ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ ΤΕΧΝΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΤΕΥΘΥΝΣΗ: ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Πτυχιακή Εργασία

Ανάλυση Κακόβουλου Λογισμικού (Malware Analysis)

Μπασιακούλης Γ. Ευάγγελος

Επιβλέποντες: **Χρήστος Ξενάκης**

Χριστόφορος Νταντογιάν

ΠΕΙΡΑΙΑΣ

ΟΚΤΩΒΡΙΟΣ 2012

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ

Στην παρούσα εργασία θα γίνει μια απόπειρα εξέτασης και ανάλυσης ενός κακόβουλου λογισμικού. Το είδος του κακόβουλου λογισμικού που θα εξεταστεί είναι το worm. Έτσι λοιπόν έχοντας μηδενική γνώση για το worm θα προσπαθήσουμε να κατανοήσουμε την λειτουργία του. Τα βασικά σημεία που θα πρέπει να εξεταστούν ώστε να μπορέσουμε να πούμε ότι έχουμε "αποκωδικοποιήσει" την συμπεριφορά του είναι ο τρόπος που εκτελείται, ο σκοπός του, και φυσικά εφόσον πρόκειται για worm ο τρόπος που μεταδίδεται μέσω του διαδικτύου.

Για την επίτευξη της πλήρης ανάλυσης ενός κακόβουλου λογισμικού απαιτούνται δύο ήδη ανάλυσης. Η *δυναμική*, κατά την οποία το κακόβουλο λογισμικό εκτελείται σε ελεγχόμενο περιβάλλον και παρατηρείται – καταγράφεται η δράση και τα αποτελέσματα της εκτέλεσης σε πραγματικό χρόνο, και η *στατική* κατά την οποία συλλέγονται διάφορα στοιχεία για το κακόβουλο λογισμικό χωρίς αυτό να εκτελείται (σε τι compiler έγινε compile, τι packer χρησιμοποιήθηκε κ.α) και εξετάζεται ο κώδικας assembly του εκτελέσιμου.

Έτσι λοιπόν ο αναλυτής έχοντας κατά νου την πορεία ανάλυσης την οποία θα ακολουθήσει και χρησιμοποιώντας τα κατάλληλα εργαλεία αρχίζει και αναλύει το κακόβουλο λογισμικό. Το ποιο δύσκολο και ταυτόχρονα το ποίο σημαντικό στάδιο της ανάλυσης είναι αυτό της εξέτασης του κώδικα assembly του εκτελέσιμου. Ένας έμπειρος αναλυτής μπορεί να κατανοήσει πλήρως την λειτουργία ενός προγράμματος εξετάζοντας τον assembly κώδικα του. Παρόλα αυτά πολύ χρήσιμα δεδομένα αντλούνται και από την δυναμική ανάλυση γεγονός που μπορεί να δώσει στον αναλυτή μια ιδέα για την λειτουργία του κακόβουλου λογισμικού, και να λειτουργήσει σαν συμπλήρωμα για την κατανόηση των δυσνόητων κομματιών του assembly κώδικα του εκτελέσιμου. Στα κεφάλαια που ακολουθούν παρουσιάζεται η δυναμική ανάλυση και περιγράφονται τα εργαλεία τα οποία χρησιμοποιήθηκαν.

ΚΕΦΑΛΑΙΟ 1
ΚΑΚΟΒΟΥΛΟ ΛΟΓΙΣΜΙΚΟ

Με τον όρο κακόβουλο λογισμικό εννοούνται προγράμματα (κώδικας) που αποσκοπούν σε επιθέσεις κατά της Εμπιστευτικότητας, της Ακεραιότητας και της Διαθεσιμότητας των συστημάτων. Για την εγκατάσταση (μόλυνση) ενός κακόβουλου λογισμικού σε έναν ηλεκτρονικό υπολογιστή, συνήθως απαιτείται η ανθρώπινη συμμέτοχη, *άμεση* (π.χ. ανταλλαγή αρχείων, άνοιγμα συνημμένων ή προεπισκόπηση μηνυμάτων αλληλογραφίας αμφιβόλου προέλευσης) ή *έμμεση* (ανεπαρκής προστασία του υπολογιστή, μη λήψη ενημερωμένων εκδόσεων – updates του λογισμικού ασφαλείας και των προγραμμάτων). Το τμήμα του κώδικα που είναι υπεύθυνο για τις παρενέργειες του λογισμικού ονομάζεται φορτίο (payload).

Εκτός από τις παρενέργειες το κακόβουλο λογισμικό περιλαμβάνει επιπλέον κώδικα με σκοπό την Αναπαραγωγή του και την Μετάδοση του ενώ ανεξάρτητα από το είδος του προβαίνει στις εξής ενέργειες:

- Να εγκατασταθεί στην κατάλληλη περιοχή, ώστε το φορτίο του να εκτελείται μία φορά, συχνά ή πάντα. Η πλέον συνήθης τακτική είναι η δημιουργία μιας εγγραφής στο μητρώο του συστήματος.
- Να εγκατασταθεί σε κατάλληλη περιοχή ώστε η εκτέλεση του να μην είναι ανιχνεύσιμη,
- Να εγκατασταθεί σε κατάλληλη περιοχή ώστε η αφαίρεση του να είναι δύσκολη.

Οι πιο γνωστοί τύποι κακόβουλου λογισμικού είναι:

Ιός (virus): Κακόβουλο λογισμικό το οποίο αφού μολύνει έναν Η/Υ έχει την ικανότητα να αναπαράγεται και να μολύνει άλλα προγράμματα στον Η/Υ ξενιστή. Η μετάδοση του σε άλλους Η/Υ μπορεί να γίνεται αυτόματα (να έχει δηλαδή τα χαρακτηριστικά ενός Σκουληκιού – Worm) ή να απαιτεί ανθρώπινη παρέμβαση (π.χ. αντιγραφή ενός αρχείου σε USB flash disk και άνοιγμα του αρχείου σε κάποιον Η/Υ).
Σκουλήκι (Worm): Κακόβουλο λογισμικό το οποίο, αφού μολύνει έναν Η/Υ, έχει την ικανότητα να μεταδίδεται αυτόματα, κάνοντας χρήση της υπάρχουσας δικτυακής

υποδομής (π.χ. Τοπικά Δίκτυα - Δίκτυα WAN) ή των υπηρεσιών του Internet (IRC chat, e-mail, newsgroups, κ.α.).

Δούρειοι Ίπποι (Trojan Horses): Κακόβουλο λογισμικό στο οποίο είναι εγγενές το στοιχείο της παραπλάνησης, καθώς συνήθως μεταμφιέζεται σε μια (καθ' όλα) χρήσιμη εφαρμογή, η οποία όμως περιέχει κακόβουλο κώδικα. Στην πιο κλασσική των περιπτώσεων, ένα Trojan δημιουργεί μια *κερκόπορτα* (backdoor) στο σύστημα, στην οποία ο επιτιθέμενος θα μπορέσει αργότερα να συνδεθεί ώστε να διαχειριστεί εξ' αποστάσεως το σύστημα. Τις περισσότερες φορές τα Trojans δεν έχουν μολυσματικό χαρακτήρα, δηλαδή δεν αναπαράγονται και για αυτό το λόγο δεν χαρακτηρίζονται επισήμως ως ιοί.

Spyware – Adware: Κακόβουλο λογισμικό με χαρακτηριστικά που πλησιάζουν στις λειτουργίες ενός Δούρειου Ίππου (κυρίως ως προς τον τρόπο μόλυνσης), με σκοπό την παρακολούθηση και υποκλοπή ευαίσθητων δεδομένων (spyware), ή την αποστολή ανεπιθύμητων διαφημιστικών μηνυμάτων (adware). Αναφέρονται ως μέλη της ίδιας κατηγορίας, καθώς συνήθως συνεργάζονται για να πετύχουν τον σκοπό τους (π.χ. παρακολούθηση της αγοραστικής συμπεριφοράς κατά την περιήγηση στο Web και στη συνέχεια αποστολή και εμφάνιση διαφημιστικών μηνυμάτων).

Rootkits: Όπως φαίνεται από την ονομασία τους, ένα rootkit είναι κακόβουλο λογισμικό το οποίο λειτουργεί σε πολύ χαμηλό επίπεδο στο λειτουργικό σύστημα , και συνήθως ενσωματώνει *λειτουργίες απόκρυψης* - stealth ώστε να παρακάμπτει τους μηχανισμούς πρόληψης και ανίχνευσης, όπως firewalls και antivirus. Ένα λογισμικό rootkit μπορεί να ανήκει σε οποιαδήποτε από τις ως άνω κατηγορίες, ωστόσο συνήθως ανοίγει κερκόπορτες (backdoors) που θα επιτρέψουν τη μετέπειτα απομακρυσμένη διαχείριση του ξενιστή από κάποιον τρίτο.

Bots – Zombies: Κακόβουλο λογισμικό που προσβάλλει Η/Υ καθιστώντας τους μέλη ενός δικτύου Η/Υ (botnet) που ελέγχεται εξ αποστάσεως από τρίτους, με σκοπό την πραγματοποίηση *Κατανεμημένων Επιθέσεων Άρνησης Εξυπηρέτησης* (DDOS attacks), δηλαδή επιθέσεων κατά τις οποίες ένας μεγάλος αριθμός μολυσμένων υπολογιστών προσπαθεί να συνδεθεί στον Η/Υ στόχο μέσω δικτύου. Ένας Η/Υ που έχει μολυνθεί

από ένα bot συχνά αναφέρεται ως «*zombie*». Οι Η/Υ – zombies μπορεί να χρησιμοποιηθούν για επιθέσεις DOS σε εξυπηρετητές Web, για την αποστολή μηνυμάτων spam, για την πραγματοποίηση επιθέσεων παραπλάνησης (phising) κ.λ.π.

ΚΕΦΑΛΑΙΟ 2
MALWARE FORENSICS

Malware Forensics, είναι η διαδικασία διερεύνησης και ανάλυσης κακόβουλου κώδικα, για να αποκαλύψει την λειτουργία και τους σκοπούς του, καθώς επίσης και να προσδιορίσει τον τρόπο διείσδυσης του κακόβουλου λογισμικού στο σύστημα. Τα περισσότερα από τα κακόβουλα προγράμματα μπλοκάρονται από τα λογισμικά antivirus, από τα εργαλεία αφαίρεσης spyware, καθώς επίσης και από άλλα παρόμοια εργαλεία. Όμως υπάρχουν και νέα λογισμικά malware που τα προγράμματα antivirus αδυνατούν να ανιχνεύσουν. Ο λόγος που συμβαίνει αυτό είναι γιατί αυτά τα προγράμματα βασίζονται σε μια signature-based μέθοδο ανίχνευσης. Αυτό σημαίνει ότι το antivirus λογισμικό συγκρίνει το περιεχόμενο του ύποπτου κώδικα με τις αποθηκευμένες υπογραφές, όπου η κάθε υπογραφή αντιπροσωπεύει ένα δείγμα κώδικα ή ένα μοναδικό χαρακτηριστικό, που έχει εξάγει από το αρχικό και γνήσιο κακόβουλο λογισμικό. Ωστόσο υπάρχουν οι τεχνικές του *πολυμορφισμού* και του *μεταμορφισμού*, που έχουν ως στόχο να εμποδίσουν τα signature-based προγράμματα αναγνώρισης, χρησιμοποιώντας τυχαία κωδικοποίηση ή κρυπτογράφηση του κώδικα προγράμματος με τέτοιο τρόπο που να διατηρεί την αρχική λειτουργία του.

Μόλις ανιχνευτεί η ύπαρξη ενός κακόβουλου προγράμματος, οι ερευνητές malware πρόκειται να ξεκινήσουν την ανάλυση και την ανατομία του. Η λειτουργία forensics επιτυγχάνεται με την αντιστροφή (reversing) του πηγαίου κώδικα του κακόβουλου λογισμικού και με την επανατοποθέτηση του στο αρχείο πληροφοριών, καθώς και με τα εργαλεία παρακολούθησης δικτύου που μπορεί να παρέχει. Η αντιστροφή (reversing) του πηγαίου κώδικα του malware, είναι η πιο ισχυρή μέθοδος. Ένας αναλυτής malware μπορεί να αποκαλύψει όλες τις λεπτομέρειες σχετικά με τα κακόβουλο λογισμικό, και αυτό έχει σαν αποτέλεσμα, οι συντάκτες κακόβουλων λογισμικών να προσπαθούν να εμποδίσουν αυτή την διαδικασία με τεχνικές αντί-αντιστροφής (anti-reversing). Πρόκειται για τεχνικές που καλύπτουν τον κώδικα, έτσι ώστε να εμποδίζουν την διαδικασία ανάλυσης, αλλά στην ουσία ποτέ δεν την εμποδίζουν πραγματικά.

Γενικά η ανάλυση ενός κακόβουλου λογισμικού μπορεί να χωριστεί σε 2 μεγάλες κατηγορίες. Την δυναμική και την Στατική ανάλυση.

2.1 Δυναμική Ανάλυση

Μια *Δυναμική ανάλυση ή ανάλυση Συμπεριφοράς* περιλαμβάνει την εκτέλεση του κακόβουλου λογισμικού και την παρακολούθηση της συμπεριφοράς του, την αλληλεπίδραση του με το σύστημα, καθώς και τις επιπτώσεις που θα έχει στο σύστημα που το φιλοξενεί. Διάφορα εργαλεία παρακολούθησης χρησιμοποιούνται για να συλλάβουν τις δραστηριότητες του κακόβουλου λογισμικού καθώς και τις αντιδράσεις του. Οι δραστηριότητες αυτές περιλαμβάνουν την προσπάθεια που κάνει για να επικοινωνεί με άλλα μηχανήματα, προσθέτοντας registry keys για να ξεκινήσει αυτόματα το πρόγραμμα όταν το λειτουργικό σύστημα ξεκινά, προσθέτοντας αρχεία σε καταλόγους του συστήματος, καθώς επίσης και το κατέβασμα αρχείων από το διαδίκτυο που με το άνοιγμα θα μολύνουν κι θα εξαπλωθούν σε άλλα αρχεία του συστήματος.

Το πρώτο βήμα σε αυτή την δυναμική ανάλυση είναι να παρθεί ένα στιγμιότυπο (snapshot) του συστήματος που μπορεί να χρησιμοποιηθεί για την σύγκριση μεταξύ της κατάστασης του συστήματος, δηλαδή πριν και μετά την λειτουργία του κακόβουλου λογισμικού. Αυτό βοηθά ώστε να εντοπιστούν τα αρχεία που έχουν προστεθεί ή τροποποιηθεί στο σύστημα. Η κατανόηση στο τι αλλαγές συνέβησαν στο σύστημα μετά την εκτέλεση, θα βοηθήσει στην ανάλυση και την απομάκρυνση του κακόβουλου λογισμικού. Στο περιβάλλον των Windows, τα *host integrity monitors* και τα *installation monitors*, παρέχουν την απαιτούμενη βοήθεια. Τα host integrity ή τα file integrity monitoring tools δημιουργούν ένα στιγμιότυπο (snapshot) του συστήματος, στο οποίο μεταγενέστερες αλλαγές σε αντικείμενα που ανήκουν στο σύστημα, συλλαμβάνονται και συγκρίνονται με το αρχικό στιγμιότυπο. Για τα windows, τα εργαλεία αυτά συνήθως παρακολουθούν τις αλλαγές που έγιναν στο σύστημα αρχείων, στην registry και επίσης στα αρχεία ρυθμίσεων του συστήματος. Δημοφιλή εργαλεία περιλαμβάνουν Winalysis, WinPooch, FileMon, RegMon and RegShot.

Σε αντίθεση με τα host integrity systems, τα installation monitoring tools, παρακολουθούν όλες τις αλλαγές που έγιναν από την εκτέλεση ή την εγκατάσταση του προγράμματος στόχου. Αυτό σημαίνει πως δεν παρακολουθεί όλες τις αλλαγές που συνέβησαν στο σύστημα, αλλά μόνο τις αλλαγές κατά την διάρκεια της εγκατάστασης και την εκκίνησης. . Για τα windows, τα εργαλεία αυτά συνήθως παρακολουθούν τις αλλαγές που έγιναν στο σύστημα αρχείων, στην registry και επίσης στα αρχεία ρυθμίσεων του συστήματος. Δημοφιλή εργαλεία περιλαμβάνουν Incrtl58, InstallSpy9, and SysAnalyzer.

2.2 Στατική Ανάλυση

Η *Στατική Ανάλυση* είναι η διαδικασία ανάλυσης εκτελέσιμου δυαδικού κώδικα, χωρίς στην πραγματικότητα να εκτελείται το αρχείο. Η Στατική ανάλυση έχει το πλεονέκτημα ότι μπορεί να αποκαλύψει, πώς ένα πρόγραμμα θα συμπεριφερθεί κάτω από ασυνήθιστες συνθήκες, αυτό συμβαίνει γιατί μπορούμε να εξετάσουμε κάποια μέρη ενός προγράμματος που κανονικά δεν εκτελούνται. Το κακόβουλο λογισμικό μπορεί να αρχίσει την εκτέλεσή του μετά από κάποιο χρονικό διάστημα ή όταν ένα συμβεί κάποιο ειδικό γεγονός. Θα μπορούσε να ξεκινήσει δηλαδή όταν ο χρήστης περιηγείται σε online κατάστημα ή κάνει κάποια online τραπεζική συναλλαγή. Είναι σημαντικό να βρεθεί πως το κακόβουλο λογισμικό μπορεί να ξεφύγει από τον εντοπισμό των προγραμμάτων antivirus, καθώς επίσης πως μπορούν να παρακάμπτουν το τείχος προστασίας και άλλες προστασίες ασφάλειας. Η στατική ανάλυση ακόμα βοηθά τους ερευνητές να αποκαλύψουν τι μπορεί να κάνει ένα κομμάτι του malware και πώς να το σταματήσουν. Επιπρόσθετα, για να επιτευχτεί στατική ανάλυση, οι ερευνητές θα πρέπει να έχουν καλή γνώση της γλώσσας assembly και του λειτουργικού συστήματος στόχου.

Με την *τεχνική reverse-engineering,* οι ερευνητές είναι σε θέση να μετατρέψουν την γλώσσα assembly, σε γλώσσα υψηλότερου επιπέδου. Η αντίστροφη μηχανική (reverse engineering) είναι η μόνη λύση για την κατανόηση και την απόκτηση του πηγαίου κώδικα, από προγράμματα κλειστού κώδικα. Τα εργαλεία στατικής ανάλυσης περιλαμβάνουν αναλυτές προγράμματος, debuggers και disassemblers. Αυτά τα

εργαλεία είναι σε θέση να ανιχνεύσουν αν το κακόβουλο λογισμικό χρησιμοποιεί κάποια από τις τεχνικές προστασίας του λογισμικού.

2.3 Reverse Engineering

Οι προγραμματιστές των λογισμικών anti-virus, τεμαχίζουν κάθε πρόγραμμα κακόβουλου λογισμικού που πέφτει στα χέρια τους, με την χρήση τεχνικών *reverse engineering*. Οι τεχνικές reversing (αντιστροφής), απαιτούν την ικανότητα κατανόησης της εσωτερικής δομής των προγραμμάτων, την γλώσσα assembly και το λειτουργικό σύστημα. Προσπαθούν να εξάγουν πολύτιμες πληροφορίες από τα προγράμματα για τα οποία ο πηγαίος κώδικας δεν είναι διαθέσιμος και είναι επίσης δυνατόν να εξάγουν δεδομένα όλων των προγραμμάτων, συμπεριλαμβανομένου του αρχικού (ή παρόμοιου) πηγαίου κώδικα. Συνήθως διεξάγεται για την απόκτηση γνώσεων που λείπουν, ιδεών, και σχεδιαστικής φιλοσοφίας όταν οι πληροφορίες δεν είναι διαθέσιμες. Σε κάποιες περιπτώσεις οι πληροφορίες ανήκουν σε κάποιον που δεν έχει ως σκοπό να τις μοιραστεί και σε άλλες περιπτώσεις οι πληροφορίες έχουν χαθεί ή έχουν καταστραφεί.

Για μερικούς ανθρώπους η σχέση μεταξύ της ασφάλειας και της αντιστροφής μπορεί να μην είναι αμέσως σαφές. Η αντιστροφή σχετίζεται με διάφορες πτυχές της ασφάλειας των υπολογιστών. Για παράδειγμα, η αντιστροφή έχει χρησιμοποιηθεί στην κρυπτογραφική έρευνα, δηλαδή, ένας ερευνητής αντιστρέφει ένα προϊόν κρυπτογράφησης και αξιολογεί το επίπεδο ασφάλειας που αυτό παρέχει. Επίσης σχετίζεται σε πολύ μεγάλο βαθμό με το κακόβουλο λογισμικό, που αυτό σημαίνει ότι χρησιμοποιείται τόσο από τους προγραμματιστές malware καθώς και από εκείνους που θα αναπτύξουν τα αντίδοτα για την αντιμετώπισή τους.

Οι προγραμματιστές κακόβουλων λογισμικών, συχνά χρησιμοποιούν την αντιστροφή για να εντοπίσουν τρωτά σημεία στα λειτουργικά συστήματα και σε άλλα λογισμικά. Τέτοια τρωτά σημεία, μπορούν να χρησιμοποιηθούν για να διαπεράσουν τα στρώματα άμυνας του συστήματος και να προκαλέσουν μόλυνση- συνήθως μέσω Internet. Πέρα από την μόλυνση, οι τεχνικές αντίστροφης μηχανικής

επιτρέπουν σε ένα κακόβουλο πρόγραμμα να αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες ή ακόμα και να λάβει τον πλήρη έλεγχο της πληροφορίας.

Οι προγραμματιστές προγραμμάτων antivirus, τεμαχίζουν και αναλύουν κάθε κακόβουλο πρόγραμμα που πέφτει στα χέρια τους. Χρησιμοποιούν τις τεχνικές reversing για την ανίχνευση του κάθε βήματος που κάνει το πρόγραμμα, να αξιολογήσουν την ζημιά που θα μπορούσε να προκαλέσει στο σύστημα, πώς θα μπορέσει να αφαιρεθεί από το σύστημα καθώς και αν η μόλυνση θα μπορούσε να αποφευχθεί εντελώς.

2.3.1 **Τα Εργαλεία**

Η αντιστροφή έχει να κάνει με τα εργαλεία. Παρακάτω θα γίνει περιγραφή των βασικών κατηγοριών των εργαλείων που χρησιμοποιούνται στην αντίστροφη μηχανική. Πολλά από αυτά τα εργαλεία δεν έχουν δημιουργηθεί ειδικά για την αντίστροφη μηχανική, ωστόσο είναι πολύ χρήσιμα.

- ***System-Monitoring Tools***

Η αντιστροφή συστήματος σε επίπεδα απαιτεί μια ποικιλία εργαλείων τα οποία ανιχνεύουν, παρακολουθούν, διερευνούν, ή και διαφορετικά, ξεσκεπάζουν το πρόγραμμα ώστε να αντιστρέφεται. Τα περισσότερα από αυτά τα εργαλεία εμφανίζουν πληροφορίες που σχετίζονται με την εφαρμογή και το περιβάλλον του, τα οποία συλλέγονται από το λειτουργικό σύστημα. Επειδή σχεδόν όλες οι επικοινωνίες μεταξύ του προγράμματος και του έξω κόσμου περνάνε μέσα από το λειτουργικό σύστημα, αυτά συνήθως μπορούν να αξιοποιηθούν για την εξαγωγή τέτοιων πληροφοριών. Τα εργαλεία συστημάτων παρακολούθησης μπορούν να παρακολουθούν διάφορες κινήσεις δικτύωσης, προσβάσεις σε αρχεία ή στην registry καθώς και σε διάφορα άλλα. Υπάρχουν επίσης εργαλεία που εκθέτουν την χρήση ενός προγράμματος του

λειτουργικού συστήματος, τέτοια αντικείμενα είναι τα mutexes, pipes, events και διάφορα άλλα.

- ***Disassemblers***

Τα disassemblers, είναι προγράμματα τα οποία λαμβάνουν τον εκτελέσιμο δυαδικό κώδικα ενός προγράμματος ως είσοδο και παράγουν αρχεία κειμένου που περιέχουν γλώσσα assembly για ολόκληρο το πρόγραμμα ή τμήματα του. Αυτή είναι μια σχετικά απλή διαδικασία, θεωρώντας ότι ο κώδικας της γλώσσας assembly είναι απλά μια κειμενική χαρτογράφηση του καταληκτικού κώδικα. Ένας υψηλής ποιότητας disassembler, είναι το βασικό κλειδί στην εργαλειοθήκη ενός αντιστροφέα, όμως μερικοί αντιστροφείς προτιμούν να χρησιμοποιούν ενσωματωμένους disassemblers τα οποία περιέχονται σε κάποια χαμηλού επιπέδου προγράμματα εντοπισμού σφαλμάτων (debuggers).

- ***Debuggers***

Η βασική ιδέα πίσω από ένα debugger είναι ότι οι προγραμματιστές δεν μπορούν στην πραγματικότητα να δουν όλα αυτά που κάνει ένα πρόγραμμα. Τα προγράμματα συνήθως είναι πάρα πολύ σύνθετα για ένα άνθρωπο ώστε να προβλέψει στην πραγματικότητα κάθε δυνατό αποτέλεσμα. Ένα πρόγραμμα εντοπισμού σφαλμάτων είναι ένα πρόγραμμα που επιτρέπει στους προγραμματιστές λογισμικών να παρατηρούν και να ελέγχουν τα προγράμματά τους, ενώ αυτά βρίσκονται σε λειτουργία. Τα δυο πιο βασικά χαρακτηριστικά ενός προγράμματος εντοπισμού σφαλμάτων είναι η δυνατότητα ορισμού σημείων διακοπής κατά την διαδικασία και η ικανότητα ανίχνευσης διαμέσου του κώδικα.

Με αυτά τα δύο σημαντικά χαρακτηριστικά, οι προγραμματιστές μπορούν να παρακολουθούν στενά τα προγράμματά τους, δεδομένου ότι εκτελεί ένα προβληματικό τμήμα του κώδικα και να προσπαθήσουν να καθορίσουν την πηγή του προβλήματος.

Για ένα αναστροφέα, το πρόγραμμα εντοπισμού σφαλμάτων (debugger) είναι σχεδόν τόσο σημαντικό όσο είναι σε έναν επαγγελματία ανάπτυξης λογισμικού, αλλά για λίγο διαφορετικούς λόγους. Πρώτο και κυριότερο, οι αντιστροφείς χρησιμοποιούν προγράμματα εντοπισμού σφαλμάτων σε λειτουργία αποσυναρμολόγησης (disassembly). Στην λειτουργία αποσυναρμολόγησης, ένας debugger χρησιμοποιεί ένα ενσωματωμένο disassembler για συνεχή αποσυναρμολόγηση του καταληκτικού κώδικα.

- ***Decompilers***

Τα decompilers είναι το επόμενο βήμα από τους disassemblers. Ένας decompiler παίρνει ένα εκτελέσιμο δυαδικό αρχείο και προσπαθεί από αυτό να παράγει αναγνώσιμο κώδικα μιας γλώσσας υψηλού επιπέδου. Η ιδέα είναι να γίνει μια προσπάθεια και να αντιστραφεί η διαδικασία compile, έτσι ώστε να αποκτηθεί το αρχικό αρχείο ή κάτι παρόμοιο με αυτό. Υπάρχουν σημαντικά στοιχεία στις περισσότερες γλώσσες υψηλού επιπέδου που παραλείπονται κατά τη διάρκεια της διαδικασίας compile και είναι αδύνατο να ανακτηθούν. Ακόμα, οι decompilers είναι ισχυρά εργαλεία τα οποία σε ορισμένες καταστάσεις και περιβάλλοντα μπορούν να ανακατασκευάσουν από δυαδική μορφή έναν υψηλά αναγνώσιμο πηγαίο κώδικα.

ΚΕΦΑΛΑΙΟ 4

ΠΡΟΕΤΟΙΜΑΣΙΑ

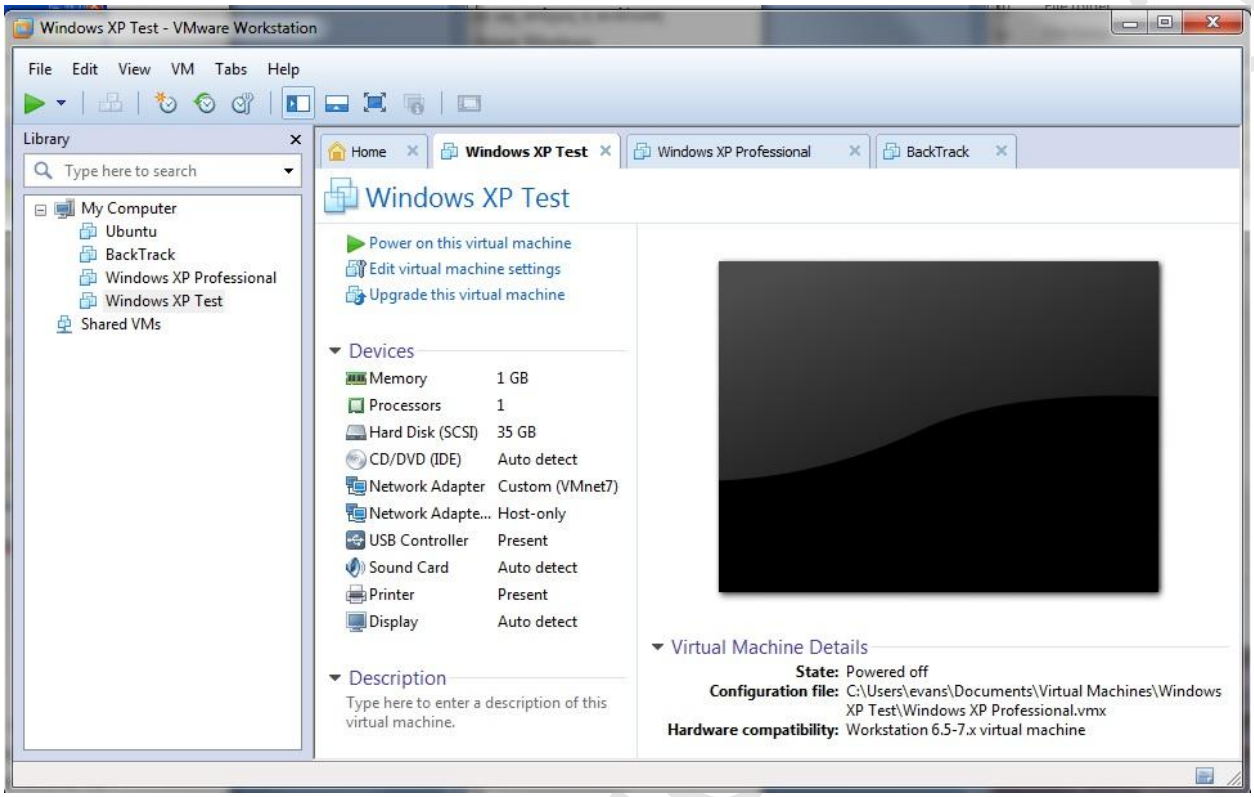
3.1 Επιλογή του Malware

Υπάρχουν πολλά είδη κακόβουλου λογισμικού που θα μπορούσαν να επιλεγούν για μελέτη και ανάλυση. Για την συγκεκριμένη περίπτωση τέθηκε ως στόχος η ανάλυση ενός Worm (σκουληκι) το οποίο “μολύνει” το λειτουργικό σύστημα Windows.
Μια αξιόπιστη ιστοσελίδα που παρέχει κακόβουλο λογισμικό για μελέτη είναι η www.offensivecomputing.net . Σε αυτήν μπορεί οποιοσδήποτε να κατεβάσει, μετά βεβαία από μια διαδικασία εγγραφής για ευνόητους λόγους, το κακόβουλο πρόγραμμα που τον ενδιαφέρει σε αρχείο .zip.

3.2 Δημιουργία Εικονικού Περιβάλλοντος

Όπως είναι φυσικό, για την μελέτη του συγκεκριμένου κακόβουλου κώδικα δεν θα χρησιμοποιηθεί το κύριο λειτουργικό σύστημα, στο οποίο θα δημιουργούσε δυσάρεστες συνέπειες, άλλα σκοπός είναι να δημιουργηθεί ένα εικονικό περιβάλλον που θα επιτρέψει να μελετήσουμε με ασφάλεια όλες τις επιπτώσεις του κακόβολου λογισμικού.

Ένα τέτοιο εικονικό περιβάλλον είναι το *VM Workstation* (εικόνα 1) του οποίου η δωρεάν έκδοση VMware Player διατίθεται δωρεάν από την ιστοσελίδα www.vmware.com. Το λειτουργικό σύστημα που επιλέξαμε να τρέχει στο εικονικό περιβάλλον είναι τα Windows XP και μέσα σε αυτό περάστηκαν όλα τα απαραίτητα προγράμματα για να ξεκινήσει η μελέτη και ανάλυση του Malware μας. Επιπλέον στο εικονικό περιβάλλον εγκαταστάθηκε και το λειτουργικό σύστημα BackTrack, το οποίο θα λειτουργήσει βοηθητικά όπως θα δούμε στην συνέχεια.



Εικόνα 1. Το πρόγραμμα VM WorkStation

Έχουμε πλέον εγκαταστήσει σε εικονικό περιβάλλον τα δύο λειτουργικά συστήματα που θα χρειαστούμε για την ανάλυση του malware μας, οπότε μπορούμε να προχωρήσουμε. Στα δύο παρακάτω κεφάλαια θα δούμε την δυναμική και την στατική ανάλυση. Πριν προχωρήσουμε όμως ας δούμε τα προγράμματα τα οποία μας βοήθησαν στο να επιτευχθεί αυτή η ανάλυση

3.3 Εργαλεία

- **VMware Workstation**

Ένα από τα ποιο διάσημα πρόγραμμα για Virtualization. Χρησιμοποιήθηκε τόσο για να εγκαταστήσουμε το μηχάνημα ανάλυσης (Windows XP) όσο και το μηχάνημα ελέγχου (BackTrack).

- **Depedency Walker**

Η εφαρμογή αυτή λειτουργεί εξετάζοντας σε βάθος των κώδικα ενός αρχείου (exe, dll, ocx, sys, κλπ) και στη συνέχεια δημιουργώντας ένα ιεραρχικό δενδροδιάγραμμα με όλες τις εξαρτήσεις των επιμέρους δομικών στοιχείων (module) του . Πρακτικά αυτό σημαίνει ότι μπορεί να εντοπίσει όλα τα dll που καλεί το αρχείο, το ποιές συναρτήσεις συγκεκριμένα χρησιμοποιεί, αλλά και τα dll και συναρτήσεις που αυτές με τη σειρά τους αξιοποιούν. Αξίζει να σημειωθεί ότι κάθε ένα από τα ονόματα των συναρτήσεων λειτουργεί ως ενεργός σύνδεσμος, παραπέμποντας στην αντίστοιχη σελίδα του MSDN για περαιτέρω πληροφορίες.

- **UPX**

Ο UPX (Ultimate Packer for eXecutables) είναι πρόγραμμα packer. Συμπιέζει το αρχικό πρόγραμμα και δυσχεραίνει σε σημαντικό βαθμό την ανάλυσή του. Διαθέτει και διαδικασία αποσυμπίεσης ,η οποία και χρησιμοποιήθηκε.

- **FakeDNS**

Μικρός DNS server. Μπορεί να τρέχει τοπικά ή απομακρυσμένα και προωθεί όλες τις εισερχόμενες αιτήσεις, είτε σε προκαθορισμένη IP, είτε στο localhost.

- **Active Ports**

Εφαρμογή που παρακολουθεί και καταγράφει σε πραγματικό χρόνο όλες τις ενεργές πόρτες, ποιες διεργασίες είναι συνδεδεμένες σε αυτές καθώς και τις εξωτερικές IP που επιχειρούν να επικοινωνήσουν με το σύστημα.

- **WireShark**

Το Wireshark είναι εργαλείο για την ανάλυση δικτυακών πρωτοκόλλων. Επιτρέπει τη σύλληψη και μελέτη σε πραγματικό χρόνο όλων των δεδομένων που διακινούνται σε κάποιο δίκτυο.

- **Netcat**

Χρησιμοποιώντας είτε το πρωτόκολλο TCP είτε το UDP , το Netcat μπορεί να διαβάσει και να γράψει δεδομένα πάνω από σύνδεση δικτύου. Σε αυτή την μελέτη περίπτωσης χρησιμοποιήθηκε σαν server ρυθμισμένος να «ακούει» σε μια προκαθορισμένη θύρα.

- **Process Monitor**

Εξελιγμένο εργαλείο παρακολούθησης για το περιβάλλον Windows που δείχνει σε πραγματικό χρόνο το σύστημα αρχείων, την Registry και την κακόβουλη δραστηριότητα. Συνδυάζει τα χαρακτηριστικά των δύο βοηθητικών προγραμμάτων της εταιρίας Sysinternals του FileMon και του RegMon.

- **Process Explorer**

Βοηθητικό πρόγραμμα που παρουσιάζει διευθύνσεις και διαδικασίες DLL έχουν φορτωθεί στο σύστημα

- **RegShot**

Βοηθιτικό open-source πρόγραμμα για σύγκριση της Registry. Συγκεκριμένα επιτρέπει την λήψη και σύγκριση δύο στιγμιότυπων (πριν και μετά το τρέξιμο του malware) από την Registry.

- **Autoruns**

Χρήσιμο πρόγραμμα το οποίο εμφανίζει τα προγράμματα – διεργασίες που είναι ρυθμισμένες να τρέχουν κατά την εκκίνηση του λειτουργικού συστήματος και τα παρουσιάζει με την σειρά που το λειτουργικό σύστημα τα επεξεργάζεται.

- **BinText**

Ένα εξαιρετικά χρήσιμο εργαλείο για την εξαγωγή συμβολοσειρών (strings) ASCII και UNICODE χαρακτήρων. Χρήσιμο για τον εντοπισμό hardcoded λέξεων όπως URLs ή ονόματα συναρτήσεων. Εντοπίζει τόσο τη θέση τους μέσα στο αρχείο, όσο και τη θέση που θα καταλάβουν στη μνήμη κατα την ώρα της εκτέλεσης.

- **PEID**

Πρόγραμμα το οποίο μας δίνει πληροφορίες σχετικά με τον τύπο του packer που έχει χρησιμοποιηθεί ή αν δεν έχει χρησιμοποιηθεί κάποιος packer μας δίνει πληροφορίες σχετικά με τον compiler με τον οποίον έγινε compile το εκτελέσιμο.

- **IDA PRO**

Ένας από τους δυνατότερους Disassemblers (αντίστροφος συμβολομεταφραστής) που αποτελεί την βασική επιλογή των περισσοτέρων αναλυτών κακόβουλων λογισμικών και γενικά των αναλυτών ευπαθειών. Υποστηρίζει πολλά file formats όπως Portable Executable (PE), Common Object File Format (COFF), Executable and Linking Format (ELF), and a.out. Οι δύο βασικές εκδόσεις τους υποστηρίζουν μόνο επεξεργαστές x86 ενώ η πιο προηγμένη έκδοση υποστηρίζει και αρχιτεκτονικές x64. Εκτός του κώδικα assembly του εκτελέσιμου (σε μορφή κειμένου και γράφου) μας δίνει και πληροφορίες σχετικά με τις εισαγόμενες συναρτήσεις (imported functions) και τις συμβολοσειρές (strings) που χρησιμοποιεί το εκτελέσιμο.

ΚΕΦΑΛΑΙΟ 5

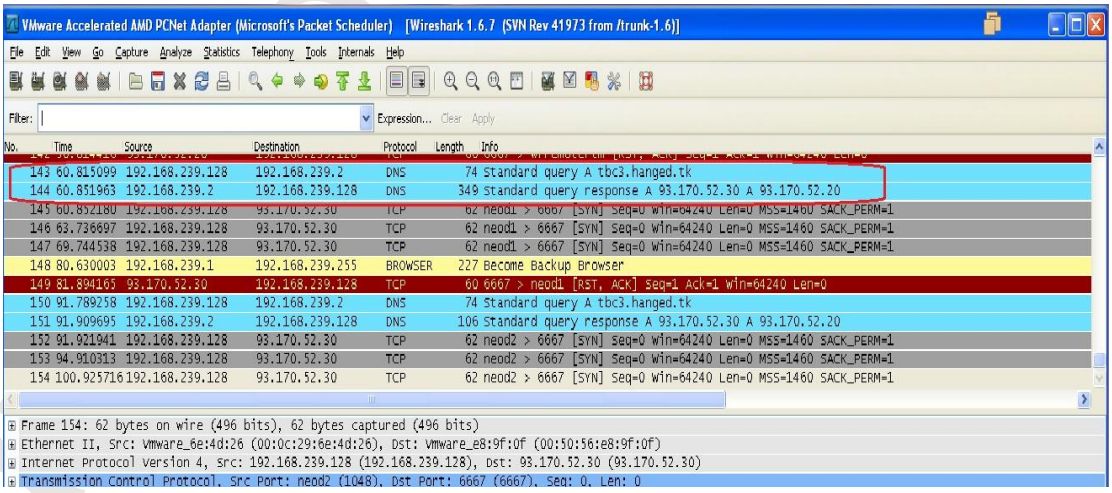
Δυναμική Ανάλυση

4.1 Μελέτη της Δικτυακής Συμπεριφοράς του Malware

Με την εκτέλεση του malware στον υπολογιστή αναμένεται να υπάρξει κάποια δικτυακή επικοινωνία λόγω της φύσης του malware. Έτσι λοιπόν αφού τρέξουμε το malware σε πραγματικές συνθήκες για μικρό χρονικό διάστημα (για λόγους ασφαλείας), για να επιβεβαιώσουμε μέσω του wireshark ότι όντως γίνεται προσπάθεια επικοινωνίας, θα προσπαθήσουμε να κάνουμε μια λεπτομερέστερη παρατήρηση της σύνδεσης αυτής (αν υπάρχει) σε ελεγχόμενο πλέον περιβάλλον.

4.1.1 Μελέτη σε Πραγματικές Συνθήκες

Σε ένα μηχάνημα με λειτουργικό Windows Xp και σύνδεση στο Διαδίκτυο εκτελούμε το malware ενώ παράλληλα τρέχουμε την εφαρμογή WireShark στο ίδιο το μηχάνημα με σκοπό να παρακολουθήσουμε την (εξερχόμενη) Δικτυακή κυκλοφορία. Παρατηρώντας την δικτυακή κυκλοφορία στο WireShark *(εικόνα 2)* βλέπουμε πως αποστέλλεται αρχικά ένα DNS query για την διεύθυνση *TBC3.HANGED.TK* και στην συνέχεια λαμβάνεται απάντηση σε αυτό το DNS query με την IP η οποία αντιστοιχεί σε αυτή την διεύθυνση. Στην συνέχεια γίνεται απόπειρα σύνδεσης σε αυτήν την διεύθυνση σε συγκεκριμένη θύρα (6667).



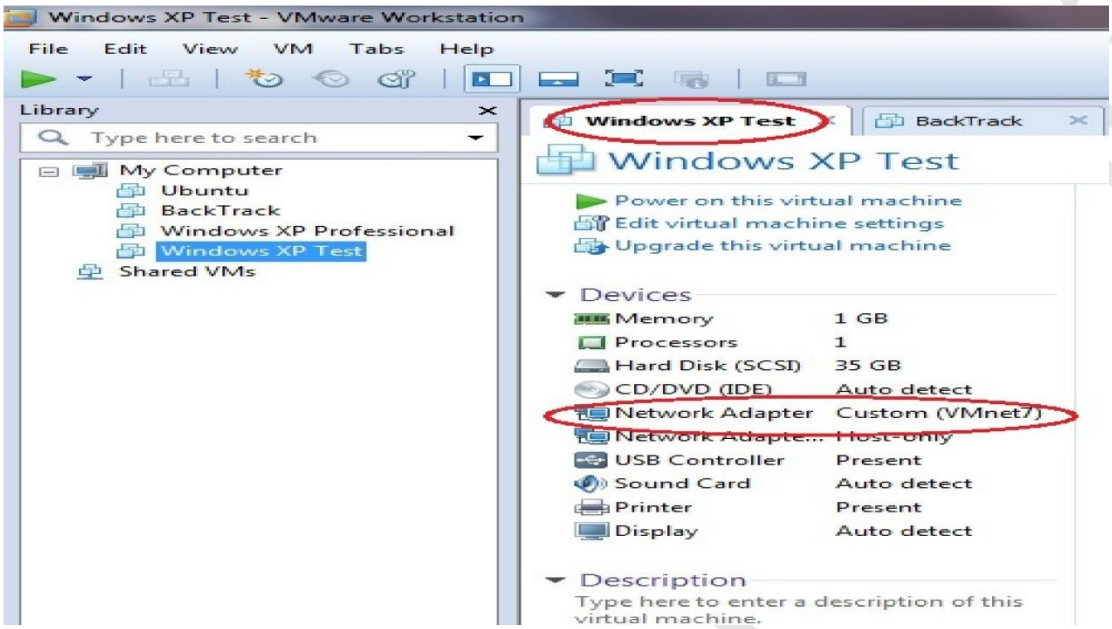
Εικονα 2. Ανάλυση Δικτυακής Κυκλοφορίας μέσω WireShark

Για λόγους ασφαλείας περεταίρω ανάλυση δικτυακής επικοινωνίας του malware σε πραγματικό περιβάλλον σταματάει εδώ. Θα συνεχίσουμε την ανάλυση σε ελεγχόμενο περιβάλλον.

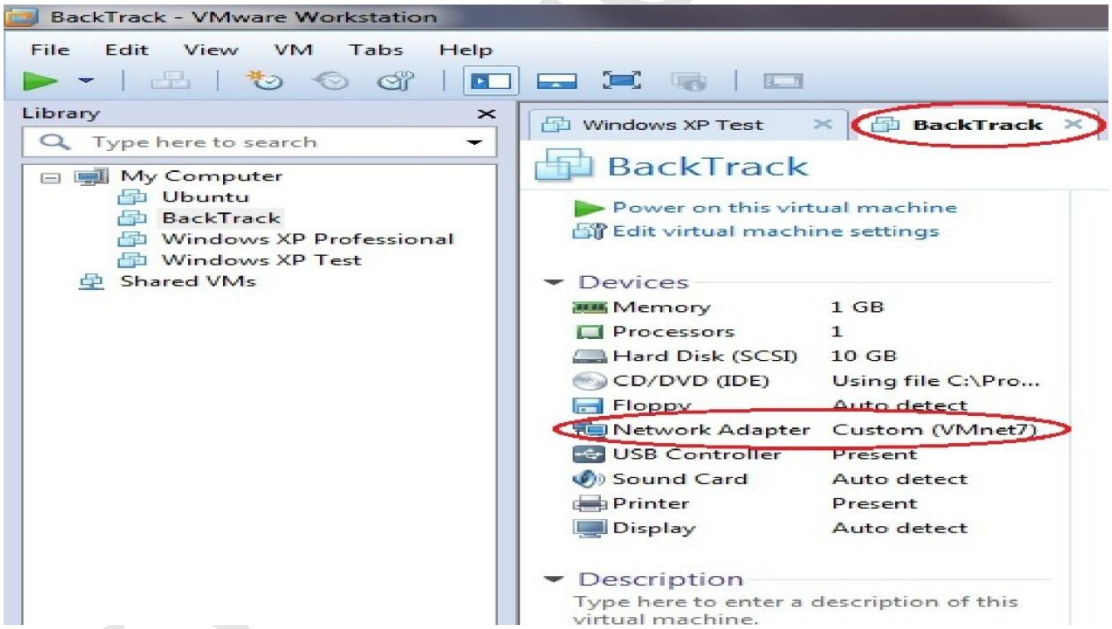
4.1.2 Μελέτη σε Ελεγχόμενο Περιβάλλον

Προκειμένου να αναλυθεί λεπτομερέστερα και με μεγαλύτερη ασφάλεια η δικτυακή επικοινωνία την οποία επιχειρεί το malware κρίθηκε απαραίτητο να δημιουργηθεί ένα ελεγχόμενο περιβάλλον εντός του οποίου θα γίνει η ανάλυση. Πως θα είναι όμως αυτό το ελεγχόμενο περιβάλλον; Ίσως υπάρχουν αρκετοί τρόποι να “στηθεί” ένα ελεγχόμενο περιβάλλον δικτυακής ανάλυσης – παρακολούθησης, εμείς κρίναμε πως το βασικό σκεπτικό και οι γενικές αρχές λειτουργίας του θα πρέπει να είναι η εξής : Σε ένα μηχάνημα με λειτουργικό windows (μηχάνημα ανάλυσης) θα εγκατασταθεί το malware (πρόκειται για worm εκτελέσιμο σε windows). Το μηχάνημα ελέγχου θα ρυθμιστεί έτσι ώστε όλη η εξερχόμενη δικτυακή κυκλοφορία να δρομολογείται προς ένα μηχάνημα ελέγχου με λειτουργικό BackTrack ώστε να αναλυθεί και να επιχειρηθεί να προσομοιωθεί οποιαδήποτε κακόβουλη επικοινωνία.

Τα δύο μηχανήματα εγκαταστάθηκαν ως εικονικά μηχανήματα (Virtual machines) με χρήση του VM Workstation. Πριν την εκκίνηση τους προσέχουμε να έχουμε ρυθμίσει να ανήκουν στο ίδιο εικονικό δίκτυο, όπως φαίνεται στις *εικόνες 3 και 4* για windows Xp και BackTrack5 αντίστοιχα.



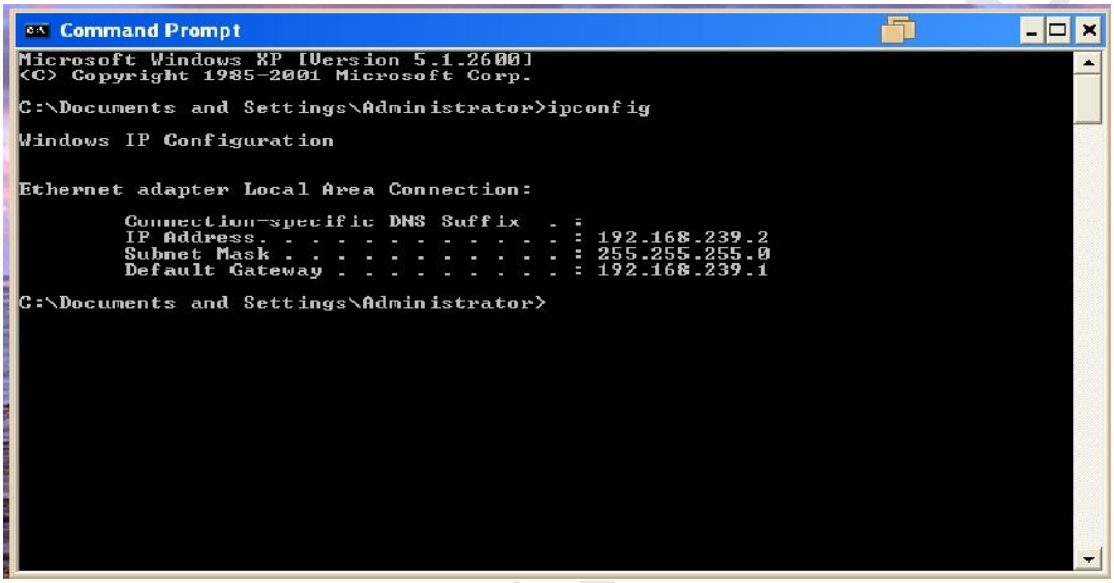
Εικόνα 3. Ρυθμίσεις Δικτύου Εικονικού Μηχανήματος Windows XP



Εικόνα 4. Ρυθμίσεις Δικτύου Εικονικού Μηχανήματος BackTrack5

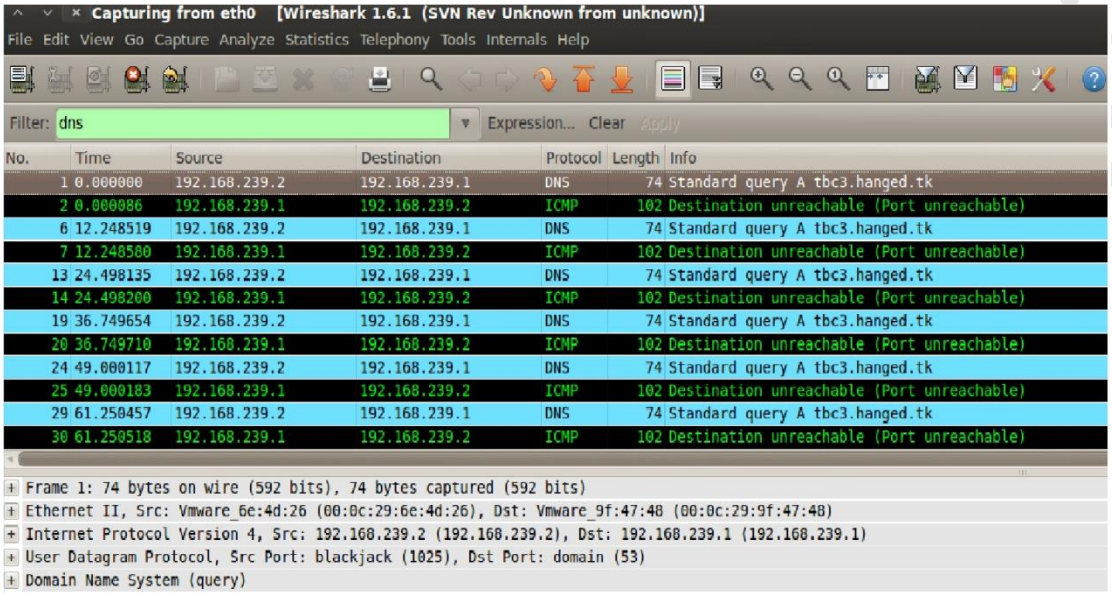
Στην συνέχεια τροποποιούμε τις ρυθμίσεις δικτύου των δύο μηχανημάτων. Ορίζουμε μια IP για το μηχάνημα ανάλυσης(Windows Xp) την 192.168.239.2 και μια IP για το μηχάνημα ελέγχου (BackTrack) την 192.168.239.1 όπως φαίνεται στην *εικόνα 5*. Στην

συνέχεια στο μηχάνημα ανάλυσης χρησιμοποιούμαι ως προεπιλεγμένη πύλη (Default Gateway) την διεύθυνση IP του μηχανήματος ελέγχου. Έτσι έχουμε πετύχει να δρομολογήσουμε όλη την εξερχόμενη δικτυακή κυκλοφορία από το μηχάνημα ανάλυσης στο μηχάνημα ελέγχου.



Εικόνα 5. Ρυθμίσεις Δικτύου Μηχανήματος Ανάλυσης

Στην συνέχεια εκτελούμε το malware στο μηχάνημα ανάλυσης ενώ στο μηχάνημα ελέγχου εκτελούμε το wireshark ώστε να μελετήσουμε σε πραγματικό χρόνο τα δεδομένα που διακινούνται στο δίκτυο που στήσαμε και κατά συνέπεια να ελέγξουμε αν όλα λειτουργούν όπως περιμένουμε. Όπως φαίνεται και στην *εικόνα 6* η επικοινωνία μεταξύ των δύο μηχανημάτων είναι επιτυχής. Συγκεκριμένα το μηχάνημα ανάλυσης στέλνει DNS queries στο μηχάνημα ελέγχου σχετικά με την διεύθυνση TBC3.HANGED.TK. Δεν μπορεί να γίνει περαιτέρω ανάλυση αυτής της απόπειρας σύνδεσης του μηχανήματος ανάλυσης με την διεύθυνση TBC3.HANGED.TK και αυτό διότι το μηχάνημα ελέγχου δεν μπορεί να απαντήσει σε αυτό το DNS query (Δεν γνωρίζει την IP που αντιστοιχεί σε αυτήν την διεύθυνση). Χρειάζεται περαιτέρω ρύθμιση του ελεγχόμενου δικτύου ώστε με κάποιον τρόπο το μηχάνημα ελέγχου να δέχεται ως απάντηση στο DNS query της διεύθυνσης αυτής την ίδια την IP του μηχανήματος ελέγχου έτσι ώστε να παρατηρήσουμε πως θα συμπεριφερθεί.

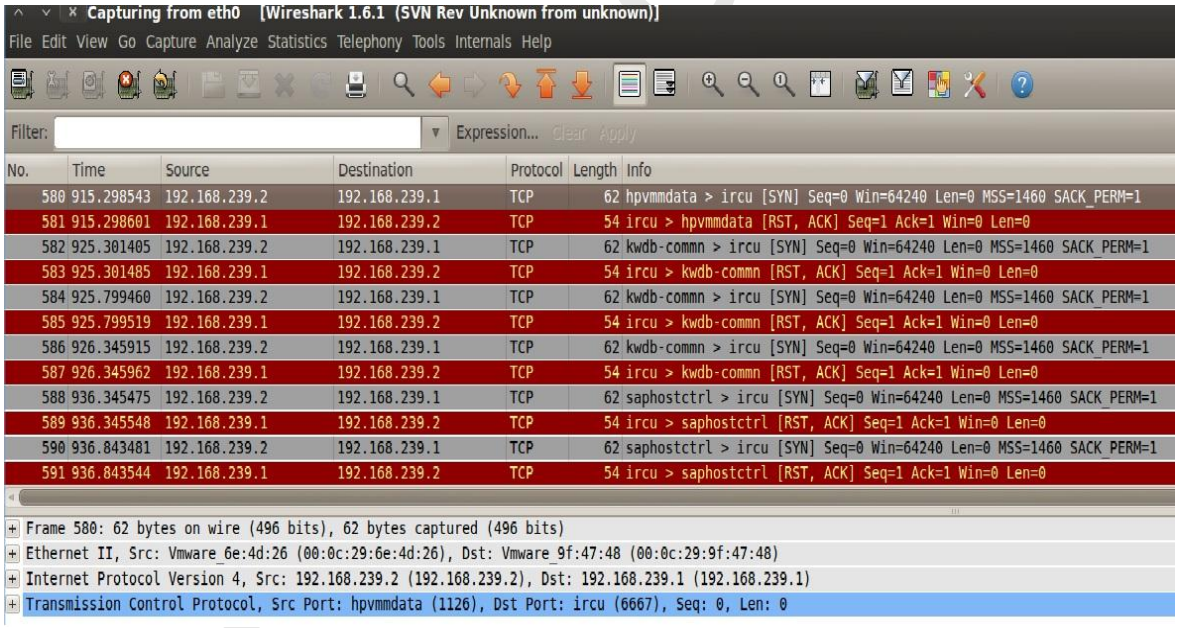


Εικόνα 6. Ανάλυση Δικτυακής Κυκλοφορίας μέσω WireShark

Η εφαρμογή mini FakeDNS μπορεί να μας βοηθήσει σε αυτό το σημείο. Εγκαθίσταται στο μηχάνημα ανάλυσης και ρυθμίζεται ώστε να απαντάει σε οποιοδήποτε DNS query με την IP του μηχανήματος ελέγχου. Αξίζει να σημειωθεί σε αυτό το σημείο ότι το FakeDNS λειτουργεί τοπικά στο μηχάνημα ελέγχου και έτσι στις ρυθμίσεις δικτύου του μηχανήματος αυτού πρέπει να ρυθμιστεί ως DNS server η IP διεύθυνση του ίδιου του μηχανήματος.

Κάνοντας τα παραπάνω το FakeDNS πρέπει να πιάσει το DNS query για την διεύθυνση TBC3.HANGED.TK και να επιστρέψει στο μηχάνημα ανάλυσης την διεύθυνση IP του μηχανήματος ελέγχου. Έτσι θα μπορέσουμε να παρατηρήσουμε τι δεδομένα θα προσπαθήσει να στείλει στην διεύθυνση αυτή. Όπως φαίνεται στην *εικόνα 7* το FakeDNS λειτούργησε αποτελεσματικά και έτσι το μηχάνημα ανάλυσης προωθεί την κυκλοφορία σε αυτήν την IP που πιστεύει πως αντιστοιχεί στην ύποπτη διεύθυνση δηλαδή την IP του μηχανήματος ελέγχου. Το wireshark στο μηχάνημα ελέγχου συλλαμβάνει αυτήν την επικοινωνία όπως φαίνεται στην *εικόνα 8*.

Εικόνα 7. Απάντηση του FakeDNS



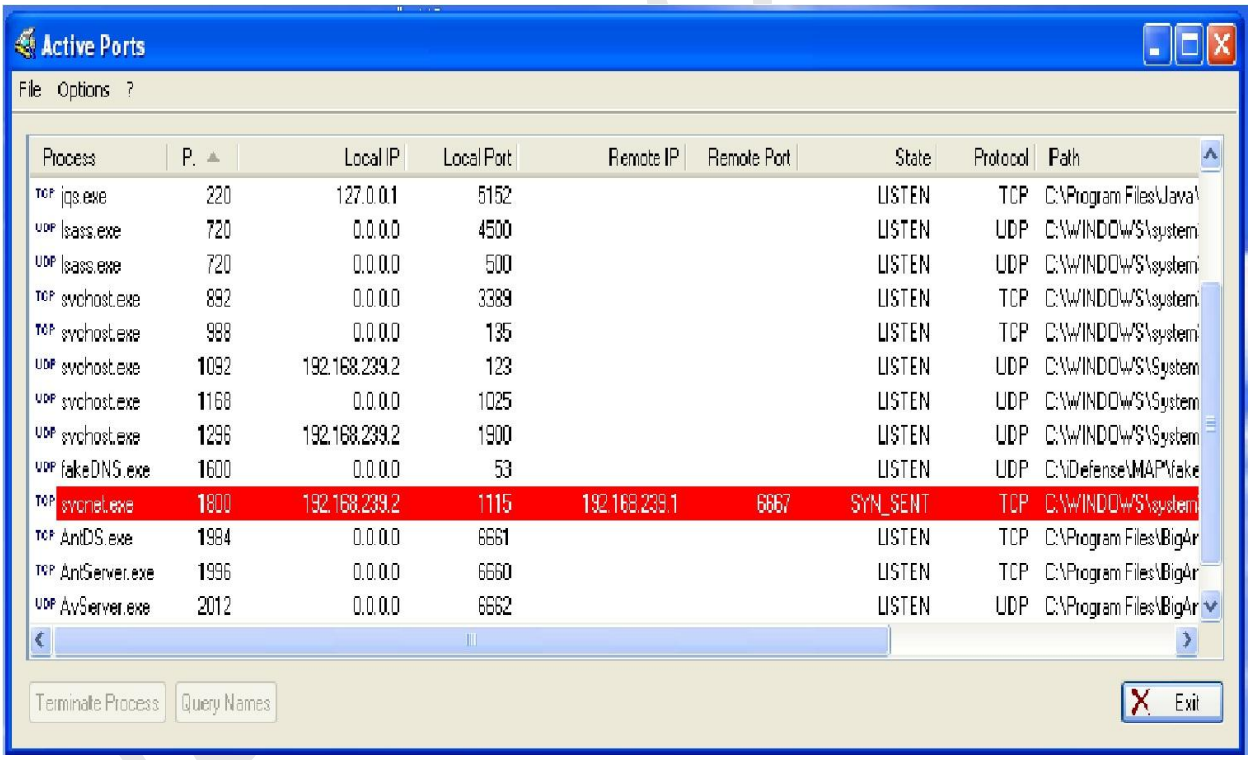
Εικόνα 8. Επικοινωνία Μηχανήματος Ανάλυσης με την Διεύθυνση TBC3.HANGED.TK στο WireShark

Παρατηρούμε πως το μηχάνημα ανάλυσης επιχειρεί να συνδεθεί στο μηχάνημα ελέγχου στην θύρα 6667 στέλνοντας ένα πακέτο SYN από την θύρα 1126. Το μηχάνημα ελέγχου απαντάει σε κάθε SYN πακέτο με ένα RST γεγονός που σημαίνει πως η σύνδεση αυτή δεν πραγματοποιήθηκε προφανώς επειδή η θύρα 6667 είναι κλειστή. Αν

αναλυθεί η κίνηση μερικών λεπτών παρατηρεί κανείς εύκολα πως γίνονται τρείς προσπάθειες σύνδεσης από την ίδια θύρα του μηχανήματος ελέγχου και στην συνέχεια οι προσπάθειες συνεχίζονται ανά τρείς από την επόμενη θύρα κατά αύξουσα σειρά.

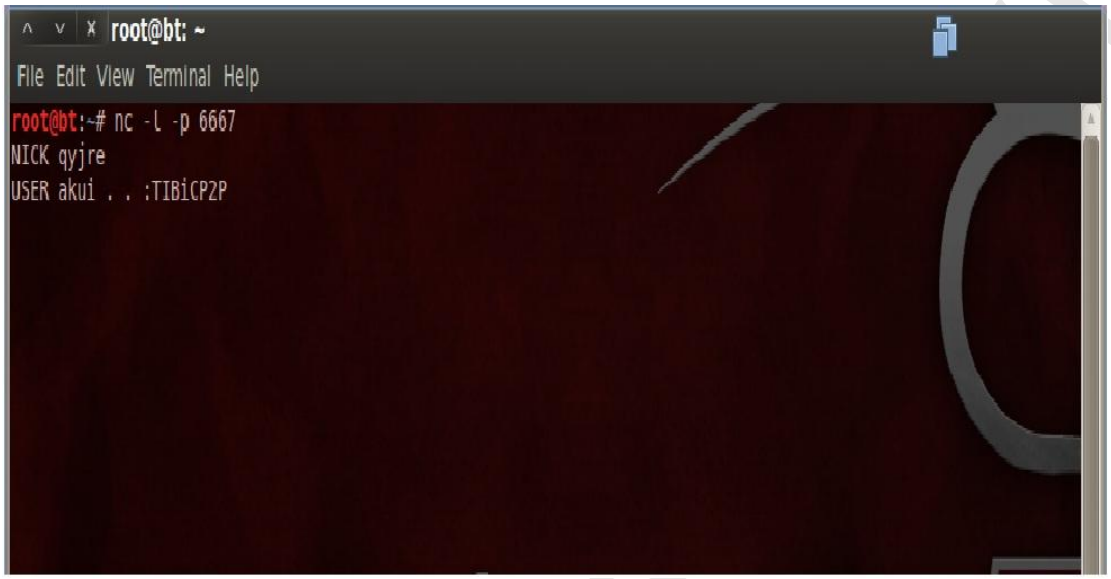
Η απόπειρα σύνδεσης επιβεβαιώνεται και από την εφαρμογή ActivePorts (*εικόνα 9)* παρατηρώντας μάλιστα πως το όνομα της διεργασίας που την επιχειρεί είναι svcnet.exe από την τοποθεσία C:\Windows\System32. Δεδομένου ότι σε ένα λειτουργικό σύστημα Windows τρέχουν αρκετές νόμιμες διεργασίες με το όνομα svchost είναι πιθανό επιλέχθηκε το συγκεκριμένο όνομα για την ύποπτη διεργασία με την πεποίθηση να περάσει απαρατήρητη.

Σε αυτό το σημείο είναι λογικό να σκεφτεί κανείς πως πρέπει να ανοίξει η θύρα 6667 του μηχανήματος ελέγχου έτσι ώστε να επιτευχθεί η σύνδεση και να παρατηρηθεί τι γίνεται κατόπιν αυτής.



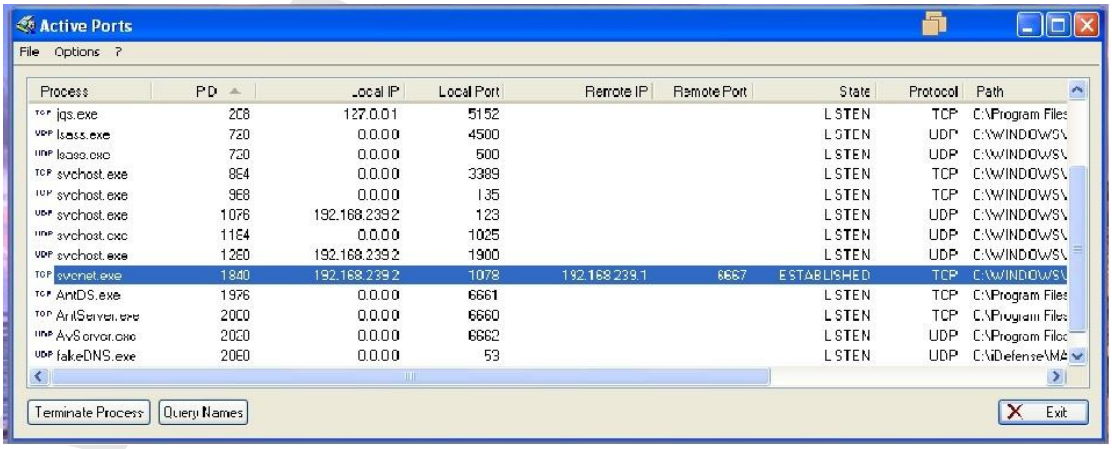
Εικόνα 9. Προσπάθεια Σύνδεσης Μηχανήματος Ανάλυσης με το Μηχάνημα Ελέγχου στην θύρα 6667 στο ActivePorts

Έτσι λοιπόν στο μηχάνημα ελέγχου ρυθμίζουμε την εφαρμογή netcat έτσι ώστε να περιμένει νέες συνδέσεις στην θύρα 6667 όπως φαίνεται στην *εικόνα 10*.

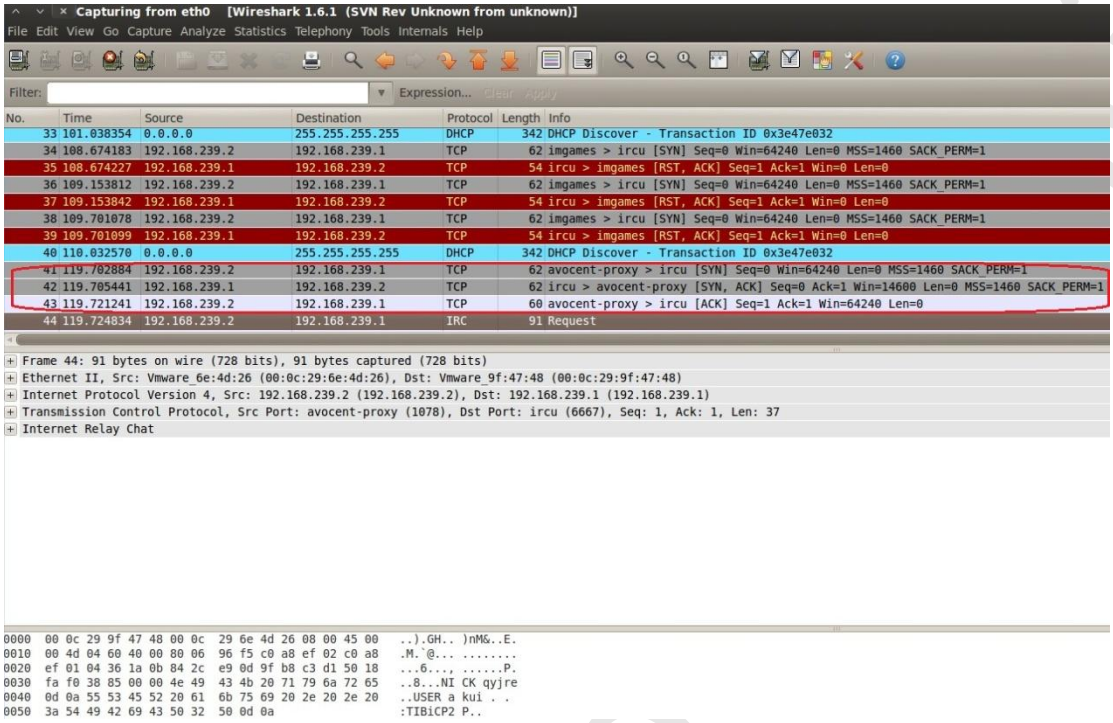


Εικόνα 10. Άνοιγμα θύρας 6667 στο Μηχάνημα Ελέγχου

Μόλις ρυθμίστηκε η θύρα 6667 να ακούει για εισερχόμενες συνδέσεις, το malware έστειλε δεδομένα. Πρόκειται μάλλον για όνομα χρήστη και κωδικό qyjre και akui αντίστοιχα. Φαίνεται να προσπαθεί να ταυτοποιηθεί στο μηχάνημα ελέγχου. Επίσης εμφανίζεται και μια επιπλέον πληροφορία το TIBicP2p η οποία δεν μας λέει κάτι. Η πραγματοποίηση της σύνδεσης επιβεβαιώνεται και μέσω του ActivePorts (εικόνα11).



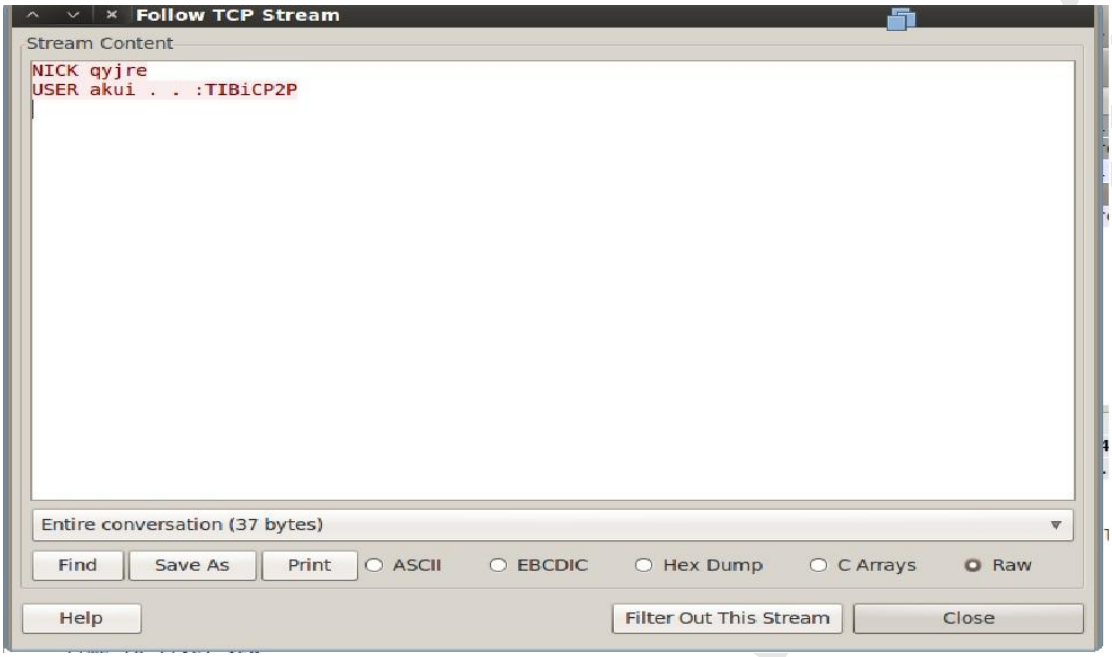
Εικόνα 11. Πραγματοποίηση Σύνδεσης όπως φαίνεται στο ActivePorts



Εικόνα 12 Εγκαθίδρυση σύνδεσης στο WireShark

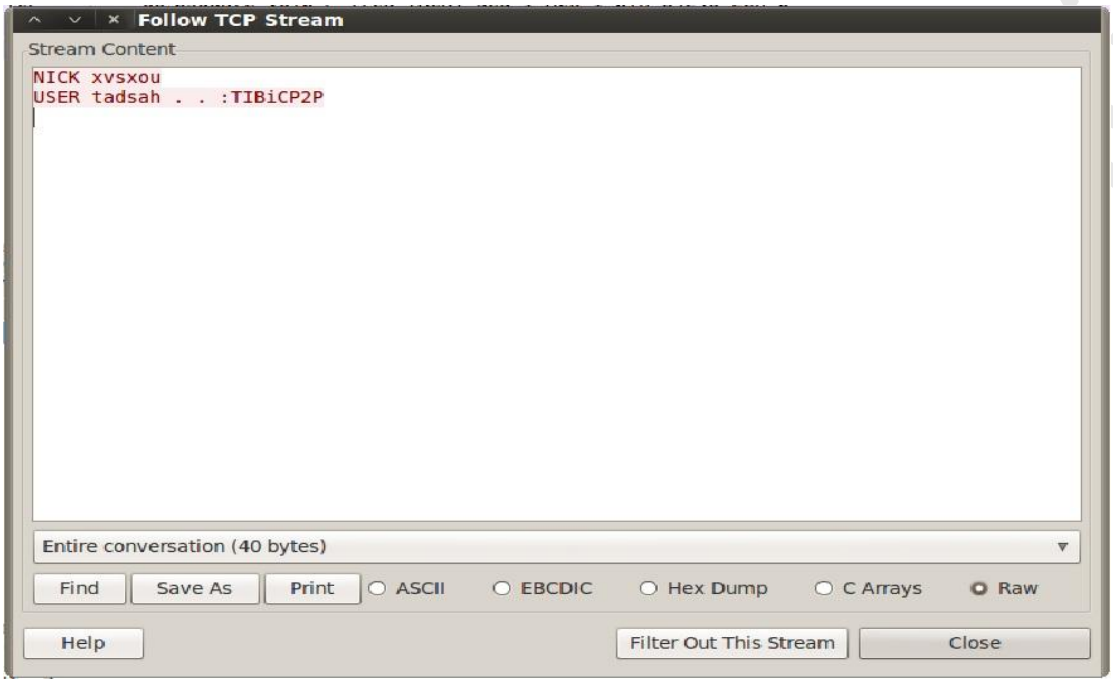
Ας παρατηρήσουμε και το wireshark το οποίο συνέλαβε και αυτήν την ροή δεδομένων (*εικόνα 12*). Παρατηρούμε πως το μηχάνημα ελέγχου έστειλε ένα SYN πακέτο του ήρθε απάντηση SYN,ACK και τέλος απάντησε με ένα πακέτα ACK. Πρόκειται για μια χειραψία τριών βημάτων (three way handshake) που σημαίνει ότι έχει εγκαθιδρυθεί μια σύνδεση. Ακριβώς από κάτω βλέπουμε πως μέσω του IRC πρωτοκόλλου πραγματοποιείται μεταφορά δεδομένων από το μηχάνημα ανάλυσης στο μηχάνημα ελέγχου. Τα δεδομένα που εστάλησαν (*εικόνα 13*) είναι αυτά που είδαμε προηγουμένως στο terminal του μηχανήματος ελέγχου.

Λόγω του είδους του πρωτοκόλλου μέσω του οποίου έγινε η ανταλλαγή δεδομένων μας οδηγεί στο συμπέρασμα πως το TIBiCP2P ίσως πρόκειται για κάποιο ιδιωτικό κανάλι IRC server στο οποίο συνδέεται το μηχάνημα ανάλυσης! Από εκεί και πέρα δεν είμαστε σε θέση να ελέγξουμε τι μπορεί να κάνει ο επιτιθέμενος. Θα προσπαθήσουμε να το ανιχνεύσουμε με άλλες μεθόδους.



Εικόνα 13. Δεδομένα που εστάλησαν μέσω IRC πρωτοκόλλου

Αξίζει να σημειωθεί πως επαναλαμβάνοντας το πείραμα το μηχάνημα ανάλυσης έστειλε στο μηχάνημα ελέγχου πάλι μέσω IRC πρωτοκόλλου διαφορετικό όνομα χρήστη και κωδικό ενώ το TIBiCP2P παρέμεινε το ίδιο (*εικόνα 14*). Προφανώς ο κώδικας του malware ενσωματώνει και μια γεννήτρια τυχαίων ονομάτων χρήστη και κωδικών τους οποίους το μολυσμένο μηχάνημα χρησιμοποιεί για να ταυτοποιηθεί στο IRC κανάλι TIBiCP2P (υπόθεση).



Εικόνα 14. Δεδομένα που εστάλησαν μέσω IRC πρωτοκόλλου (2η φορά)

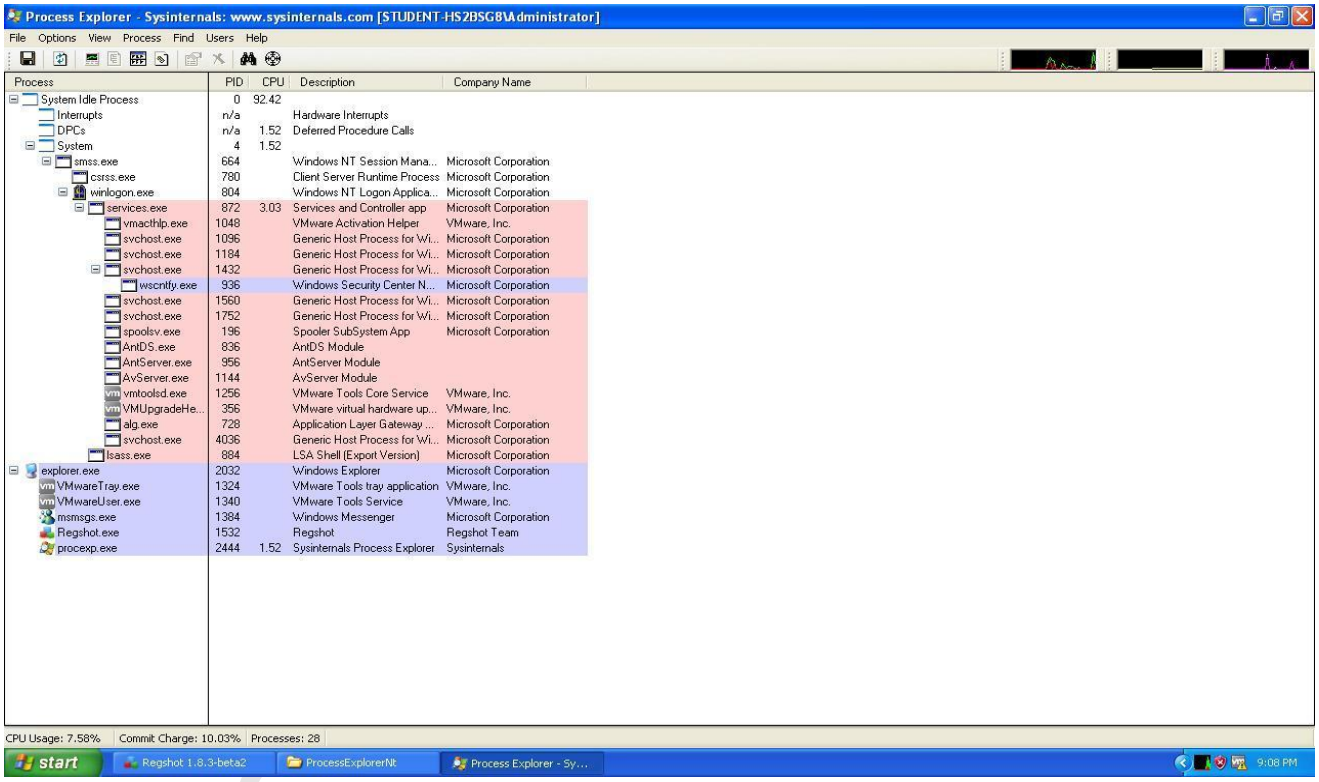
4.2 Μελέτη Συστήματος κατά την Εκτέλεση του Malware

Θα συνεχίσουμε με την δυναμική ανάλυση για την παρακολούθηση της αρχικής εκτέλεσης και διαδικασίας του malware. Θα πρέπει να συλλέξουμε πληροφορίες για τα αρχεία που δημιουργεί, που τροποποιεί και διαβάζει, καθώς και να δούμε αν μπορεί να προσπελάσει την ασφάλεια του firewall. Για την συγκεκριμένη δουλεία θα χρησιμοποιήσουμε το εργαλείο *Process Monitor (ProcMon) v.2.96,* το οποίο είναι ένα εξελιγμένο εργαλείο παρακολούθησης για το περιβάλλον των Windows που δείχνει σε πραγματικό χρόνο το System Directory, την Registry και την κακόβουλη δραστηριότητα. Συνδυάζει τα χαρακτηριστικά των δύο βοηθητικών προγραμμάτων της εταιρείας Sysinternals, του FileMon και του RegMon. Άλλο ένα πρόγραμμα από την Sysinternals που θα με βοηθήσει να δω πληροφορίες σχετικά με το ποιες διευθύνσεις και ποιες διαδικασίες DLL έχουν φορτωθεί ή ανοίξει στο σύστημα είναι το *Process Explorer v15.04*.

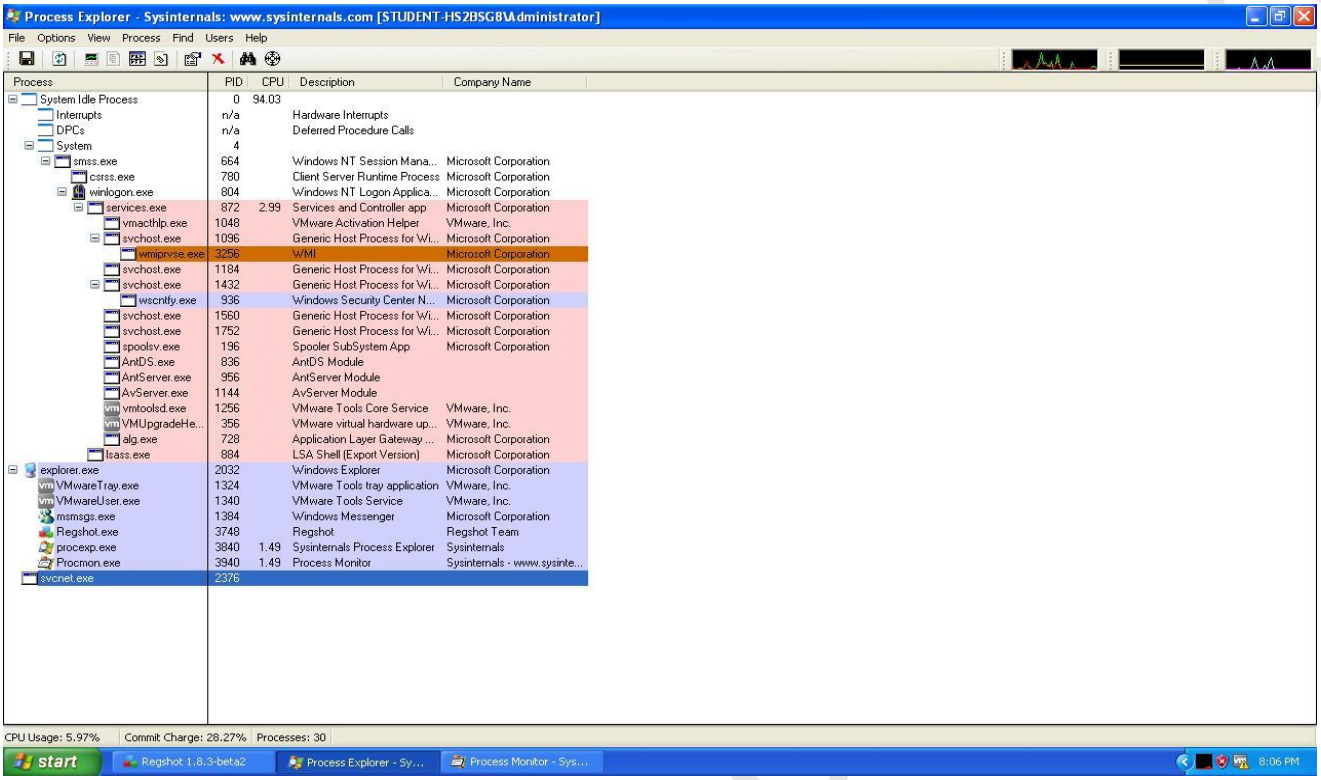
Επίσης θα γίνει και χρήση του προγράμματος *Regshot 1.8.3*, το οποίο είναι ένα βοηθητικό open-source πρόγραμμα για σύγκριση της registry και το οποίο επιτρέπει να

πάρουμε γρήγορα ένα στιγμιότυπο από την registry και στην συνέχεια αφού τρέξουμε το malware, βγάζοντας ένα δεύτερο στιγμιότυπο, να κάνουμε την σύγκριση.
Τέλος άλλο ένα πρόγραμμα που είναι χρήσιμο για την ανάλυση είναι το *Autoruns v.10.07* για Windows. Αυτό θα μας δείξει ποια προγράμματα είναι ρυθμισμένα να τρέχουν κατά την εκκίνηση του συστήματος ή της σύνδεσης, και επίσης παρουσιάζει τις καταχωρήσεις με τη σειρά όπως τις επεξεργάζονται τα Windows.

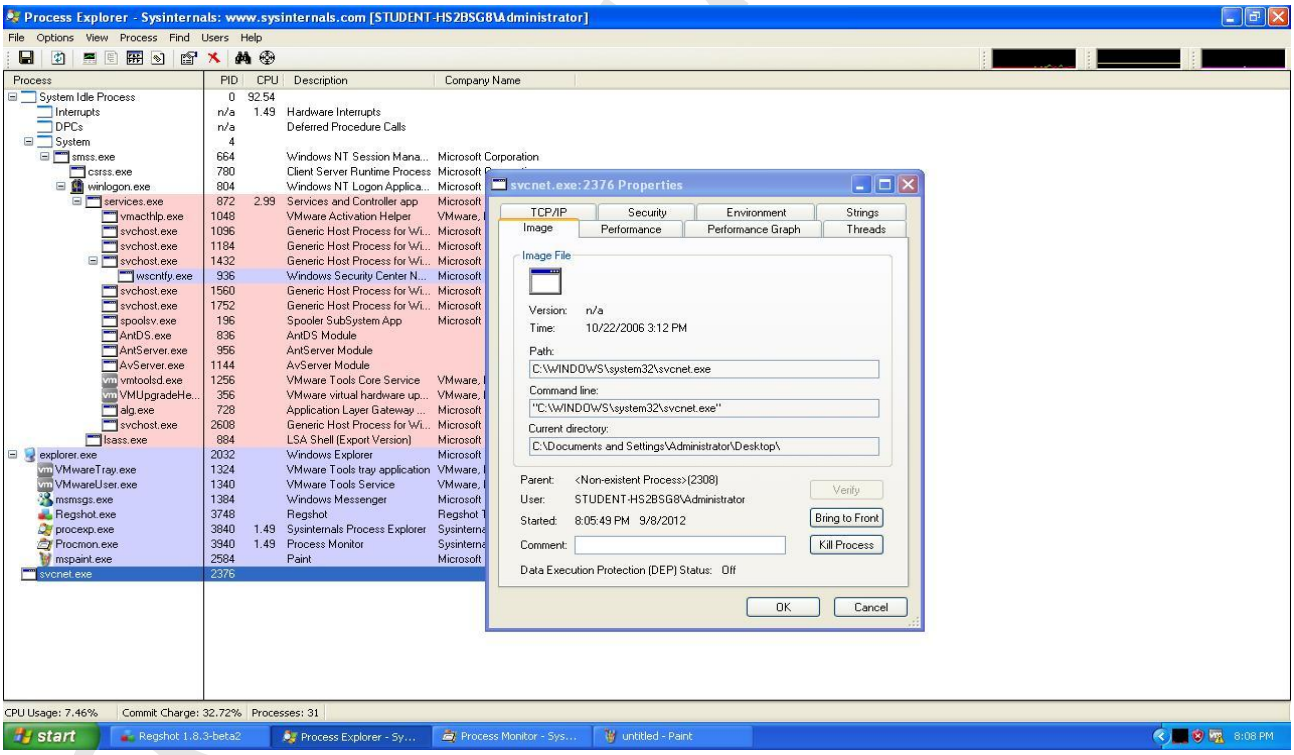
Αφού κάνουμε εξαγωγή του zip αρχείου που περιέχει το malware και έχουμε προετοιμάσει κατάλληλα τα παραπάνω προγράμματα, επιχειρούμε να τρέξουμε το malware. Όπως φαίνεται και στην παρακάτω (εικόνα 15,16 και 17) από το process explorer, βλέπουμε ότι δημιουργείται κάτω από το σύστημα αρχείων ένα καινούριο αρχείο, με την ονομασία *svcnet.exe.*



Εικόνα 15: Το Process Explorer σε κατάσταση πριν την εκτέλεση του malware

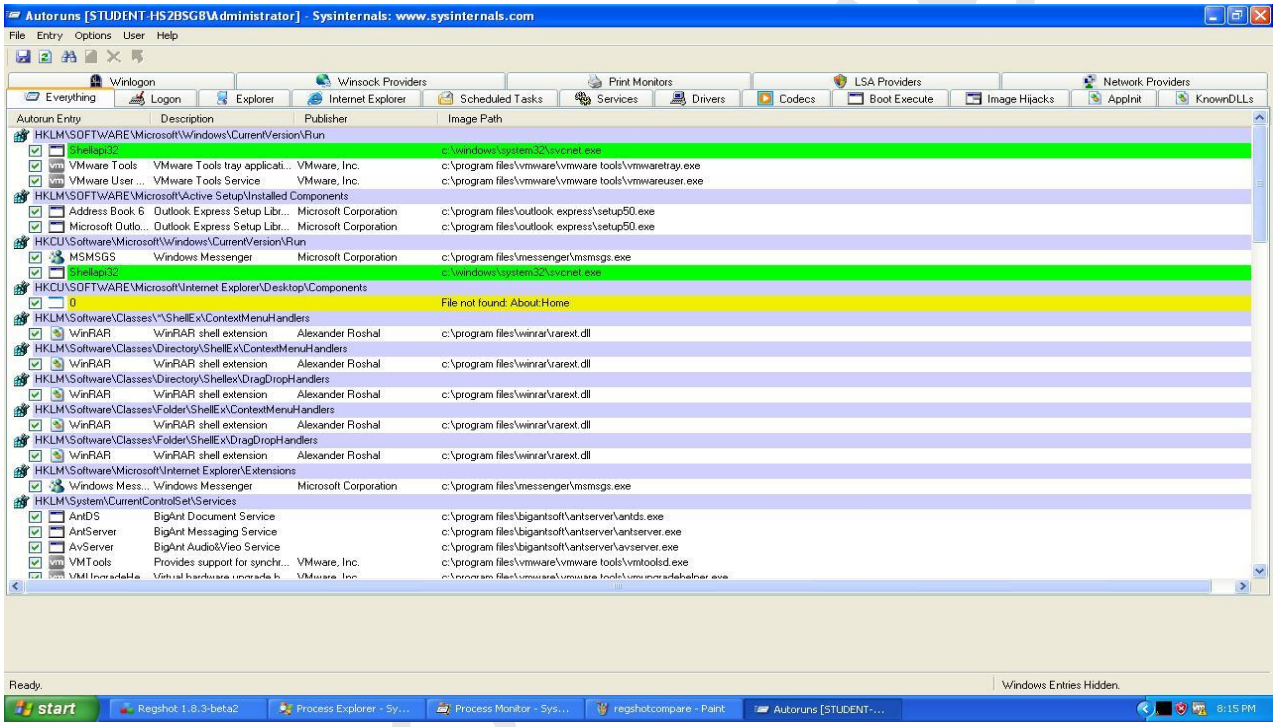


Εικόνα 16: Το Process Explorer σε κατάσταση μετά την εκτέλεση του malware

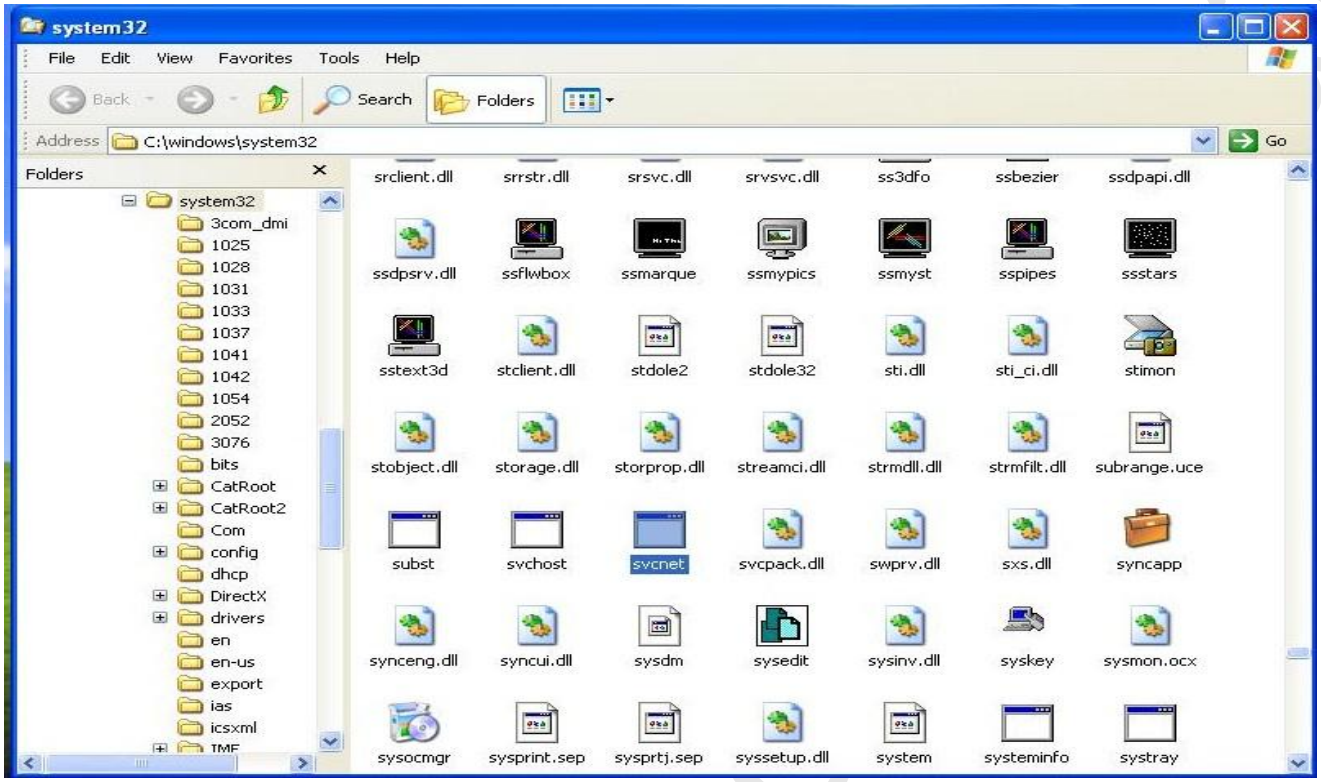


Εικόνα 17: Το Process Explorer σε κατάσταση μετά την εκτέλεση του malware

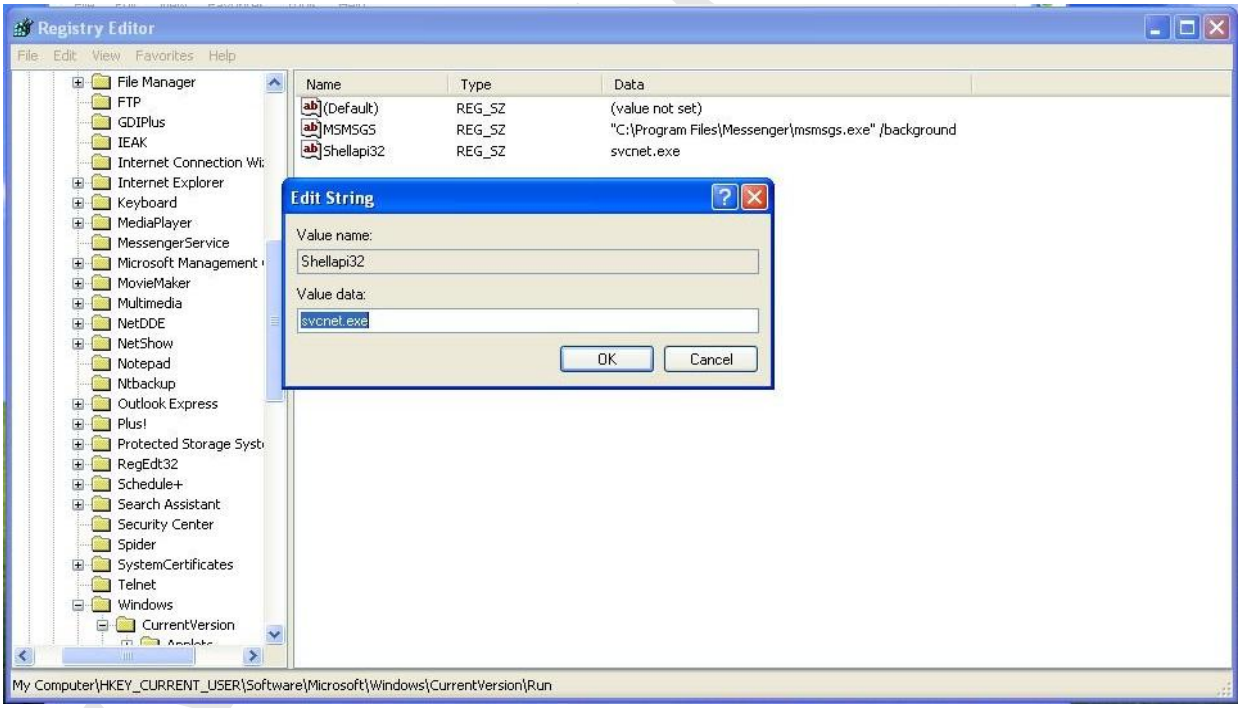
Παρακολουθώντας επίσης το πρόγραμμα Autoruns, παρατηρούμε ότι μόλις εκτελέστηκε το αρχείο malware, έγινε εγγραφή μιας νέας διαδικασίας που δεν εμφανιζόταν πριν στο σύστημα με την ονομασία *Shellapi32* της οποίας το path είναι *c:\windows\system32\scvnet.exe*. Καταλαβαίνουμε δηλαδή ότι έχουμε να κάνουμε με μια κακόβουλη διαδικασία από την στιγμή που σχετίζεται με το ύποπτο εκτελέσιμο αρχείο svcnet.exe. Επίσης με αυτή την διαδικασία το malware είναι ρυθμισμένο να τρέχει κατά την εκκίνηση του συστήματος, όπως φαίνεται και παρακάτω στην εικόνα 18,19 και 20.



Εικόνα 18: Εγγραφή νέας ύποπτης διαδικασίας με την ονομασία Shellapi32 στο πρόγραμμα autoruns



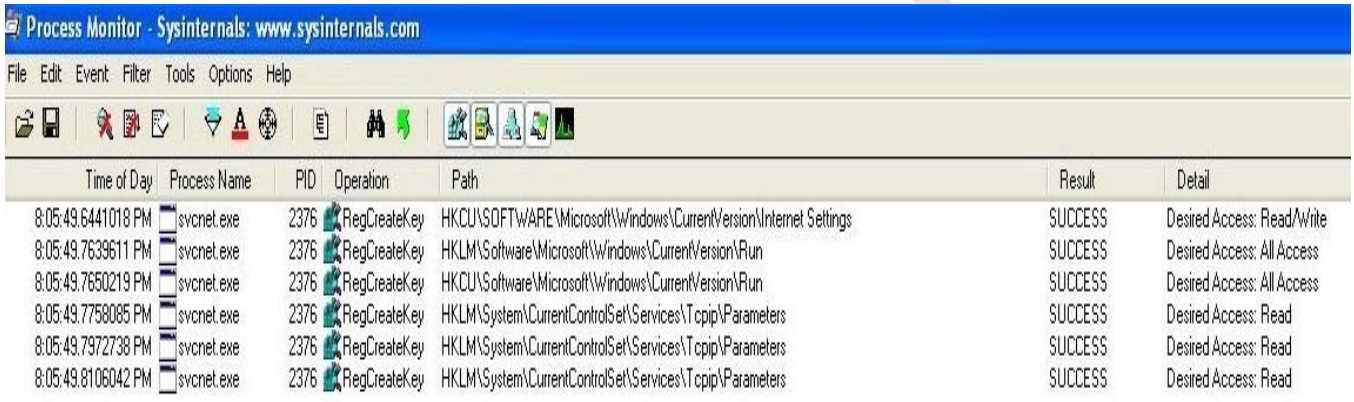
Εικόνα 19: Το κακόβουλο εκτελέσιμο αρχείο svcnet.exe καταχωρημένο μέσα στο system32.



Εικόνα 20: Η διαδικασία Shellapi32 καταχωρημένη στην Registry

Το windows firewall προσπάθησε να μπλοκάρει το malware από το να συνδεθεί στο δίκτυο, προειδοποιώντας με το χαρακτηριστικό μήνυμα. Επιλέξαμε εσκεμμένα να μην μπλοκάρει το αρχείο ώστε να συνεχίσουμε την διαδικασία ανάλυσης και να μελετήσουμε περεταίρω το κακόβουλο αυτό πρόγραμμα.

Επόμενο βήμα στην δυναμική ανάλυση είναι να ελέγξουμε το τι αλλαγές έχουν γίνει στην registry καθώς και ποια κλειδιά έχουν προστεθεί από την συγκεκριμένη διαδικασία. Χρησιμοποιώντας το πρόγραμμα Process Monitor (ProcMon) καταφέραμε να δούμε ότι από την εκτέλεση του αρχείου, δημιουργήθηκαν στην registry κάποια κλειδιά τα οποία φαίνονται στην επόμενη εικόνα (21).



Εικόνα 21: Δημιουργία κλειδιών στην Registry από το κακόβουλο αρχείο svcnet.exe

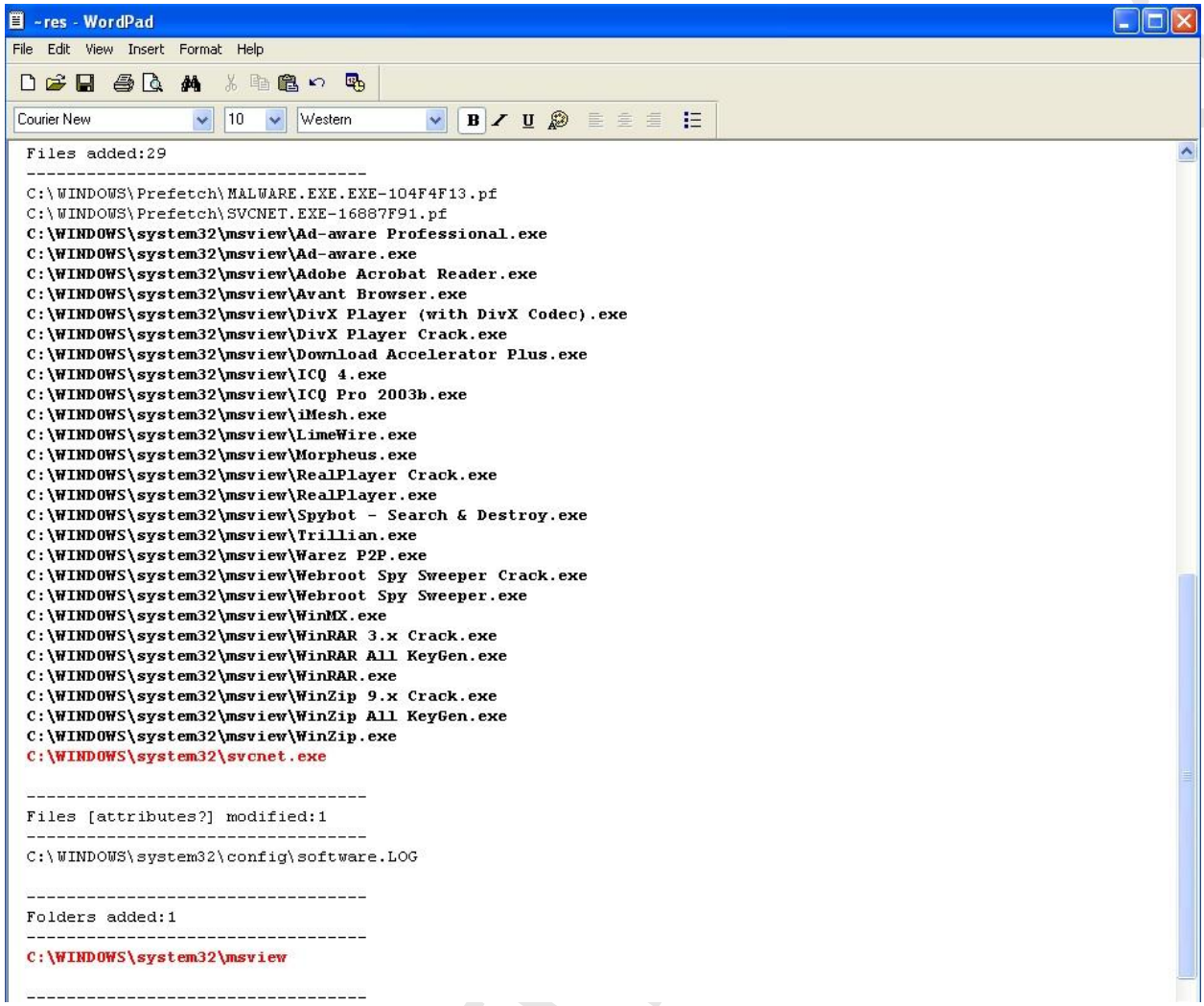
Από την παραπάνω εικόνα συμπεραίνουμε ότι τα τρία αυτά κλειδιά που έχουν εγγραφεί στην Registry έχουν ως σκοπό να μπορούν να διαβάζουν ώστε να αποκτήσουν τις θέσεις διαφόρων λειτουργιών που βρίσκονται στο σύστημα.

Συγκεκριμένα, με το πρώτο κλειδί που δημιουργείται στην registry, το *HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings* μπορεί ο επιτιθέμενος να αλλάξει τις ρυθμίσεις πρόσβασης στο διαδίκτυο και να εξουδετερώσει όλες τις εφαρμογές που στηρίζουν τις πληροφορίες τους στον Microsoft Proxy Server, όπως ο Internet Explorer, Microsoft office και άλλα.

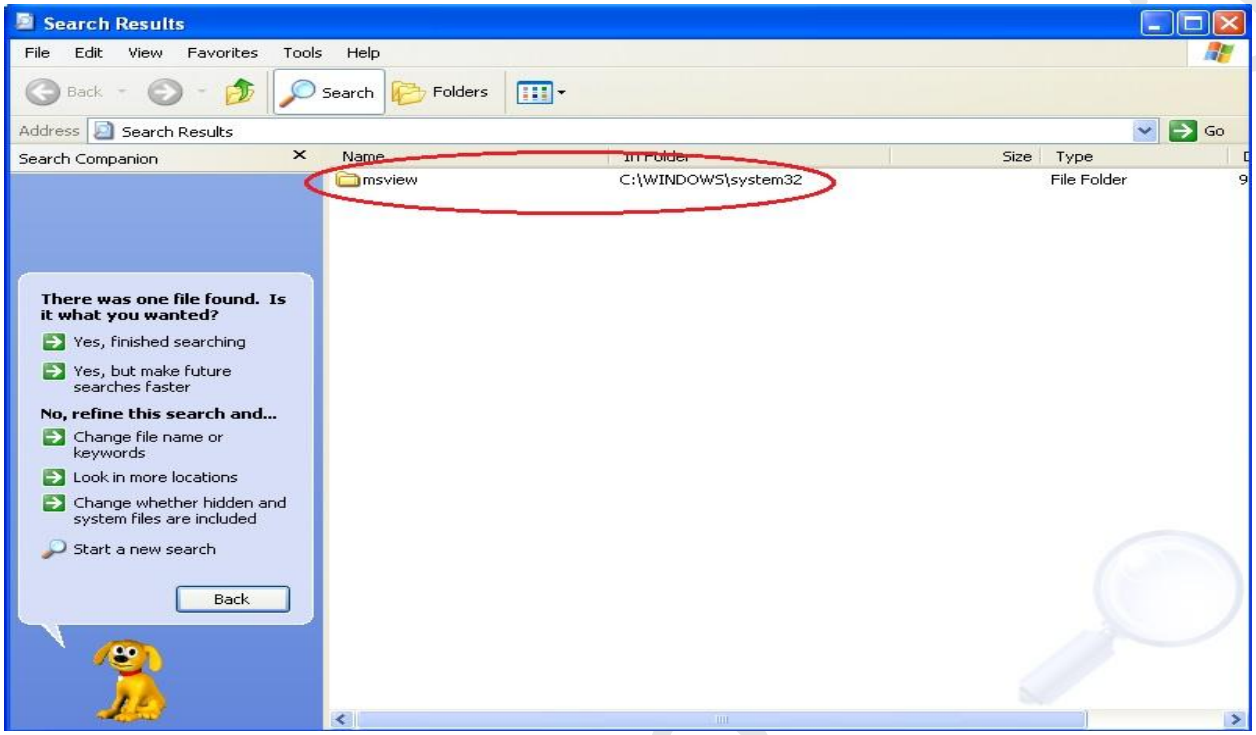
Τα δύο επόμενα κλειδιά που δημιουργούνται, με path *HKLM\Software\Microsoft\Windows\CurrentVersion\Run* και *HKCU\Software\Microsoft\Windows\CurrentVersion\Run* είναι ίσως από τα πιο σημαντικά κλειδιά που εγγράφονται στην registry και σκοπός τους είναι η αυτόματη εκκίνηση του malware όταν το λειτουργικό σύστημα των windows ξεκινήσει.

Τέλος, το τρίτο κλειδί που έχει προστεθεί στην registry με path **HKLM\SystemCurrentControlSet\Tcpip\Parameters** προσπαθεί να αποθηκεύσει ορισμένα δεδομένα, τα οποία επηρεάζουν κατά κύριο λόγο τις ρυθμίσεις του πρωτοκόλλου TCP/IP. Επειδή στο σύστημα υπάρχει μία λειτουργία που καθορίζει ποιές ενεργές συνδέσεις TCP υποστηρίζονται, με την εγγραφή αυτού του κλειδιού στην ουσία «νομιμοποιείται» η παρουσία του malware στο δίκτυο.

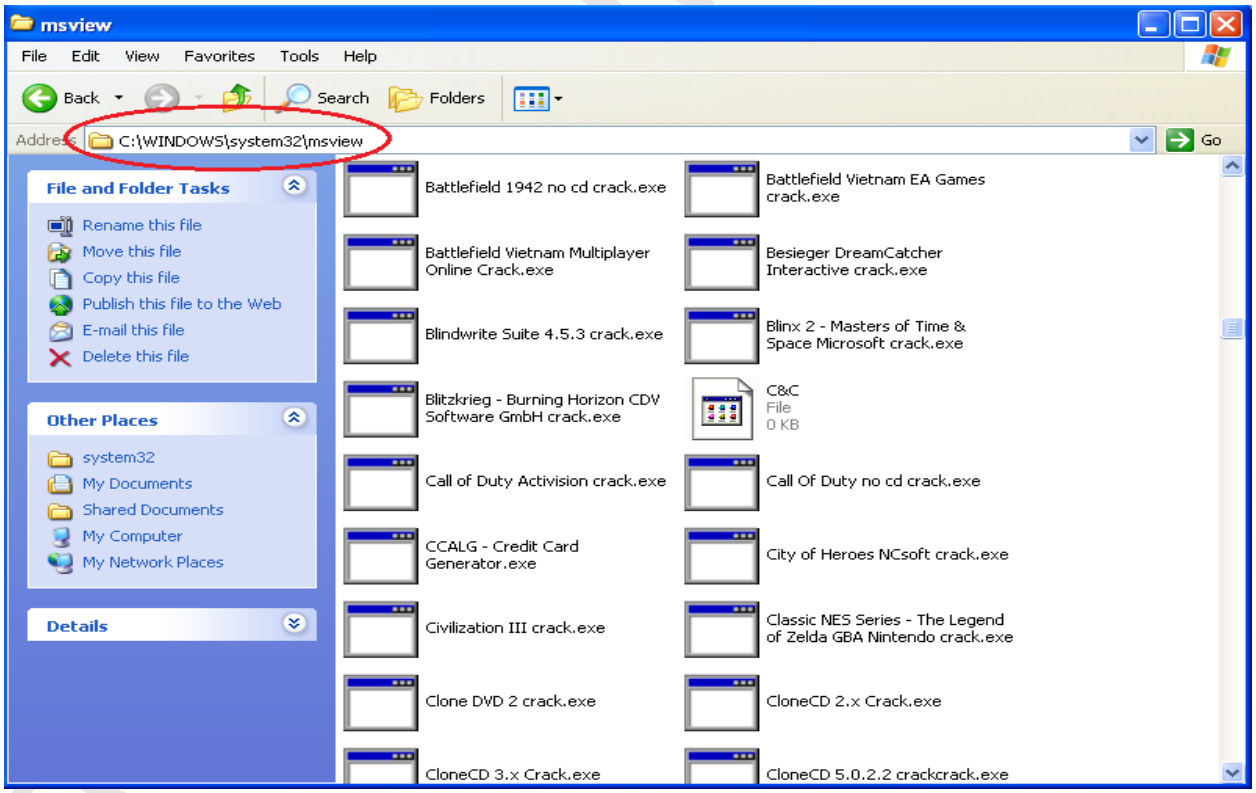
Στην συνέχεια, με την χρήση του προγράμματος regshot, του οποίου την λειτουργία αναφέραμε παραπάνω, καταφέραμε να δούμε την δημιουργία φακέλων που έχουν να κάνουν αποκλειστικά με το κακόβουλο λογισμικό μέσα στο System Directory του λειτουργικού συστήματος. Όλη η κίνηση με την δημιουργία φακέλων μετά την εκτέλεση του malware φαίνεται στην παρακάτω εικόνα. Παρατηρούμε πως οι βασικές αλλαγές που έγιναν στο λειτουργικό σύστημα πρίν και μετά την εκτέλεση του malware είναι η εισαγωγή κλειδιών στην registry (που εξετάσαμε και παραπάνω), η δημιουργία του αρχείου svcnet.exe μέσα στο System Directory, η δημιουργία του φακέλο msview μέσα στο System Directory και τέλος ένα πλήθος αρχείων που σχετίζεται με Crack και Serial Numbers γνωστών προϊόντων λογισμικού τα οποία είναι τοποθετημένα μέσα στον φάκελο msview (εικόνα 22). Στις εικόνες 23 και 24 βλέπουμε τον φάκελο αυτόν και τα περιεχόμενα του.



Εικόνα 22: Δημιουργία του φακέλου svcnet.exe στο system32 των windows.



Εικόνα 23. Αποτέλεσμα Αναζήτησης φακέλου με όνομα msview



Εικόνα 24. Περιεχόμενα του φακέλου msview

Επίλογος – Συμπεράσματα

Η διαδικασία που ακολουθήσαμε κατά την ανάλυση του malware, είναι η βασική γραμμή η οποία ακολουθείται κατά την ανάλυση ενός κακόβουλου λογισμικού. Φυσικά ανάλογα με την “φύση” του κακόβουλου λογισμικού μπορεί να μην είναι απαραίτητες όλες οι ενέργειες που ακολουθήθηκαν στην παρούσα εργασία ή να χρειάζονται επιπλέον τεχνικές ανάλυσης, οι βασικές αρχές πάντως είναι αυτές οι οποίες παρουσιάστηκαν. Μια δυναμική ανάλυση σε συνδυασμό με μια προσεκτική στατική ανάλυση η οποία συμπεριλαμβάνει και την εξέταση του κώδικα assembly του λογισμικού, δίνει σίγουρα πολύ σημαντικές πληροφορίες στον αναλυτή.

Όσο αφορά την περίπτωση μας, το κακόβουλο λογισμικό το οποίο αναλύσαμε είναι ένα worm. Η ανάλυση έδειξε πως αντιγράφεται στο System32 με το όνομα svcnet.exe το οποίο είναι και το όνομα της διεργασίας και δημιουργεί εγγραφές στην Registry ώστε να τρέχει αυτόματα με την έναρξη του λειτουργικού συστήματος και να προβαίνει σε λειτουργίες σχετικά με το διαδίκτυο. Συνδέεται σε έναν irc-server, και φαίνεται πως εισέρχεται (join) σε συγκεκριμένο κανάλι. Δημιουργεί έναν φάκελο με το όνομα msview μέσα στο System32 στον οποίον τοποθετεί πιθανώς τον εαυτό του με διάφορα ονόματα τα οποία σχετίζονται με cracks και serial numbers γνωστών εφαρμογών μάλλον για να προβεί σε ενέργειες ώστε να συμπεριλάβουν τον φάκελο αυτό στους διαμοιραζόμενους φακέλους τους P2P προγράμματα.

ΙΣΤΟΣΕΛΙΔΕΣ ΕΡΓΑΛΕΙΩΝ

[1] UPX, *http://upx.sourceforge.net/*

[2] TrIDNet, *http://mark0.net/soft-tridnet-e.html*

[3] BinText *http://www.mcafee.com/us/downloads/free-tools/bintext.aspx*

[4] Dependency Walker *http://www.dependencywalker.com/*

[5] FakeDNS *http://labs.idefense.com/software/malcode.php*

[6] CaptureBAT *https://www.honeynet.org/node/315*

[7] Active Ports *http://www.devicelock.com/freeware.html*

[8] Wireshark *http://www.wireshark.org/*

[9] Netcat *http://nc110.sourceforge.net/*

[10] VMWare Player *http://www.vmware.com/products/player/*

[11] Process Eplorer *http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx*

[12] RegShot *http://sourceforge.net/projects/regshot/*

[13] ProcessMonitor *http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx*

[14] Autoruns *http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx*

[15] PEID *http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml*

[16] IDA Pro *http://www.hex-rays.com/products/ida/index.shtml*

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Michael Sikorski, Andrew Honig, "*Practical Malware Analysis,Feb 2012* "

[2] Lenny Zaltser, "*Introduction to Malware Analysis*", 2010

[3] Eldad Eilam, Elliot Chikofsky, "*Reversing: Secrets of Reverse Engineering, 2005*"

[4] Craig Valli, Murray Brand, "*The Malware Analysis Body of Knowledge*", 2008

[5] Anders Orsten Flaglien, "Cross-Computer Malware Detection in Digital Forensics", Master's Thesis in Information Security 30 ECTS, 2010

[6] Mohammad Nour Saffaf, "Malware Analysis", Barchelor's Thesis, Helsinki Metropolia Univercity of Applied Scienes Degree Programme in Information Technology, Μάιος 2005

[7] Dennis Distler, "Malware Analysis: An Introduction", SANS Institute InfoSec Reading Room, December 2007

[8] Martin Overtoon, "Malware Forensics: Detecting the Unkown", presented at 2008 Virus Bulletin conference, Ottawa Canada, October 2008

[9] Harlan Carvey, Eoghan Casey, "Windows Forensic Analysis 2nd E" , 2009

[10] Michael Hale Ligh, Steven Adair, Blake Harstein, Matthew Richard "Malware Analyst's CookBook: Tools and Techniques for Fighting Malicious Code", 2011

[11] Michael Erbschloe, "Trojans Worms and Spyware", 2005

[12] Michael A. Davis, Sean M. Bodmer, Aaron LeMasters, "Malware & Rootkits Secrets & Solutions", 2010

[13] Nicholas Weaver, Vern Paxson, Stuart Stanifor, Robert Cunningham "A Taxonomy of Computer Worms" October 2003

[15] Dorn Seeley “Tour of the Worm” Department of Computer Science Univercity of Utah,