http://coolweb.gr/virus-trojan-worm-spyware-adware-malware/
5. http://el.wikipedia.org/wiki/Δούρειος_Ίππος
6. Practical Malware Analysis, Kris Kendall, Mandiant Security Company, US
7. 6th March 2012 **THOR:Another P2P Botnet** http://dart-ngo.gr/news/67-general/594-thor--another-p2p-botnet
8. http://en.wikipedia.org/wiki/Botnet
9. Matthew Humphries 15th August 2008 **Shadow botnet creators caught** http://www.geek.com/news/shadow-botnet-creators-caught-577487/
10. Lenny Zeltser 11th October 2010 **SANS: Digital Forensics and Incident Response** http://digital-forensics.sans.org/blog/2010/10/11/3-phases-malware-analysis-behavioral-code-memory-forensics/
11. http://en.wikipedia.org/wiki/Honeypot_(computing)
12. Michael Sikorski and Andrew Honig **Practical Malware Analysis The Hands-On Guide to Dissecting Malicious Software**
13. Chris Sanders 6th July 2011 **Bulding a Malware Analysis Lab** http://www.windowsecurity.com/articles-tutorials/viruses_trojans_malware/Building-Malware-Analysis-Lab.html
14. http://www.winmd5.com
15. http://technet.microsoft.com/en-us/sysinternals/bb897439
16. http://www.mcafee.com/us/downloads/free-tools/bintext.aspx
17. http://www.softpedia.com/dyn-postdownload.php?p=4102&t=5&i=1
18. http://www.ntcore.com/pedetective.php
19. http://www.dependencywalker.com
20. http://thepiratebay.se/torrent/5235650/IDA_Pro_Advanced_v5.5.0.925t_and_Hex-Rays_v1.1.0.090909
21. Open Malware http://oc.gtisc.gatech.edu:8080/search.cgi?search=trojan
22. https://www.virustotal.com
23. Ultimate Packer for eXecutable http://upx.sourceforge.net
24. http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx
25. http://freewarefiles.com/downloads_counter.php?programid=46335
26. http://sourceforge.net/projects/regshot/
27. http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx
28. http://www.wireshark.org

29. http://practicalmalwareanalysis.com/fakenet/
30. http://www.ollydbg.de/download.htm
31. Θεμιστοκλής Γλαβέλης **ΓΡΗΓΟΡΟΤΕΡΗ ΕΚΚΙΝΗΣΗ ΤΩΝ WINDOWS** http://pc-news.gr/home/426----windows.html
32. http://en.wikipedia.org/wiki/Process_(computing)
33. http://en.wikipedia.org/wiki/Child_process
34. http://en.wikipedia.org/wiki/Child_process
35. http://en.wikipedia.org/wiki/Windows_Registry#Keys_and_values
36. http://regshot.software.informer.com
37. http://en.wikipedia.org/wiki/Process_Monitor
38. http://en.wikipedia.org/wiki/Wireshark
39. http://en.wikipedia.org/wiki/Tcpdump
40. http://en.wikipedia.org/wiki/OllyDbg
41. http://en.wikipedia.org/wiki/Sandbox_(computer_security)
42. http://en.wikipedia.org/wiki/Sandbox_(software_development)
43. 2010-2014, Cuckoo Sandbox. http://docs.cuckoosandbox.org/en/latest/introduction/what/
44. http://www.computerhope.com/jargon/t/tasksced.htm
45. http://en.wikipedia.org/wiki/Cron
46. http://www.ultraedit.com