Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Ανάλυση Κακόβουλου Λογισμικού που Σχετίζεται με Απάτες Συναλλαγών

Δημήτρης Κουτρώτσιος

Επιβλέπων Καθηγητής
Ιωάννης Μαυρίδης

Μάιος 2017

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Ανάλυση Κακόβουλου Λογισμικού που Σχετίζεται με Απάτες Συναλλαγών

Δημήτρης Κουτρώτσιος

Επιβλέπων Καθηγητής
Ιωάννης Μαυρίδης

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
Στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου.

Μάιος 2017

Περίληψη

Σκοπός της παρούσας διατριβής είναι η μελέτη και ανάλυση των κακόβουλων λογισμικών τα οποία χρησιμοποιούνται για τη διενέργεια απατών που αφορούν ηλεκτρονικές συναλλαγές. Για τον σχηματισμό καλύτερης εικόνας σχετικά με τα εν λόγω κακόβουλα λογισμικά παρουσιάζεται αρχικά μια ανασκόπηση σχετικά με την έννοια του κακόβουλου λογισμικού και τις επιπτώσεις της χρήσης του σε απάτες συναλλαγών. Για σκοπούς της πληρέστερης μελέτης των συγκεκριμένων κακόβουλων λογισμικών παρουσιάζεται η τρέχουσα κατάσταση, σε ότι αφορά την ονοματοδοσία και ταξινομία τους, ενώ παράλληλα γίνεται αναφορά στην ανάλυση κακόβουλου λογισμικού ως μέθοδο μελέτης τους. Στην συνέχεια, παρουσιάζονται αναλύσεις μερικών εκ των πλέον επικίνδυνων κακόβουλων λογισμικών τα οποία χρησιμοποιούνται σε απάτες συναλλαγών. Επιπλέον, προτείνεται και εφαρμόζεται μια νέα μέθοδος ταξινόμησής τους. Τέλος, προτείνονται μέτρα αποτελεσματικότερης αντιμετώπισης του παγκόσμιου φαινομένου διενέργειας απάτης συναλλαγών μέσω κακόβουλου λογισμικού.

Summary

The main objective of this dissertation is to study and analyze malicious software used to commit frauds in electronic transactions. In order to get a better understanding of such malicious software, a review of the concept of malware and the impact of its use on committing electronic transaction frauds is initially presented. For a better studying, the current situation of such malware's naming and taxonomy is presented. Moreover, malware analysis as a method of studying is discussed. Next, an analysis of some of the most dangerous malicious software used in transaction frauds is presented. Furthermore, a new classification method is proposed and applied to the above mentioned malware. Finally, a number of controls are proposed to effectively address issues related with the global phenomenon of utilizing malware in committing fraud in electronic transactions.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Δρ. Ι. Μαυρίδη και τον κ. Γ. Σακελλαρίου για την αμέριστη συμπαράσταση και υποστήριξη κατά την διάρκεια της εκπόνησης της παρούσας διατριβής, όπως επίσης και την οικογένεια μου για την στήριξη της. Τέλος θα ήθελα να ευχαριστήσω τους συναδέλφους μου οι οποίοι επίσης συνέβαλλαν καθοριστικά με τις γνώσεις και τις συμβουλές τους.

Περιεχόμενα

Ευρετήριο Εικόνων και Πινάκων

Εικόνες

Πίνακες

Κεφάλαιο 1

Εισαγωγή

Οι απάτες συναλλαγών είναι μια νέα μοντέρνα μορφή εγκληματικότητας η οποία αναπτύχθηκε παράλληλα με το διαδίκτυο και την ενσωμάτωσή του σε αυτό των καθημερινών εμπορικών ή μη συναλλαγών μας. Λόγω της διαρκούς ανάπτυξης της τεχνολογίας και της αδυναμίας των καταναλωτών, αλλά πολλές φορές και των κρατών ή των επιχειρήσεων να την ακολουθήσουν, αποτελεί έναν σχετικά εύκολο τρόπο εγκληματικής δράσης, ο οποίος μάλιστα έχει το πλεονέκτημα του παγκόσμιου βελησνεκούς.

Ένας εγκληματίας με γνώσεις τεχνολογιών πληροφορίας και επικοινωνιών (ΤΠΕ), έχει τη δυνατότητα να διαπράττει απάτες συναλλαγών πρακτικά οπουδήποτε στον πλανήτη και μάλιστα, το ίδιο σχετικά εύκολα, μπορεί να διατηρήσει την ανωνυμία του. Αυτό καθιστά της απάτες συναλλαγών έναν ιδιαίτερα θελκτικό τρόπο εγκληματικής δράσης ο οποίος μάλιστα δεν αφορά μόνο τους κατ' επάγγελμα εγκληματίες αλλά οποιονδήποτε έχει τις γνώσεις ώστε να διαπράξει μια τέτοια απάτη με σκοπό το πρόσκαιρο χρηματικό κέρδος.

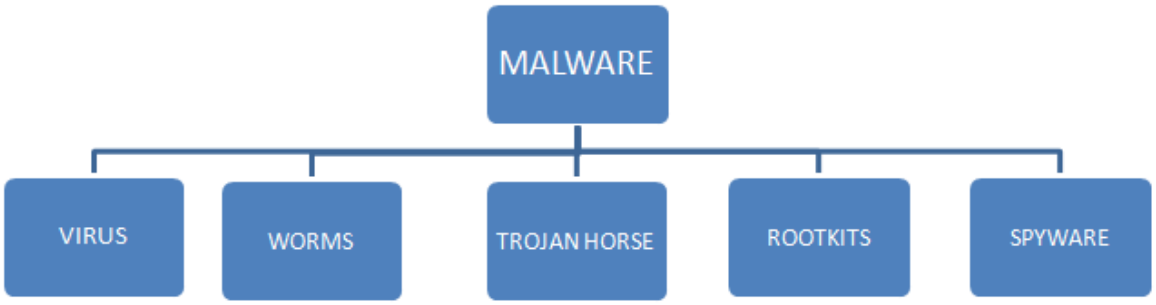
Σύμφωνα με την τελευταία ετήσια αναφορά της Nilson για το άμεσο κόστος λόγω απάτης σε συναλλαγές μόνο στο σύστημα των τραπεζικών καρτών παγκοσμίως ανήλθε στα 16,3 δις δολάρια ενώ έως το 2020 αναμένεται να εκτοξευτεί στα 35 δις δολάρια [1], χωρίς δε να συνυπολογίζεται το έμμεσο κόστος το οποίο αφορά τα έξοδα για ανίχνευση, παρακολούθηση και αποτροπή απατών τέτοιου είδους. Εάν στο παραπάνω κόστος, το οποίο αφορά μόνο ένα τραπεζικό σύστημα και μόνο το άμεσο κόστος, προσθέσουμε τα κόστη για τα υπόλοιπα είδη απάτης συναλλαγών και τα έξοδα που συνεπάγεται η αντιμετώπισή τους σε παγκόσμιο επίπεδο, τότε καθίσταται επιτακτική η ανάγκη της ευρέσεως μια μεθοδολογίας αντιμετώπισης του προβλήματος αυτού, η οποία όχι μόνο θα βοηθάει στην αποτροπή τέτοιου είδους απατών, αλλά θα θέτει τις απαραίτητες αρχές και

βάσεις σε ότι αφορά τις μεθόδους και τις διαδικασίες οι οποίες θα πρέπει να τηρούνται από κάθε οργανισμό ή ιδιώτη.

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής που ακολουθεί θα προσπαθήσουμε να συμβάλουμε προς αυτή την κατεύθυνση, μελετώντας συγκεκριμένα, τις απάτες συναλλαγών στις οποίες χρησιμοποιείται κακόβουλο λογισμικό ή αλλιώς malware και την δημιουργία ενός μοντέλου αναγνώρισης και κατάταξης των διαφόρων malware.

Το malware είναι κακόβουλος προγραμματιστικός κώδικας ή λογισμικό το οποίο εισέρχεται σε ένα σύστημα, συνήθως με μη εμφανή τρόπο με σκοπό να απειλήσει την εμπιστευτικότητα, την ακεραιότητα και την διαθεσιμότητα των δεδομένων, των εφαρμογών και του λειτουργικού συστήματος του θύματος ή απλά να το παρενοχλήσει. [2]

Υπάρχουν πολλών ειδών malwares όπως spyware, key loggers, viruses, worms και γενικά κάθε είδους λογισμικό το οποίο μπορεί να τοποθετηθεί σε έναν Η/Υ χωρίς να γίνει αντιληπτό με σκοπό την υποκλοπή πληροφοριών ή την καταστροφή του συστήματος. Γενικά οποιοδήποτε λογισμικό μπορεί να χαρακτηριστεί ως malware, περισσότερο ανάλογα με τους σκοπούς του δημιουργού του παρά ως προς τις δυνατότητες του σαν λογισμικό.



Εικόνα 1.1 Είδη Κακόβουλου Λογισμικού

Με τα παραπάνω γίνεται εύκολα αντιληπτό ότι τα malware μπορεί να συσχετιστούν με τις απάτες συναλλαγών και αποτελούν ίσως εν δυνάμει εργαλεία για την σχεδίαση και την εφαρμογή μιας τέτοιας απάτης. Οι δυνατότητες τις οποίες προσδίδει το malware είναι

μάλιστα τόσο μεγάλες, ώστε υπάρχει υπερπληθώρα διαφορετικών λογισμικών τα οποία εμπίπτουν στην κατηγορία και καθημερινά κατασκευάζονται νέα, πολλές φορές με την αρωγή εγκληματικών οργανώσεων αποκλειστικά και μόνο για χρήση σε απάτες συναλλαγών. [3]

Παρόλα αυτά, όμως, με δεδομένη την ύπαρξη συγκεκριμένων μεθόδων συναλλαγών αλλά και συγκεκριμένων δικλείδων ασφαλείας σε αυτές, τα malware δεν μπορεί παρά να είναι εστιασμένα στην υπερπήδηση αυτών των δικλείδων και ως εκ τούτου δεν μπορούν παρά να υπάρχουν πεπερασμένου αριθμού τύποι malware.

Παρακάτω, λοιπόν, οι διάφοροι τύποι malware, που σχετίζονται με απάτες συναλλαγών, θα μελετηθούν και θα κατηγοριοποιηθούν, ανάλογα με τον στόχο τους, τον τρόπο λειτουργίας τους αλλά και τις ιδιότητες τους.

Κεφάλαιο 2

Θεωρητική Θεμελίωση

2.1 Απάτες Συναλλαγών

Σύμφωνα με το National Fraud & Cyber Crime Reporting Centre της Μ. Βρετανίας, απάτη συναλλαγής ονομάζεται κάθε απάτη η οποία εμπεριέχει τη δημιουργία ή την εσφαλμένη προώθηση μιας πληρωμής.

Συγκεκριμένα μπορεί να περιέχει:

- την δημιουργία ψεύτικων αρχείων πελατών και τραπεζικών λογαριασμών για την προώθηση πληρωμών
- την υποκλοπή επιταγών ή διαταγών πληρωμής, την αλλαγή των στοιχείων τους και την προσπάθεια εξαργύρωσής τους
- την δημιουργία ψευδών οικονομικών στοιχείων και στοιχείων πληρωμής με σκοπό την διεκδίκηση επανορθώσεων
- την προώθηση ψευδών ισχυρισμών κατά συνεργών για την οικειοποίηση παροχών, επιδοτήσεων ή επιστροφών
- πληρωμές προς όφελος του ιδίου

Σύμφωνα με την IBM [4], απάτη συναλλαγής θεωρείται κάθε μορφή παρεμβολής με την βοήθεια της οποίας μεταφέρονται χρήματα από τον τραπεζικό λογαριασμό του θύματος στον τραπεζικό λογαριασμό του απατεώνα.

Από τα παραπάνω γίνεται εύκολα αντιληπτό ότι οι απάτες πληρωμών αφορούν ένα αρκετά ευρύ πεδίο εγκληματικών ενεργειών και αυτό σχετίζεται με τα πολλά είδη απάτης συναλλαγών που υπάρχουν.

2.2 Κατηγορίες Μεθόδων

Οι τρεις μεγάλες κατηγορίες μεθόδων οι οποίες χρησιμοποιούνται σε απάτες συναλλαγών είναι:

- Η **Κλοπή Ταυτότητας (Identity Theft)**, η οποία στη διαδικτυακή κοινότητα αποτελεί έναν διαδεδομένο τρόπο απάτης. Μέσω αυτής ένας εγκληματίας μπορεί να οικειοποιηθεί τα στοιχεία ταυτότητας κάποιου χρήστη και με την βοήθειά τους να προβεί σε οποιαδήποτε ηλεκτρονική συναλλαγή προς όφελος του ή να τα χρησιμοποιήσει σε επιθέσεις Phishing αποκτώντας σημαντικές πληροφορίες άλλων χρηστών. Επίσης, οι απάτες με τη μορφή επιθέσεων Man-In-The-Middle εμπίπτουν σε αυτή την κατηγορία όπως και οι απάτες συναλλαγών με την χρήση πλαστών επιταγών αλλά και η μέθοδος του Pagejacking.

- Η **Εξαπάτηση**, κατά την οποία ο απατεώνας δεν προσπαθεί να προσποιηθεί κάποιον άλλο, αλλά προσπαθεί να εξαπατήσει το θύμα ώστε να πραγματοποιήσει πληρωμές προς όφελος του. Αυτού του είδους οι απάτες συναλλαγών εμπεριέχουν τις μεθόδους των προκαταβολών και της ηλεκτρονικής μεταφοράς χρημάτων.

- Οι **Επιθέσεις σε Server**, οι οποίες αφορούν απάτες από την μεριά των πιστωτικών ιδρυμάτων και εμπεριέχουν συνήθως την δράση υπαλλήλων των τραπεζών ή την εκμετάλλευση σφαλμάτων στις μεθόδους και τον τρόπο λειτουργίας των τραπεζικών συστημάτων. [5]

Σε αυτές τις 3 μεγάλες κατηγορίες περιέχονται διάφορες μέθοδοι απάτης ηλεκτρονικών συναλλαγών, που εκμεταλλεύονται εκτενώς διάφορες τεχνικές κυβερνοεπιθέσεων, όπως:

- **Phishing,** με την οποία υποκλέπτονται προσωπικές πληροφορίες όπως αριθμοί τραπεζικών λογαριασμών, στοιχεία πιστωτικών καρτών και στοιχεία αυθεντικοποίησης μέσω πλαστών ιστοσελίδων ή emails, τα οποία ζητούν την υποβολή των παραπάνω στοιχείων προβάλλοντας συνήθως ψευδώς ότι προέρχονται από κάποιον καθόλα αξιόπιστο οργανισμό.

- **Pagejacking**, με την οποία οι εγκληματίες ανακατευθύνουν την διαδικτυακή κίνηση από κάποια ιστοσελίδα διαδικτυακών αγορών σε μια δική τους πανομοιότυπη ιστοσελίδα, με σκοπό την εγκατάσταση κακόβουλου λογισμικού και την δημιουργία ψευδών ηλεκτρονικών συναλλαγών.

- Αίτηση **προκαταβολών και ηλεκτρονικής μεταφοράς χρημάτων**, κατά την οποία οι εγκληματίες στοχεύουν τους ιδιοκτήτες πιστωτικών καρτών ή ηλεκτρονικών καταστημάτων ζητώντας προκαταβολές, σε αντάλλαγμα χρέωσης της πιστωτικής τους κάρτας ή υπόσχεσης ηλεκτρονικής μεταφοράς χρημάτων σε μεταγενέστερη ημερομηνία.

- **Man-In-The-Middle**, κατά την οποία ο εγκληματίας παρεμβάλλεται στην επικοινωνία μεταξύ 2 μερών με σκοπό την υποκλοπή στοιχείων ταυτότητας ή την ανακατεύθυνση της διαδικτυακής κίνησης προς όφελος του.

2.3 Κακόβουλο Λογισμικό

Κακόβουλο λογισμικό (Malware) ονομάζεται οποιοδήποτε λογισμικό εισέρχεται σε ένα πληροφοριακό σύστημα με σκοπό να το βλάψει ή να το χρησιμοποιήσει για διαφορετικό σκοπό από εκείνο για τον οποίο το προορίζουν οι ιδιοκτήτες του. [6] Το κακόβουλο λογισμικό μπορεί να αποκτήσει πρόσβαση σε ένα πληροφοριακό σύστημα, να καταγράψει

και να αποστείλει δεδομένα από αυτό το σύστημα σε κάποιον τρίτο χωρίς την άδεια ή την γνώση του χρήστη, να αποκρύψει ότι το συγκεκριμένο σύστημα έχει μολυνθεί με τέτοιου είδους λογισμικό, να απενεργοποιήσει τα μέτρα ασφαλείας του συστήματος, να προκαλέσει ζημιά στο σύστημα ή διαφορετικά να επηρεάσει την ακεραιότητα του συστήματος. Παρόλο που δεν αποτελεί το μόνο μέσο με το οποίο μπορεί να επηρεαστεί η ασφάλεια ενός πληροφοριακού συστήματος το κακόβουλο λογισμικό παρέχει στους επιτιθέμενους άνεση, ευκολία χρήσης και τους απαραίτητους αυτοματισμούς όπου αυτοί χρειάζονται, ώστε πλέον να είναι δυνατόν να πραγματοποιηθούν επιθέσεις σε κλίμακες τις οποίες ήταν αδύνατον να φανταστούμε μερικά χρόνια πριν.

Μερικές από τις ιδιαίτερες δυνατότητες του κακόβουλου λογισμικού είναι οι παρακάτω:

- Είναι **πολυλειτουργικό** και **σπονδυλωτό**, δίνοντας την δυνατότητα συνδυασμού διαφορετικών τύπων κακόβουλου λογισμικού για την επίτευξη του επιθυμητού αποτελέσματος. Μπορούν εύκολα να προστεθούν νέες δυνατότητες σε υπάρχον λογισμικό, με σκοπό να βελτιωθεί η αποτελεσματικότητα του αλλά και να αλλαχθεί ο τρόπος λειτουργίας του ακόμα και απομακρυσμένα.

- **Είναι εύκολα διαθέσιμο** και **φιλικό προς το χρήστη**. Το κακόβουλο λογισμικό είναι διαθέσιμο στον οποιονδήποτε στο διαδίκτυο έναντι πολύ μικρού κόστους σε αναλογία με την ζημιά την οποία μπορεί να προκαλέσει.

- **Είναι επίμονο και αποτελεσματικό**. Το κακόβουλο λογισμικό είναι ολοένα και πιο δύσκολο να ανιχνευθεί και να αφαιρεθεί από ένα πληροφοριακό σύστημα και είναι αποτελεσματικό στο να αναστέλλει όλα τα μέτρα προστασίας ενός πληροφοριακού συστήματος. Μερικά είδη δε είναι ικανά να υπερπηδούν ακόμα και αυθεντικοποίησεις πολλαπλών παραγόντων ή ψηφιακά πιστοποιητικά ασφαλείας.

- **Μπορεί να επηρεάσει μια ποικιλία συστημάτων.** Λόγω του ότι το κακόβουλο λογισμικό αποτελεί απλά ένα κομμάτι λογισμικού είναι εύκολο να διαμορφωθεί κατάλληλα, έτσι ώστε να είναι σε θέση να επηρεάσει μια ποικιλία συστημάτων, από

προσωπικές συσκευές επικοινωνίας όπως έξυπνα τηλέφωνα μέχρι servers σε διαφορετικά είδη δικτύων. Όλα αυτά τα συστήματα, συμπεριλαμβανομένων των routers, είναι ευάλωτα σε επιθέσεις κακόβουλου λογισμικού.

- **Είναι μέλος ενός ευρύτερου συστήματος κυβερνοεπιθέσεων.** Το κακόβουλο λογισμικό μπορεί να χρησιμοποιηθεί είτε σαν αυτόνομο μέσο επίθεσης είτε προς υποστήριξη μιας ευρύτερης επίθεσης σε πληροφοριακά συστήματα, όπως το spam και το phishing, τα οποία επίσης μπορούν με την σειρά τους να χρησιμοποιηθούν στην διασπορά κακόβουλου λογισμικού.

- **Είναι κερδοφόρο.** Το κακόβουλο λογισμικό δεν είναι απλά ένα παιχνίδι στα χέρια των προγραμματιστών, έχει περάσει πλέον σε επαγγελματικό επίπεδο και αποτελεί κύρια πηγή εσόδων για εγκληματίες σε ολόκληρο τον κόσμο. Μαζί με άλλες μορφές εργαλείων αποτελεί ένα φθηνό και επαναχρησιμοποιήσιμο εργαλείο για την διενέργεια ηλεκτρονικών εγκλημάτων.

2.3.1. Χαρακτηριστικά και Είδη

Το κακόβουλο λογισμικό μπορεί να χωριστεί σε κατηγορίες με βάση τα τρία (3) ακόλουθα χαρακτηριστικά [6]:

- **Αναπαραγωγή**, κατά την οποία το ίδιο το κακόβουλο λογισμικό έχει την δυνατότητα να αναπαράγει τον εαυτό του.

- **Αύξηση του πληθυσμού**, η οποία περιγράφει την αλλαγή του αριθμού των κακόβουλων λογισμικών που υπάρχουν σε ένα πληροφοριακό σύστημα. Ακόμη και ένα κακόβουλο λογισμικό που δεν φαίνεται να αναπαράγεται (έχει μηδενική αύξηση πληθυσμού) μπορεί να έχει την δυνατότητα να αναπαραχθεί.

- **Παρασιτισμός**, το κακόβουλο λογισμικό έχει την δυνατότητα να αποκρύπτει την παρουσία του μέσα σε άλλο λογισμικό και να ενεργοποιείται με την εκτέλεση του.

Τα είδη κακόβουλου λογισμικού ορίζονται κυρίως ως προς τον τρόπο λειτουργίας τους και είναι τα εξής [6]:

- **Logic Bomb**: Αποτελούνται από 2 μέρη, το payload το οποίο είναι το ενεργό μέρος το οποίο παράγει τα κακόβουλα αποτελέσματα και το trigger το οποίο είναι μία λογική συνθήκη υπό την οποία το payload εκτελείται π.χ. σε συγκεκριμένη ημερομηνία. Τα logic bomb μπορούν να είναι μέρος άλλου λογισμικού, να είναι δηλαδή παρασιτικά, ή αυτόνομα και δεν αυτοαναπαράγονται. Τα logic bomb μπορούν να χρησιμοποιηθούν σε περιπτώσεις εκδίκησης ή ακόμα και εκβιασμού καθώς η δυνατότητα τους να αποτελούν μέρος άλλου λογισμικού και να ενεργοποιούνται υπό συνθήκες τα καθιστά άκρως επικίνδυνα ιδιαίτερα για εταιρείες.

- **Trojan Horse**: Είναι προϊόντα λογισμικού τα οποία ενώ εμφανίζονται να εκτελούν χρήσιμες λειτουργίας, έχουν επιπλέον την δυνατότητα να εκτελέσουν και άλλες επιμελώς κρυμμένες κακόβουλες λειτουργίες. Οι λειτουργίες αυτές μπορούν να εκτείνονται από την συλλογή κωδικών πρόσβασης μέχρι την απόκτηση πλήρους πρόσβασης στο μολυσμένο πληροφοριακό σύστημα. Η δυνατότητα του να κρύβει κακόβουλες λειτουργίες ανάμεσα σε αθώες το καθιστά παρασιτικό, ενώ δεν αυτοαναπαράγεται.

- **Back Door**: Αποτελούν ένα μηχανισμό, ο οποίος παρακάμπτει τους συνηθισμένους ελέγχους ασφαλείας ενός συστήματος. Μπορούν να κρυφτούν σε αθώα λογισμικά, είναι δηλαδή παρασιτικά, ενώ επίσης δεν αναπαράγονται. Ένα είδος Back Door είναι τα RAT (Remote Administration Tools), τα οποία επιτρέπουν την απομακρυσμένη παρακολούθηση και έλεγχο ενός υπολογιστικού συστήματος.

- **Virus**: Πρόκειται για κακόβουλα προϊόντα λογισμικού που επιδιώκουν κατά την εκτέλεση τους να αναπαραχθούν και να μολύνουν κάποιο άλλο αθώο λογισμικό. Όταν το μολυσμένο αθώο λογισμικό εκτελεστεί με την σειρά του τότε μολύνει άλλο λογισμικό κ.ο.κ. Η ιδιότητα αυτή της αναπαραγωγής είναι χαρακτηριστικό των Virus. Είναι επίσης παρασιτικά κακόβουλα λογισμικά, ενώ μεταδίδονται μόνο με χειροκίνητη μεταφορά τους μέσω ανθρώπινης παρέμβασης. Πρακτικά έχουν

απεριόριστες δυνατότητες να εκτελέσουν κάποια κακόβουλη ενέργεια στο μολυσμένο σύστημα.

- **Worm**: Τα Worm μοιράζονται αρκετά χαρακτηριστικά με τα Virus, με το σημαντικότερο να είναι η δυνατότητα αναπαραγωγής του. Σε αντίθεση όμως με την κατηγορία Virus, δεν κρύβονται σε άλλο λογισμικό, δεν είναι δηλαδή παρασιτικά και έχουν την δυνατότητα να εξαπλώνονται από σύστημα σε σύστημα μέσω των δικτύων. Οι δυνατότητες τους να εκτελούν κακόβουλες ενέργειες είναι ανάλογες των Virus, διαφέρουν δηλαδή μόνο στον τρόπο εξάπλωσης και στο ότι δεν είναι παρασιτικά.

- **Rabbit**: Ονομάζονται τα κακόβουλα λογισμικά που αναπαράγονται με γρήγορους ρυθμούς. Υπάρχουν 2 κατηγορίες, αυτών που αναπαράγονται γρήγορα με σκοπό να καταναλώσουν τους πόρους του μολυσμένου συστήματος και αυτών που αναπαράγονται με σκοπό να εξαπλωθούν σε ένα δίκτυο. Η δεύτερη κατηγορία αποτελεί δηλαδή μια ειδική περίπτωση Worm με την διαφορά ότι κατά την εξάπλωση στο δίκτυο πάντα διαγράφεται το προηγούμενο αντίγραφο, υπάρχει δηλαδή πάντα ένα Rabbit στο δίκτυο το οποίο μεταφέρεται μεταξύ των hosts.

- **Spyware**: Είναι κακόβουλα λογισμικά που συλλέγουν πληροφορίες από ένα πληροφοριακό σύστημα και τις μεταδίδουν σε ένα άλλο. Το είδος των πληροφοριών, που μπορεί να συλλέξουν, ποικίλει από στοιχεία πρόσβασης μέχρι email, στοιχεία τραπεζικών λογαριασμών ή άδειες χρήσης λογισμικού. Παρόμοιες πληροφορίες μπορούν να συλλέξουν και τα Viruses και τα Worms με την διαφορά ότι τα Spyware δεν αναπαράγονται Επίσης τα Spyware δεν είναι παρασιτικά λογισμικά αλλά εγκαθίστανται αυτόνομα.

- **Adware:** Είναι παρόμοια με τα Spyware με την διαφορά ότι επικεντρώνονται περισσότερο στην συλλογή πληροφοριών για σκοπούς marketing και την προβολή διαφημιστικών μηνυμάτων στο μολυσμένο πληροφοριακό σύστημα ή στην ανακατεύθυνση των browsers σε εμπορικές ιστοσελίδες. Δεν αναπαράγονται ενώ

έχουν την δυνατότητα της αποστολής των πληροφοριών που συλλέγουν όπως και τα Spyware.

- **Rootkit:** Τα εν λόγω κακόβουλα λογισμικά επιτρέπουν την πρόσβαση σε υπολογιστικά συστήματα με δικαιώματα διαχειριστή ενώ παράλληλα αποκρύπτουν την εκτέλεσή τους από τους πραγματικούς διαχειριστές του συστήματος.

- **Keyloggers:** Καταγράφουν οτιδήποτε πληκτρολογεί ο χρήστης σε ένα σύστημα και το αποστέλλουν στην πηγή του κακόβουλου λογισμικού, με σκοπό την περαιτέρω εκμετάλλευση των πληροφοριών μέσω του διαχωρισμού τους ανάλογα με την σημασία τους.

- **Ransomware:** Τα συγκεκριμένα κακόβουλα λογισμικά κρυπτογραφούν τα δεδομένα ενός υπολογιστικού συστήματος με ισχυρή κρυπτογράφηση και στην συνέχεια απαιτούν λύτρα από τον χρήστη την αποστολή του κλειδιού αποκρυπτογράφησης τους.

2.4 Χρήση Κακόβουλου Λογισμικού σε Απάτες Συναλλαγών

2.4.1 Script-based

Ένα είδος κακόβουλου λογισμικού το οποίο χρησιμοποιείται στις απάτες συναλλαγών, είναι το script based το οποίο στοχεύει τις συναλλαγές μέσω των υπηρεσιών internet banking, που προσφέρουν οι τράπεζες. Αυτό το κακόβουλο λογισμικό δημιουργείται και πωλείται με στόχο συστήματα συγκεκριμένης τράπεζας και είναι «κλειστού» κώδικα, μη επιτρέποντας την περαιτέρω επεξεργασία τους από τους «χειριστές» του. Η μέθοδος που χρησιμοποιείται από λογισμικά αυτού του είδους είναι η «web inject».

Η μέθοδος ενσωματώνει κακόβουλο κώδικα στις απαντήσεις του διακομιστή της τράπεζας προς τον web browser του θύματος, έχοντας, με αυτόν τον τρόπο, την δυνατότητα

να υποκλέψει τα στοιχεία λογαριασμού του χρήστη, τους κωδικούς εισόδου στο τραπεζικό σύστημα ή ακόμα και να ανακατευθύνει τραπεζικές συναλλαγές προς άλλους λογαριασμούς.

Το πλέον επικίνδυνο χαρακτηριστικό αυτού του είδους κακόβουλου λογισμικού είναι η δυνατότητα του να αλλάζει, στην στιγμή, τις λεπτομέρειες της συναλλαγής και τις απαντήσεις του τραπεζικού διακομιστή, να ξεγελάει ακόμα και συστήματα με επιπλέον μεθόδους αυθεντικοποίησης όπως τους κωδικούς TAN (Transaction Authentication Number), συσκευές δημιουργίας κωδικών ή ακόμα και SMS επιβεβαίωσης συναλλαγών. Η δυνατότητα να τροποποιεί τις απαντήσεις του τραπεζικού διακομιστή του επιτρέπει να κρατά ανυποψίαστο το θύμα μεταβάλλοντας την εικόνα των τραπεζικών λογαριασμών που έχει στον browser του. [7]

2.4.2 RAT

Ένα άλλο είδος που συναντάται στις απάτες συναλλαγών είναι το RAT, με χαρακτηριστικότερο εκπρόσωπο του το malware Blackshades, το οποίο πουλήθηκε ή διανεμήθηκε σε χιλιάδες κακόβουλους χρήστες σε περισσότερες από 100 χώρες και με το οποίο μολύνθηκαν περισσότερο από μισό εκατομμύριο ηλεκτρονικοί υπολογιστές.

Το συγκεκριμένο κακόβουλο λογισμικό δίνει πρόσβαση στο μολυσμένο υπολογιστικό σύστημα και παρέχει την δυνατότητα να υποκλαπούν οποιαδήποτε στοιχεία, από τραπεζικούς λογαριασμούς μέχρι κωδικούς πρόσβασης σε οποιαδήποτε διαδικτυακή υπηρεσία. Το Blackshades κόστιζε μόλις 40$ και μπορούσε να παραμετροποιηθεί. [8]

2.4.3 Trojan Zeus

Τέλος, το είδος με τον πιο διάσημο εκπρόσωπο στις απάτες συναλλαγών, είναι τα Trojan Horse, με εκπρόσωπο τους το Trojan Zeus.

Το Trojan Zeus έχει την δυνατότητα να δημιουργεί ένα δίκτυο μολυσμένων συστημάτων ή αλλιώς botnet τα οποία μπορούν να συλλέγουν τεράστιο όγκο πληροφοριών από τους

χρήστες τους, οι οποίες μπορούν να χρησιμοποιηθούν για οποιονδήποτε κακόβουλο σκοπό όπως οι επιθέσεις μεγάλης κλίμακας.

Η χρησιμότητα του Trojan Zeus στις απάτες συναλλαγών έγκειται στην δυνατότητα του να καταγράφει τις επισκέψεις σε ιστοσελίδες και την πληκτρολόγηση κωδικών σε πραγματικό χρόνο. Πολλές εκδόσεις του μάλιστα έχουν την δυνατότητα να μολύνουν και φορητές συσκευές όπως τα κινητά τηλέφωνα σε μια προσπάθεια να υπερπηδήσουν την αυθεντικοποίηση δύο παραγόντων (two-factor authentication) την οποία χρησιμοποιούν πολλά τραπεζικά ιδρύματα για μεγαλύτερη ασφάλεια.

Επιπλέον, το 2011 ο δημιουργός του Zeus έκανε διαθέσιμο στο κοινό τον προγραμματιστικό του κώδικα επιτρέποντας με αυτόν τον τρόπο την δημιουργία νέων πιο αποτελεσματικών εκδόσεων. Το παραπάνω είχε ως συνέπεια, παρόλο που το Zeus έχει στην πλειοψηφία του καταπολεμηθεί αποτελεσματικά, να υπάρχουν ακόμα μέρη του προγραμματιστικού του κώδικα σε νέα κακόβουλα λογισμικά τα οποία έρχονται στην επιφάνεια. [8]

Ένα από τα κακόβουλα λογισμικά, που διαδέχθηκαν το Zeus χρησιμοποιώντας μέρος του προγραμματιστικού του κώδικα, είναι το SpyEye το οποίο είναι υπεύθυνο για απώλειες άνω του 1δις δολάρια μόνο κατά την περίοδο 2010-2012 ,όπου μολύνθηκαν περισσότερα από 50 εκατομμύρια συστήματα παγκοσμίως. Το συγκεκριμένο κακόβουλο λογισμικό χρησιμοποιεί botnet για την υποκλοπή τραπεζικών στοιχείων των θυμάτων. Οι δημιουργοί του συνελήφθησαν το 2013 και καταδικάστηκαν σε πολυετή φυλάκιση.

Στην παρούσα μεταπτυχιακή διατριβή θα ασχοληθούμε συγκεκριμένα με κακόβουλα λογισμικά των οποίων τα παραπάνω αποτελούν χαρακτηριστικά παραδείγματα.

Κεφάλαιο 3

Ανάλυση Κακόβουλου Λογισμικού

3.1 Εισαγωγή

Ανάλυση κακόβουλου λογισμικού ονομάζεται η διαδικασία της συλλογής και μελέτης κακόβουλου λογισμικού, ώστε να διερευνηθεί το πώς λειτουργεί, τι δυνατότητες έχει και πως μπορεί να αναγνωριστεί στο μέλλον. [9]

Σκοπός της ανάλυσης κακόβουλου λογισμικού είναι η κατανόηση του τρόπου λειτουργίας ενός συγκεκριμένου κακόβουλου λογισμικού, ώστε να είναι δυνατή η αντιμετώπιση τόσο του ίδιου όσο και παραλλαγών του βάσει κοινών χαρακτηριστικών. Η διαδικασία της ανάλυσης ξεκινάει συνήθως με την αναγνώριση πιθανής μόλυνσης από κακόβουλο λογισμικό. Από εκείνη την στιγμή είναι απαραίτητο να μελετηθεί ο τρόπος λειτουργίας του κακόβουλου λογισμικού, ώστε να περιοριστεί η μετάδοση του και να γίνει μια αποτίμηση αλλά και περιορισμός της πιθανής ζημιάς που μπορεί να προκαλέσει. Εφόσον εντοπισθούν αρχεία τα οποία αποτελούν μέρος κακόβουλου λογισμικού είναι απαραίτητη η πλήρης ανάλυση τους ώστε να δημιουργηθούν οι ανάλογες ψηφιακές υπογραφές τους, οι οποίες θα χρησιμεύσουν στην περαιτέρω ανίχνευση για κακόβουλο λογισμικό εντός του δικτύου. Επίσης πέρα από τις ψηφιακές υπογραφές μπορεί να γίνει ανάλυση και της συμπεριφοράς κατά την εκτέλεση του κακόβουλου λογισμικού, η οποία μπορεί περιλαμβάνει τις διευθύνσεις της πηγής και του προορισμού του κακόβουλου λογισμικού, τους τύπους επισυναπτόμενων αρχείων στα οποία είναι πιθανώς ενσωματωμένο το κακόβουλο λογισμικό, καθώς και πιθανές στατιστικές ανωμαλίες στα προσβεβλημένα συστήματα. [10] Οι ψηφιακές υπογραφές διακρίνονται σε host-based και network-based,

με τις μεν πρώτες να αφορούν την ανίχνευση κακόβουλου λογισμικού σε συγκεκριμένους υπολογιστές και τις δεύτερες την ανίχνευση ψηφιακών υπογραφών κακόβουλου λογισμικού εντός των δεδομένων που διακινούνται σε ένα δίκτυο υπολογιστών. [11]

Για την μελλοντική προστασία ενός πληροφοριακού συστήματος από ένα κακόβουλο λογισμικό πέρα από την ανάλυση γίνεται ταυτόχρονα και ταξινόμηση βάσει χαρακτηριστικών, την οποία θα μελετήσουμε διεξοδικά σε επόμενο κεφάλαιο της παρούσας μεταπτυχιακής διατριβής.

3.2 Κατηγορίες Ανάλυσης

Η ανάλυση κακόβουλου λογισμικού χωρίζεται σε 4 μεγάλες κατηγορίες, την βασική στατική, την βασική δυναμική και τις εξελιγμένες στατική και δυναμική. [11]

Η βασική στατική ανάλυση περιέχει την μελέτη εκτελέσιμων αρχείων χωρίς να μελετώνται οι εντολές τις οποίες περιέχουν. Λόγω ακριβώς αυτού του χαρακτηριστικού είναι αναποτελεσματική απέναντι σε εξελιγμένο κακόβουλο λογισμικό.

Η βασική δυναμική ανάλυση κακόβουλου λογισμικού αφορά την εκτέλεση του με σκοπό την μελέτη της συμπεριφοράς του, ώστε να καταστεί εφικτή η απομάκρυνσή του από ένα σύστημα ή δημιουργία κατάλληλων υπογραφών για ευκολότερη μελλοντική ανίχνευση του.

Η εξελιγμένη στατική ανάλυση ουσιαστικά αποτελεί την «αποσυναρμολόγηση» του προγραμματιστικού κώδικα ενός κακόβουλου λογισμικού με σκοπό την μελέτη των επιμέρους εντολών και την πλήρη κατανόηση του τρόπου ενέργειας του βήμα βήμα.

Η εξελιγμένη δυναμική ανάλυση χρησιμοποιεί debugger για την μελέτη ενός κακόβουλου λογισμικού την στιγμή που αυτό εκτελείται. Σε περίπτωση που οι υπόλοιπες τεχνικές ανάλυσης αδυνατούν να συλλέξουν τις απαιτούμενες πληροφορίες. [11]

3.3 Εργαστήριο Ανάλυσης

Ένα εργαστήριο ανάλυσης κακόβουλου λογισμικού πρέπει, πρώτα από όλα, να είναι απλό, ώστε να μπορεί να διατηρηθεί υπό όλες τις συνθήκες. Ένα εργαστήριο ανάλυσης πρέπει επίσης να είναι απομονωμένο, καθώς ο κίνδυνος εξάπλωσης του κακόβουλου λογισμικού είναι πάντα υπαρκτός. Τέλος, ένα εργαστήριο ανάλυσης πρέπει να είναι ευέλικτο ώστε να μπορεί να επανασυσταθεί σε σύντομο χρονικό διάστημα.

Τις παραπάνω παραμέτρους ικανοποιεί η τεχνολογία των virtual machines, η οποία επιτρέπει την δημιουργία εικονικών μηχανημάτων στα οποία μπορεί να εκτελεστεί ένα κακόβουλο λογισμικό αναλύοντας παράλληλα την συμπεριφορά του. Τα virtual machines όμως, λόγω της διάδοσης τους και της πρακτικότητας τους στην ανάλυση κακόβουλου λογισμικού, αποτελούν πλέον έναν παράγοντα τον οποίο λαμβάνουν υπόψη οι δημιουργοί κακόβουλου λογισμικού προσαρμόζοντας το λογισμικό τους, ώστε είτε να μην λειτουργεί είτε να λειτουργεί διαφορετικά όταν εκτελείται μέσα από virtual machines, δυσκολεύοντας έτσι την ανάλυση του. [12] Σε πρακτικό επίπεδο λοιπόν, η δημιουργία ενός εργαστηρίου ανάλυσης κακόβουλου λογισμικού εξαρτάται άμεσα από την κατηγορία του κακόβουλου λογισμικού το οποίο πρόκειται να αναλυθεί.

Βλέπουμε λοιπόν, ότι στον σχεδιασμό ενός εργαστηρίου ανάλυσης κακόβουλου λογισμικού λοιπόν θα πρέπει να ληφθούν υπόψη διάφορες παράμετροι, οι οποίοι εν γένει είναι [12]:

- Οι φυσικοί και οικονομικοί περιορισμοί του ερευνητή, βάση των οποίων θα πρέπει να οριστεί το μέγεθος και η τεχνολογία που θα χρησιμοποιηθεί στο εργαστήριο ανάλυσης κακόβουλου λογισμικού.

- Το είδος του κακόβουλου λογισμικού που θα αναλυθεί και το οποίο θα επιβάλει το ελάχιστο όριο των πόρων τους οποίους θα πρέπει να έχουμε διαθέσιμους.

- Η δυνατότητα επανασύστασης του εργαστηρίου σε σύντομο χρονικό διάστημα, στην περίπτωση που η ανάλυση αποτύχει για οποιονδήποτε λόγο και θα πρέπει να επαναληφθεί υπό άλλες συνθήκες.

Σύμφωνα με τα παραπάνω παρατηρούμε ότι η τεχνολογία των virtual machines φαίνεται εκ πρώτης όψεως ιδανική για τον σκοπό της ανάλυσης κακόβουλου λογισμικού όντας οικονομική, γρήγορη στην επαναδημιουργία και ευέλικτη με μόνο μειονέκτημα την δυνατότητα κάποιων κακόβουλων λογισμικών να την παρακάμψουν.

3.4 Ταξινομία και Ονοματοδοσία

Ο όρος Ταξινομία χρησιμοποιήθηκε για πρώτη φορά στην επιστήμη της Βιολογίας και ο ορισμός που της έχει αποδοθεί από τον Σουηδό βιολόγο Carl Linnaeus έχει ως εξής «Ταξινομία είναι η επιστήμη η οποία ονοματίζει, περιγράφει και κατηγοριοποιεί όλους τους ζώντες οργανισμούς». Στην περίπτωση μας, ο όρος ταξινομία χρησιμοποιείται αντίστοιχα για την ονομασία, περιγραφή και κατηγοριοποίηση κακόβουλων λογισμικών βάσει των ιδιοτήτων και του τρόπου λειτουργίας τους.

Οι οργανισμοί, οι οποίοι πρώτοι έρχονται σε επαφή με κάθε νέο κακόβουλο λογισμικό, είναι οι εταιρείες παραγωγής λογισμικού προστασίας ή όπως συνηθίζεται να αποκαλούνται anti-virus. Συνήθως κάθε νέο κακόβουλο λογισμικό ονομάζεται από τον ερευνητή-υπάλληλο της εταιρείας, που θα κληθεί να το αντιμετωπίσει και να παράξει τα αντίστοιχα αντίμετρα. Τα ονόματα που δίνονται συνήθως βασίζονται σε μοναδικά χαρακτηριστικά του συγκεκριμένου κακόβουλου λογισμικού, στις ικανότητες που αυτό έχει ή στην επίδραση που έχει σε ένα πληροφοριακό σύστημα. Η απουσία μιας κεντρικής αρχής ονοματοδοσίας περιπλέκει ακόμα περισσότερο την κατάσταση καθώς κάθε εταιρεία παραγωγής anti-virus έχει την δυνατότητα να δίνει διαφορετικά ονόματα σε κακόβουλα λογισμικά τα οποία αναλύει.

Γενικά στην ονοματοδοσία κακόβουλων λογισμικών χρησιμοποιούνται ευρέως παραλλαγές από συγκεκριμένα αναγνωριστικά με συγκεκριμένη σειρά σύμφωνα με το

σύστημα το οποίο προτάθηκε από τον οργανισμό CARO (Computer AntiVirus Researcher's Organization) στις αρχές της δεκαετίας του 90. [13] Μερικά από αναγνωστικά αυτά με την σειρά την οποία παρουσιάζονται στο όνομα ενός κακόβουλου λογισμικού είναι [6]:

- Ο τύπος του κακόβουλου λογισμικού, π.χ. Virus.

- Η πλατφόρμα λειτουργίας του, π.χ. W32 ή Win32 για Windows 32bit

- Το όνομα της οικογένειας στην οποία ανήκει, το οποίο συνήθως καθορίζεται από τον ερευνητή ο οποίος θα κάνει την ανάλυση.

- Η έκδοση, η οποία συνήθως καθορίζεται με ένα γράμμα του αγγλικού αλφαβήτου.

- Οι τροποποιητές (modifiers), που δίνουν περισσότερες πληροφορίες για το κακόβουλο λογισμικό όπως π.χ. το mm το οποίο σημαίνει mass mailing.

Βάσει των παραπάνω, βλέπουμε πλέον κακόβουλα λογισμικά με ονόματα όπως Worm.Mytob.C ή W32/Mytob.C-mm ή Win32.Worm.Mytob.C ή Worm/Mydoom.BC ,τα οποία ωστόσο αντιστοιχούν στο ίδιο κακόβουλο λογισμικό και κάνουν εμφανές το πρόβλημα της έλλειψης κεντρικής αρχής ονοματοδοσίας.

Το πρόβλημα της ονοματοδοσίας παρουσιάζει αναλυτικά ο Vesselin Bontchev [14], σύμφωνα με τον οποίο:

- Εμφανίζονται μηνιαίως χιλιάδες νέα είδη κακόβουλου λογισμικού
- Υπάρχει έλλειψη χρόνου και εφοδίων για την αντιμετώπιση τους
- Δεν υπάρχει κοινό πρότυπο ονοματοδοσίας
- Δεν υπάρχουν αξιόπιστες μέθοδοι αναγνώρισης και αυτόματης ονοματοδοσίας κακόβουλου λογισμικού
- Δεν υπάρχει η ικανότητα επιβολής συγκεκριμένου προτύπου ονοματοδοσίας

Όλα τα παραπάνω είναι οι βασικές πηγές του προβλήματος της ονοματοδοσίας, για την επίλυση του οποίου έγιναν διάφορες προσπάθειες όπως το CARO, η γεωγραφική ονοματοδοσία, η ονοματοδοσία μολυσματικού μήκους, η περιγραφική ονοματοδοσία, η ονοματοδοσία από κείμενο εντός του κακόβουλου λογισμικού, η αριθμητική, αλλά και η μέθοδο ονοματοδοσίας Bezrukov. [14]

Οι παραπάνω προσπάθειες όμως δεν κατάφεραν να γίνουν ευρέως αποδεκτές λόγω τον εγγενών μειονεκτημάτων της καθεμίας.

Συγκεκριμένα, το CARO δεν κατάφερε να επιβληθεί ως αρχή ονοματοδοσίας ενώ παράλληλα προσπαθούσε να εισάγει τα γνωρίσματα κάθε κακόβουλου λογισμικού ως μέρος του ονόματος, πράγμα το οποίο καθιστούσε τα ονόματα μεγάλα και δύσχρηστα. Η γεωγραφική οδηγούσε σε σύγχυση και είναι μη πρακτική. Η μολυσματικού μήκους δεν μπορεί να περιγράψει κακόβουλο λογισμικό με ίδιο μήκος. Η περιγραφική είναι υποκειμενική και απαιτεί πολύωρη ανάλυση. Η ονοματοδοσία από κείμενο εντός του κακόβουλου λογισμικού προϋποθέτει την ύπαρξη του. Η αριθμητική ονοματοδοσία είναι δύσκολη στο να την θυμάται κάποιος όπως και η μέθοδος Bezrukov η οποία επιπλέον χρησιμοποιεί παρόμοια ονόματα για διαφορετικά κακόβουλα λογισμικά.

Η εμφάνιση κακόβουλου λογισμικού με πολύ μεγάλη συχνότητα δημιουργεί ένα επιπλέον πρόβλημα, αυτό της επικοινωνίας και ανταλλαγής πληροφοριών σχετικά με κακόβουλο λογισμικό το οποίο έχει αναλυθεί, μεταξύ των ενδιαφερόμενων οργανισμών. Έχοντας σαν δεδομένο ότι μια ανάλυση κακόβουλου λογισμικού μας κάνει γνωστά τα γνωρίσματα και την συμπεριφορά του, γίνεται κατανοητή η ανάγκη αυτά να συμπεριλαμβάνονται στην ονοματοδοσία του. Με αυτό τον τρόπο γίνεται αποτελεσματικότερη η αντιμετώπιση και συντομότερος ο χρόνος απόκρισης μετά από μια μόλυνση με κακόβουλο λογισμικό, καθώς ο κάθε οργανισμός γνωρίζει τι αντιμετωπίζει. Επίσης αποφεύγονται οι πολλαπλές αναλύσεις για το ίδιο κακόβουλο λογισμικό.

Με βάση τα παραπάνω καθίσταται εμφανές ότι η περιγραφή ενός κακόβουλου λογισμικού με βάση τα μοτίβα επίθεσης, τα γνωρίσματα και τις ενέργειες του θα έπρεπε να ενσωματωθεί σε ένα ευρέως χρησιμοποιούμενο πρότυπο. Με αυτό τον τρόπο, θα μπορούσε

όχι μόνο να κωδικοποιηθεί η συμπεριφορά και τα γνωρίσματα ενός κακόβουλου λογισμικού αλλά και να γίνει ευκολότερη η ανίχνευσή του, όπως και να καθοριστεί το μέγεθος της απειλής που αυτό αντιπροσωπεύει, διευκολύνοντας έτσι την αντιμετώπιση του.

3.4.1 Συστήματα CME και MAEC

Προς την κατεύθυνση της κωδικοποίησης της συμπεριφοράς και των γνωρισμάτων ενός κακόβουλου λογισμικού κινήθηκε το Malware Attribute Enumeration and Characterization (MAEC). [15]

Οι βασικές αδυναμίες του συστήματος ονοματοδοσίας CARO έγινε προσπάθεια να επιλυθούν από τον μη κερδοσκοπικό οργανισμό MITRE δημιουργώντας το CME (Common Malware Enumeration), με σκοπό την δημιουργία αναγνωριστικών, σε αντικατάσταση των ονομάτων που προσέδιδαν οι κατασκευαστές anti-virus, σε ένα κοινό ευρετήριο κακόβουλων λογισμικών. Σκοπός της παραπάνω προσπάθειας, δεν ήταν η αντικατάσταση των ευρέως χρησιμοποιούμενων ονομάτων από την βιομηχανία καταπολέμησης κακόβουλου λογισμικού, αλλά η δημιουργία ενός κοινού προτύπου ταξινομίας. Αυτό επιτεύχθηκε με την χρήση ενός αναγνωριστικού CME, το οποίο προστίθετο στο όνομα του κακόβουλου λογισμικού, βοηθώντας έτσι στην δημιουργία ενός παγκόσμιου ευρετηρίου ως σημείο αναφοράς.

Στις αρχές του 2007 παρατηρήθηκε ότι το νέο κακόβουλο λογισμικό, που παρουσιαζόταν, είχε αλλάξει σκοπό και αντί να εμφανίζονται λίγα κακόβουλα λογισμικά στοχεύοντας σε ευρεία γκάμα συστημάτων, εμφανίζονταν πολλά κακόβουλα λογισμικά με πολύ συγκεκριμένους στόχους. Η τάση αυτή είχε ως στόχο να εμποδίσει την ανίχνευση τους μέσω ψηφιακών υπογραφών και κατά συνέπεια έριξε το βάρος της ανίχνευσης στα heuristics και σε άλλες τεχνικές, οι οποίες δεν βασίζονται στην ψηφιακή υπογραφή. Τα παραπάνω κατέστησαν αδύνατη την χρήση των αναγνωριστικών CME και ακόμα εμφανέστερη την ανάγκη ύπαρξης κοινής γλώσσας επικοινωνίας για τα βασικά χαρακτηριστικά του κακόβουλου λογισμικού.

Η αποτυχία του CME να ακολουθήσει την εξέλιξη του κακόβουλου λογισμικού οδήγησε στην δημιουργία του DHS/DoD/NIST Software Assurance Malware Working Group, το οποίο περιλάμβανε αντιπροσώπους από το MITRE και την Anti-Spyware Coalition (ASC). Το Malware Working Group κλήθηκε να δημιουργήσει ένα πλαίσιο περιγραφικών γνωρισμάτων κακόβουλου λογισμικού. Σκοπός του πλαισίου αυτού ήταν [15]:

- Να δημιουργήσει πρότυπα αναγνώρισης βασισμένα στα γνωρίσματα και της συμπεριφορές του κακόβουλου λογισμικού, βελτιώνοντας έτσι την επικοινωνία μεταξύ των ενδιαφερομένων.

- Να επιτρέψει στους χρήστες να γνωρίζουν πότε ένα λογισμικό μπορεί να είναι κακόβουλο πριν το εγκαταστήσουν σε οποιοδήποτε σύστημα.

- Να δημιουργήσει νομικούς όρους για το κακόβουλο λογισμικό.

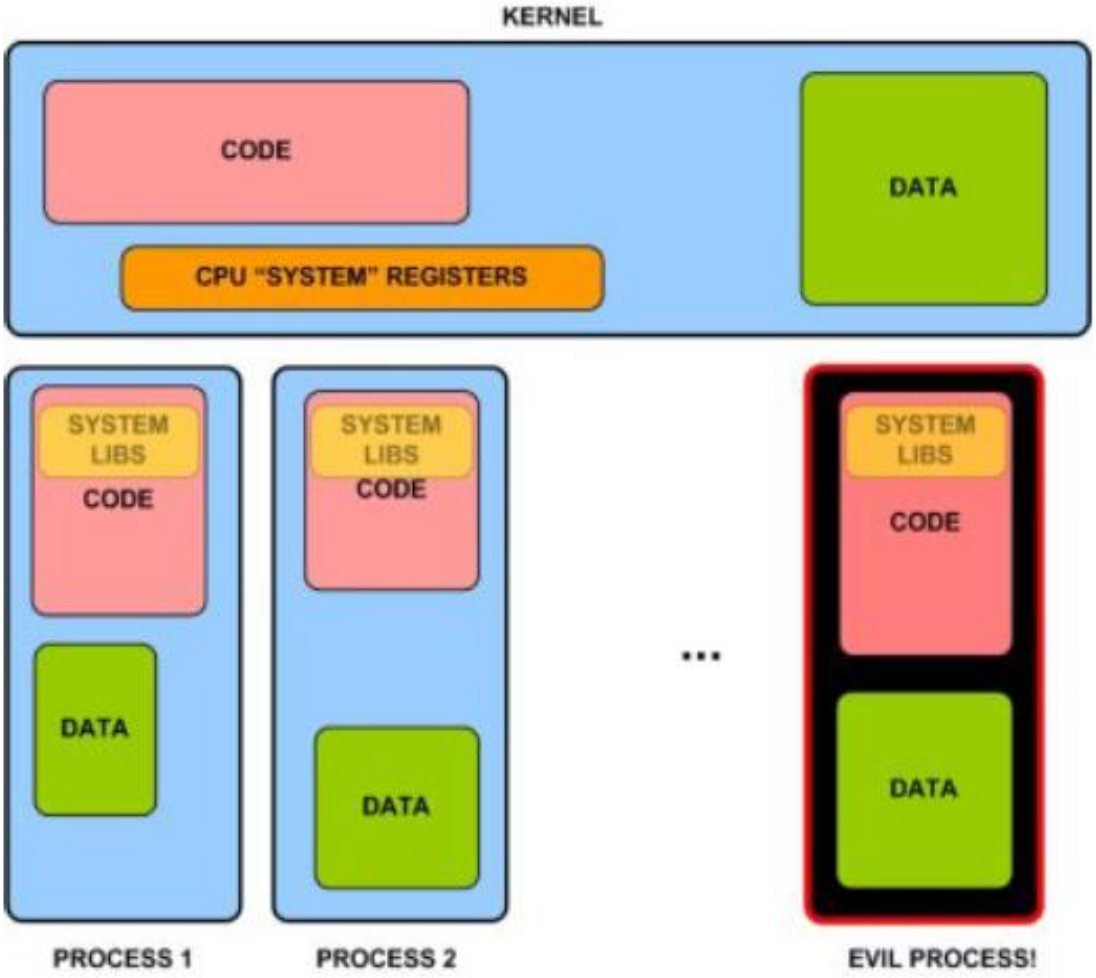
- Να δημιουργήσει αντικειμενικά κριτήρια αξιολόγησης του λογισμικού ανίχνευσης και καταπολέμησης κακόβουλου λογισμικού.

Η παραπάνω προσπάθεια οδήγησε στην δημιουργία του MAEC, το οποίο πέρα από τον νομικό ορισμό του κακόβουλου λογισμικού στόχευε στην χρήση του ως κοινή μέθοδο κατηγοριοποίησης του κακόβουλου λογισμικού βάσει των γνωρισμάτων και των συμπεριφορών του. Αυτό θα επέτρεπε την περιγραφή ενός κακόβουλου λογισμικού με χρήση συγκεκριμένων γνωρισμάτων αντί των μεταδεδομένων της ψηφιακής τους υπογραφής, τα οποία όπως είδαμε είχαν ήδη καταστεί παρωχημένα.

3.4.2 Άλλες Μορφές Ταξινομίας

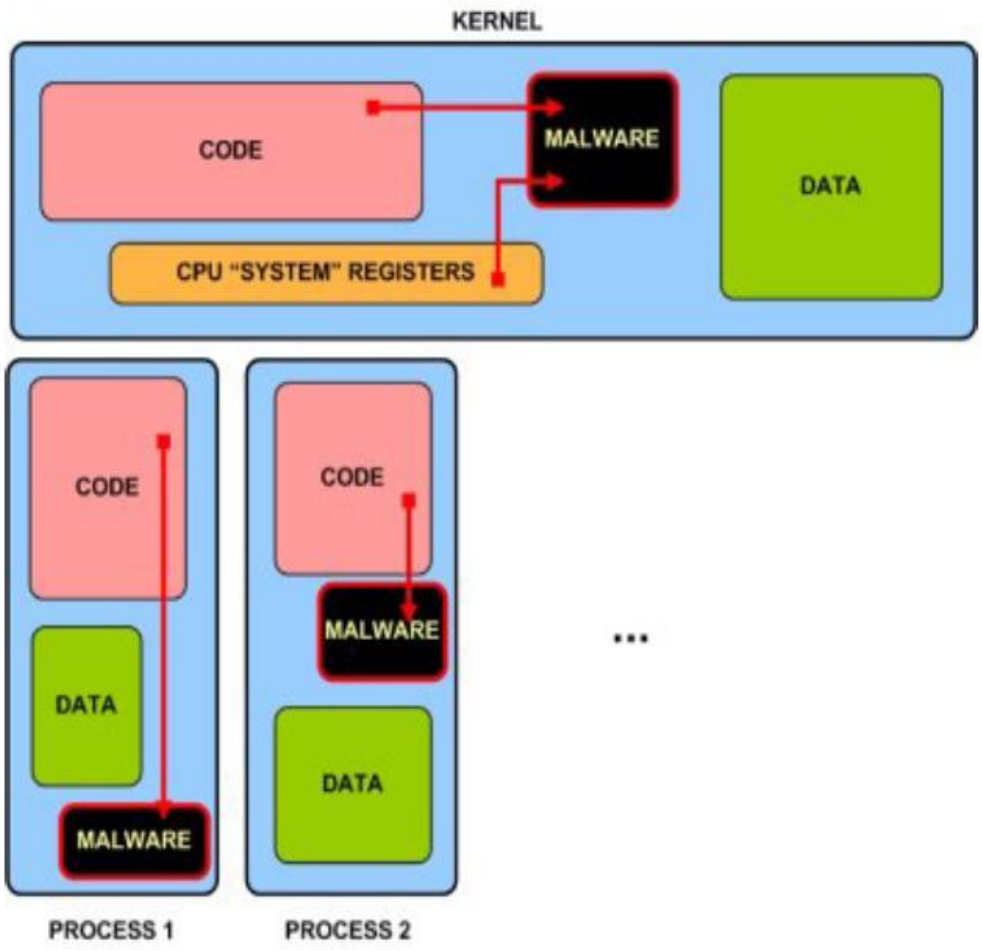
Τέλος μια ακόμα εναλλακτική πρόταση σε ότι αφορά την ταξινομία κακόβουλου λογισμικού προτάθηκε από την Joanna Rutkovska το 2006. Σύμφωνα με την πρόταση της, ο τρόπος αλληλεπίδρασης του κακόβουλου λογισμικού με το λειτουργικό σύστημα του μολυσμένου πληροφοριακού συστήματος θα μπορούσε να χρησιμοποιηθεί για την ταξινόμηση τους σε τέσσερις κατηγορίες. [16] Οι κατηγορίες αυτές είναι:

- **Type 0**, κατά την οποία το κακόβουλο λογισμικό δεν αλληλεπιδρά με το λειτουργικό σύστημα, με την έννοια ότι δεν αλλάζει την συμπεριφορά του ή άλλων εφαρμογών οι οποίες λειτουργούν στο εν λόγω σύστημα.



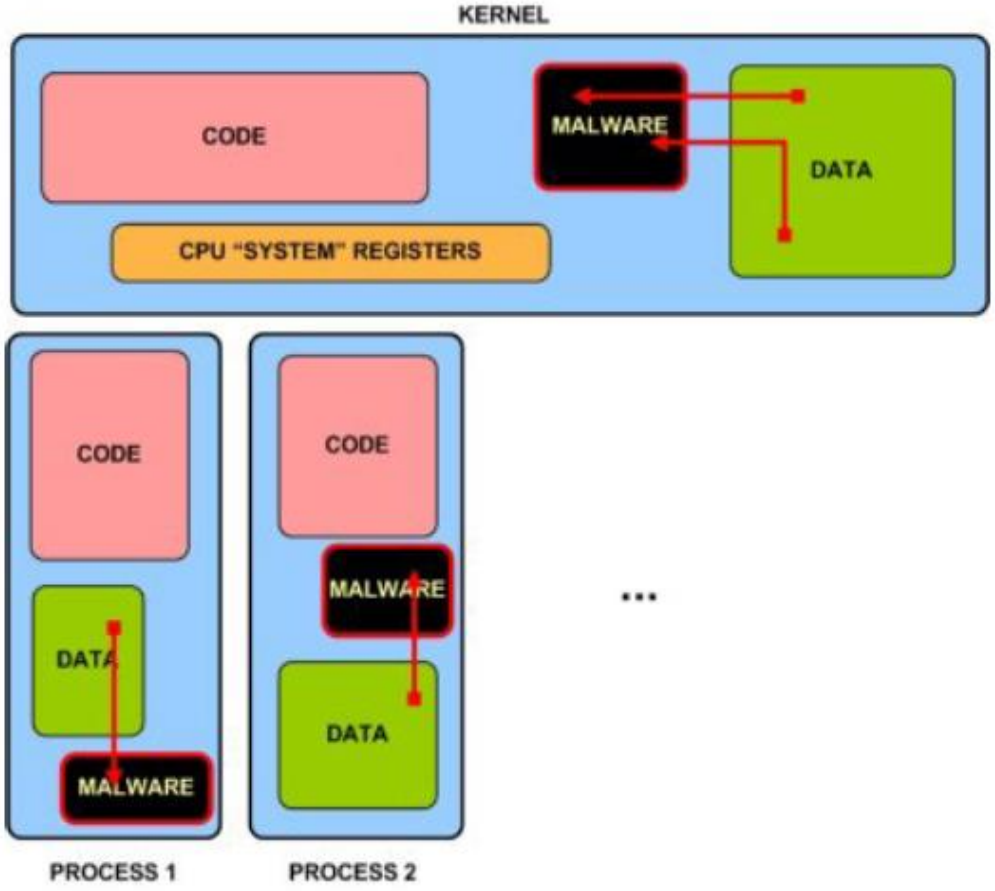
Εικόνα 3.1 Κατηγορία Type 0 [16]

- **Type I**, της οποίας το κακόβουλο λογισμικό επιδρά μόνο σε σταθερές μεταβλητές του λειτουργικού συστήματος όπως το BIOS, εκτελέσιμα αρχεία ή διεργασίες και κομμάτια κώδικα αποθηκευμένα στην μνήμη.



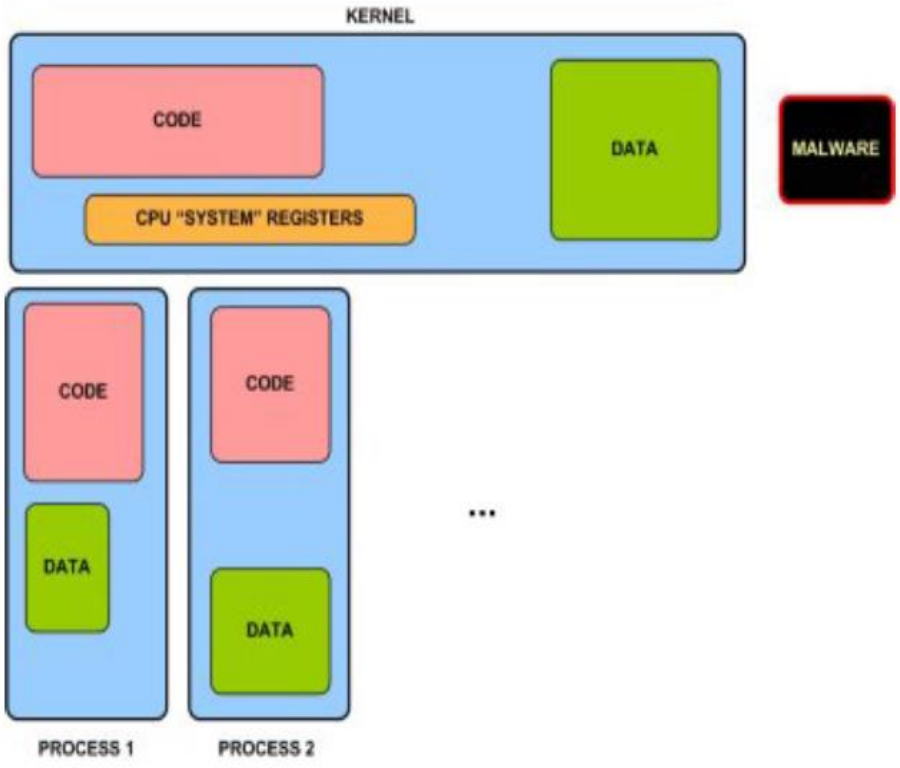
Εικόνα 3.2 Κατηγορία Type I [16]

- **Type II**, της οποίας το κακόβουλο λογισμικό επιδρά σε δεδομένα τα οποία χρησιμοποιεί το λειτουργικό σύστημα και εκτελείται αντί αυτών.



Εικόνα 3.3 Κατηγορία Type II [16]

- **Type III**, της οποίας κατηγορίας τα κακόβουλα λογισμικά είναι και τα πλέον επικίνδυνα καθώς έχουν την ικανότητα να αναλάβουν το έλεγχο ενός λειτουργικού συστήματος χωρίς να αλλάξουν το παραμικρό. Τα εν λόγω κακόβουλα λογισμικά είναι σε θέση να εκμεταλλευτούν τις δυνατότητες virtualization των σύγχρονων επεξεργαστών, ώστε να ενεργούν χωρίς να γίνονται αντιληπτά καθώς δεν επηρεάζουν την ακεραιότητα τους συστήματος.



Εικόνα 3.4 Κατηγορία Type III [16]

Παρατηρούμε ότι η μέθοδος κατηγοριοποίησης των κακόβουλων λογισμικών με ανάλυση της συμπεριφοράς τους (behavioral analysis), κατά την οποία τα κακόβουλα λογισμικά κατηγοριοποιούνται με βάση το πώς αυτά ενεργούν εντός του μολυσμένου πληροφοριακού συστήματος είναι η πλέον διαφωτιστική. Για τον παραπάνω λόγο αυτή θα είναι και η μέθοδος κατηγοριοποίησης την οποία θα χρησιμοποιήσουμε στην παρούσα μεταπτυχιακή διατριβή.

Κεφάλαιο 4

Περιπτώσεις Κακόβουλου Λογισμικού για Απάτες Συναλλαγών

4.1 Zeus

Zeus (επίσης γνωστό ως Zbot) ονομάζεται ένα σετ εργαλείων το οποίο χρησιμοποιείται για να δημιουργηθούν διαφορετικές εκδόσεις κακόβουλου λογισμικού υποκλοπής προσωπικών στοιχείων. Το κακόβουλο λογισμικό, το οποίο δημιουργείται με αυτά τα εργαλεία, τρέχει αθόρυβα στο background του μολυσμένου λειτουργικού συστήματος συλλέγοντας πληροφορίες, τις οποίες στέλνει πίσω στον δημιουργό του κακόβουλου λογισμικού. Η κύρια χρήση του είναι η υποκλοπή τραπεζικών στοιχείων και κυρίως στοιχείων αυθεντικοποίησης. [17]

Τα εργαλεία αυτά διατίθενται προς αγορά στο διαδίκτυο και είναι πολύ εύκολο να χρησιμοποιηθούν οδηγώντας στην δημιουργία τεράστιου αριθμού botnets με διαφορετικούς χρήστες. Αυτό έχει ως αποτέλεσμα, η βιομηχανία Antivirus να είναι αντιμέτωπη με έναν τεράστιο αριθμό κακόβουλου λογισμικού το οποίο έχει διαφορετικά χαρακτηριστικά.

4.1.1 Ανάλυση

Οι πρώτες εκδόσεις του Zeus είχαν σαν κοινό στοιχείο την χρήση αρχείων με το ίδιο όνομα σαν φορείς του κακόβουλου λογισμικού. Μερικά παραδείγματα ήταν μεταξύ άλλων τα:

- Ntos.exe
- Oembios.exe
- Twext.exe

Ενώ τα αρχεία δεδομένων αποθηκεύονταν στους φακέλους:

- \wsnpoem
- \wsnpoem
- \twain_32

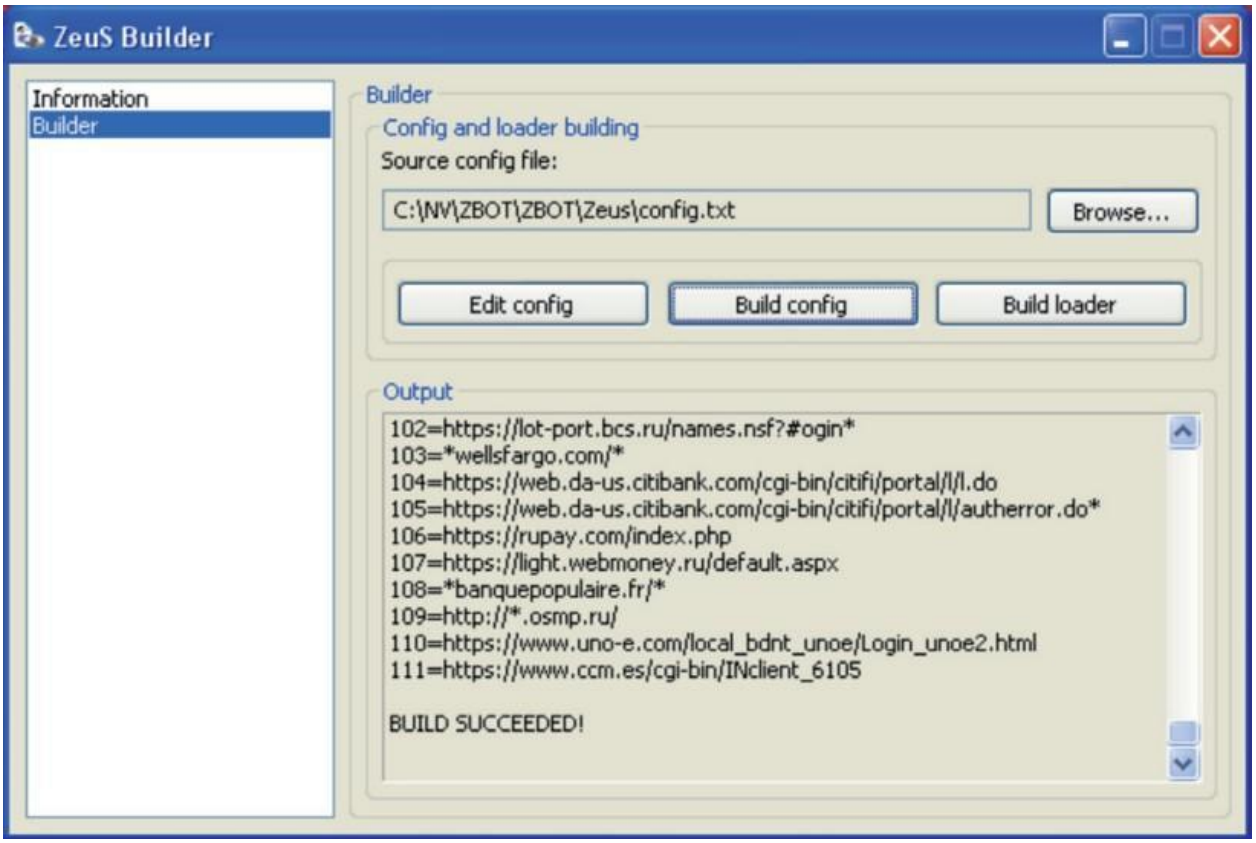
Η επόμενη ευρέως διαδεδομένη έκδοση του Zeus χρησιμοποιούσε κυρίως το αρχείο sdra64.exe και αποθήκευε τα δεδομένα στον φάκελο \lowsec. Οι τελευταίες εκδόσεις χρησιμοποιούν τυχαία εκτελέσιμα αρχεία ως φορείς και αποθηκεύουν δεδομένα σε τυχαίους υποφακέλους εντός του φάκελου Application Data του χρήστη. [17]

4.1.1.1 Δομικά Στοιχεία

Βασικό δομικό στοιχείο του Zeus είναι ο Builder, μέσω του οποίου δημιουργούνται οι διάφορες εκδόσεις του κακόβουλου λογισμικού. Συγκεκριμένα μέσω του Builder δημιουργούνται το encrypted configuration file και το εκτελέσιμο αρχείο. Το εκτελέσιμο αρχείο είναι μοναδικό για κάθε χρήστη ακόμα και αν χρησιμοποιούν την ίδια έκδοση builder, καθώς το URL του configuration file και το encryption key το οποίο χρησιμοποιείται ενσωματώνονται στο εκτελέσιμο αρχείο.

Το configuration file είναι το πρώτο που δημιουργείται καθώς σε αυτό ο χρήστης περιλαμβάνει όλες τις πληροφορίες που αυτός θέλει και στην συνέχεια το δημιουργεί με το

Build Config. Μέσω του build config ο builder μετατρέπει το text file σε binary το οποίο θα μπορεί στην συνέχεια να χρησιμοποιηθεί από το εκτελέσιμο αρχείο ενώ στην συνέχεια το συμπιέζει και το κάνει encrypt. Ακολούθως, ο χρήστης τοποθετεί το encrypted αρχείο στο URL, το οποίο έχει ορίσει για να χρησιμοποιηθεί από το εκτελέσιμο αρχείο, όταν ζητηθεί κατά την εκτέλεση του. (Εικ. 4.1)



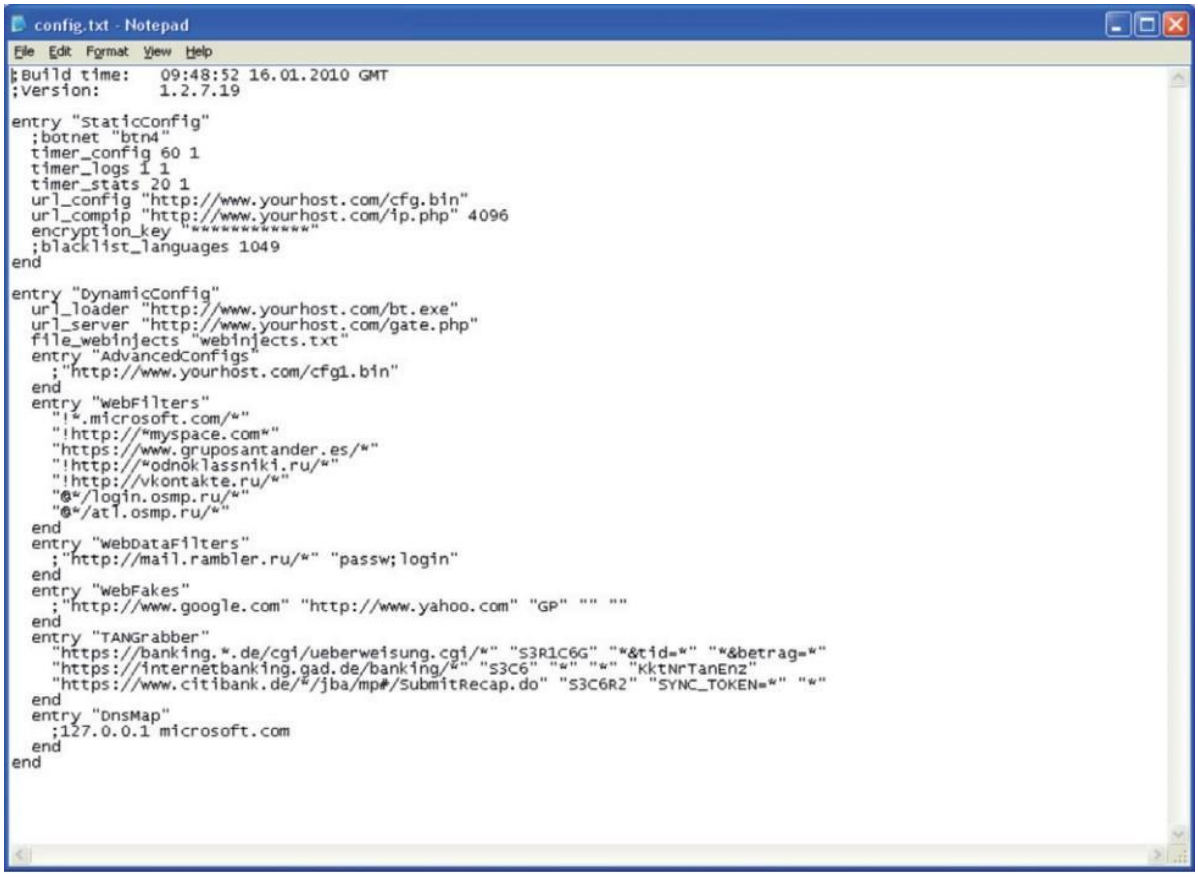
Εικόνα 4.1 Zeus Builder [17]

Το configuration file είναι το ποιο σημαντικό αρχείο του Zeus καθώς αυτό ορίζει τις ενέργειες τις οποίες θα εκτελεί. Περιέχει μεταξύ άλλων την διεύθυνση στην οποία θα στέλνονται τα υποκλαπέντα στοιχεία.

Το format, το οποίο παίρνει το configuration file, είναι μια σειρά blocks τα οποία ενεργοποιούν και ορίζουν τις διάφορες ενέργειες του Zeus. Παρακάτω, στην Εικόνα 4.2, βλέπουμε ότι τα διάφορα blocks αρχίζουν με την λέξη entry η οποία ορίζει είτε StaticConfig είτε DynamicConfig. Αυτά τα δυο καθορίζουν τα settings τα οποία θα ενσωματωθούν σε

binary καθώς και αυτά που θα γραφτούν στο configuration file και θα γίνουν download κατά την εκτέλεση. (Εικ. 4.2)

Οι επιλογές του StaticConfig περιλαμβάνουν το χρονισμό του κακόβουλου λογισμικού, δηλαδή πόσο να περιμένει επιχειρώντας να κατεβάσει το configuration file από το URL που έχει οριστεί, το ίδιο το URL καθώς και ένα URL το οποίο χρησιμοποιείται για να ελεγχθεί η εξωτερική διεύθυνση IP με την οποία το κακόβουλο λογισμικό επικοινωνεί με τον χρήστη του.



Εικόνα 4.2 Zeus Configuration file [17]

Οι επιλογές του DynamicConfig επικεντρώνονται κυρίως σε συγκεκριμένες διευθύνσεις web, τις οποίες στοχεύει ο χρήστης του κακόβουλου λογισμικού συμπεριλαμβανομένων των παρακάτω:

- Ένα URL από το οποίο το οποίο θα κατέβει το εκτελέσιμο αρχείο

- Ένα URL στο οποίο θα σταλούν τα υποκλαπέντα δεδομένα

- Ένα URL από το οποίο μπορεί να κατέβει ένα συμπληρωματικό configuration file

Επιπλέον επιλογές, οι οποίες συμπεριλαμβάνονται, είναι:

- URL μάσκες οι οποίες ενεργοποιούν ή απενεργοποιούν το logging για τα συγκεκριμένα URL

- Ζευγάρια URL τα οποία κάνουν redirect το ένα στο άλλο

- Ομάδες URL από τα οποία θα συλλέγονται TAN's (Transaction Authentication Numbers)

- Ομάδες IP/domains τα οποία θα εγγραφούν στο μολυσμένο σύστημα ώστε να τροποποιούν τα DNS requests

- Μάσκες URL, οι οποίες θα κάνουν inject κώδικα HTML σε κάθε σελίδα της οποίας το request θα ταιριάζει με το URL της μάσκας.

Η τελευταία επιλογή είναι και αυτή με την οποία ο χρήστης του Zeus μπορεί να εκμεταλλευτεί οικονομικά το κακόβουλο λογισμικό. Μέσω του injection δεδομένων σε κάθε ιστοσελίδα που επιθυμεί μπορεί να προσθέσει έξτρα στοιχεία στις φόρμες αυθεντικοποίησης μια σελίδας web banking και μέσω αυτών να υποκλέψει επιπλέον στοιχεία όπως PIN από ATM χωρίς αυτά να περνούν από την διαδικασία encryption της ιστοσελίδας web banking.

Όλα τα δεδομένα του DynamicConfig αποθηκεύονται στο configuration file το οποίο με την σειρά του είναι αποθηκευμένο σε έναν server. Το εκτελέσιμο αρχείο στην συνέχεια αναζητά στο URL που του έχει δοθεί το configuration file και εφαρμόζει όσα αυτό περιέχει. Με αυτό τον τρόπο, η λειτουργία του κακόβουλου λογισμικού καθίσταται δυναμική με την

συμπεριφορά του να μεταβάλλεται ανάλογα με τον στόχο του χρήστη, ο οποίος έχει την δυνατότητα να αλλάζει το configuration file κατά βούληση.

Το εκτελέσιμο αρχείο είναι αυτό που γίνεται build από τον Builder και το οποίο μοιράζεται καθ' οποιονδήποτε τρόπο από τον χρήστη. Διαφορετικοί χρήστες του Zeus μπορούν να δημιουργήσουν πανομοιότυπα εκτελέσιμα αρχεία χρησιμοποιώντας την ίδια έκδοση Builder με μόνη διαφορά το URL στο οποίο θα βρίσκεται το configuration file. Με όλα τα άλλα στοιχεία του configuration file ίδια, η συμπεριφορά και οι ενέργειες του Zeus θα είναι ακριβώς ίδιες.

Ο server του Zeus είναι ένα σύνολο scripts γραμμένα σε γλώσσα PHP, μέσω των οποίων ο χρήστης έχει την δυνατότητα να επιτηρεί την κατάσταση των κακόβουλων λογισμικών στα μολυσμένα συστήματα καθώς και να μεταβάλλει την λειτουργικότητα τους μέσω των configuration files τους. Επίσης, με αυτό τον τρόπο συλλέγει και όλα τα υποκλαπέντα στοιχεία από τα διάφορα συστήματα.

4.1.1.2 Binary File

Το κακόβουλο λογισμικό Zeus γενικά μπορεί να εκτελέσει τις παρακάτω ενέργειες:

- Να αντιγράφει τον εαυτό του σε άλλη τοποθεσία, να εκτελέσει το αντίγραφο και να διαγράψει το παλιό αρχείο
- Να αλλάξει τα security settings του internet explorer επεμβαίνοντας στην registry
- Να κάνει inject κώδικα σε άλλες διεργασίες
- Να υποκλέψει στοιχεία αυθεντικοποίησης αποθηκευμένα στο σύστημα
- Να κατεβάσει και να χρησιμοποιήσει config files
- Να στείλει δεδομένα πίσω στον χρήστη του

Οι τελευταίες εκδόσεις του Zeus όμως, έχουν αρκετές διαφορές στην εκτέλεση τους. Οι προηγούμενες εκδόσεις αντέγραφαν τον εαυτό τους, συνήθως με το όνομα sdra64.exe, στον φάκελο του λειτουργικού συστήματος και δημιουργούσαν έναν φάκελο με τον όνομα lowsec, στον οποίο αποθήκευαν το κατεβασμένο configuration file καθώς και ένα

προσωρινό αρχείο με τα υποκλαπέντα δεδομένα πριν αυτά σταλούν στον χρήστη του κακόβουλου λογισμικού. Επίσης, δημιουργούσαν μια εγγραφή στην registry στο Userinit κάτω από το "HKLM\Software\Microsoft\ Windows NT\CurrentVersion\Winlogon για να εκτελούνται σε κάθε εκκίνηση του συστήματος. [17]

Στις πρόσφατες εκδόσεις το Zeus αντιγράφει τον εαυτό του στον φάκελο Application Data, χρησιμοποιώντας ένα τυχαίο όνομα τόσο για το ίδιο το εκτελέσιμο αρχείο όσο και για τον φάκελο που δημιουργεί. Ένα προσωρινό αρχείο δεδομένων αποθηκεύεται επίσης στον φάκελο %AppData% με τυχαίο όνομα και το configuration file αποθηκεύεται στο registry αντί σε φάκελο του συστήματος. Επίσης δημιουργεί ένα runkey στο HKCU της registry αντί να χρησιμοποιεί το Winlogon, πράγμα που σημαίνει ότι ένα σύστημα μπορεί να μολυνθεί από διαφορετικά αντίγραφα του Zeus, όπως επίσης και ότι το σύστημα μπορεί να μολυνθεί ανεξάρτητα από τον λογαριασμό χρήστη.

Τέλος, το κακόβουλο λογισμικό Zeus χρησιμοποιεί τεχνικές απόκρυψης από ανίχνευση μέσω Checksum. Οι παλαιότερες εκδόσεις ενσωμάτωναν τυχαία δεδομένα στο αρχείο binary έτσι ώστε να μεταβάλλεται το checksum. Στις νεότερες εκδόσεις χρησιμοποιείται μια πιο περίπλοκη τεχνική για να εξασφαλίσει ότι το αρχείο το οποίο εισάγεται/αντιγράφεται σε ένα σύστημα έχει διαφορετικό checksum από το αρχείο που το εισήγαγε. Όταν το αρχείο του Zeus εισέρχεται σε ένα σύστημα μέσω της αντιγραφής στον φάκελο Application Data, ενσωματώνεται σε αυτό ένα μικρό block κρυπτογραφημένων δεδομένων. Τα δεδομένα αυτά περιέχουν την διαδρομή στην οποία εισήχθη το αρχείο καθώς και ένα GUID του δίσκου στον οποίο βρίσκεται. Επειδή, όπως αναφέραμε προηγουμένως, ο φάκελος που δημιουργείται έχει τυχαία ονόματα, η διαδρομή η οποία περιέχεται σε αυτό το μικρό block δεδομένων θα είναι πάντα διαφορετική και κατά συνέπεια και το checksum του. Το παραπάνω καθιστά την πιθανότητα να παραχθούν 2 ίδια αρχεία μηδενική.

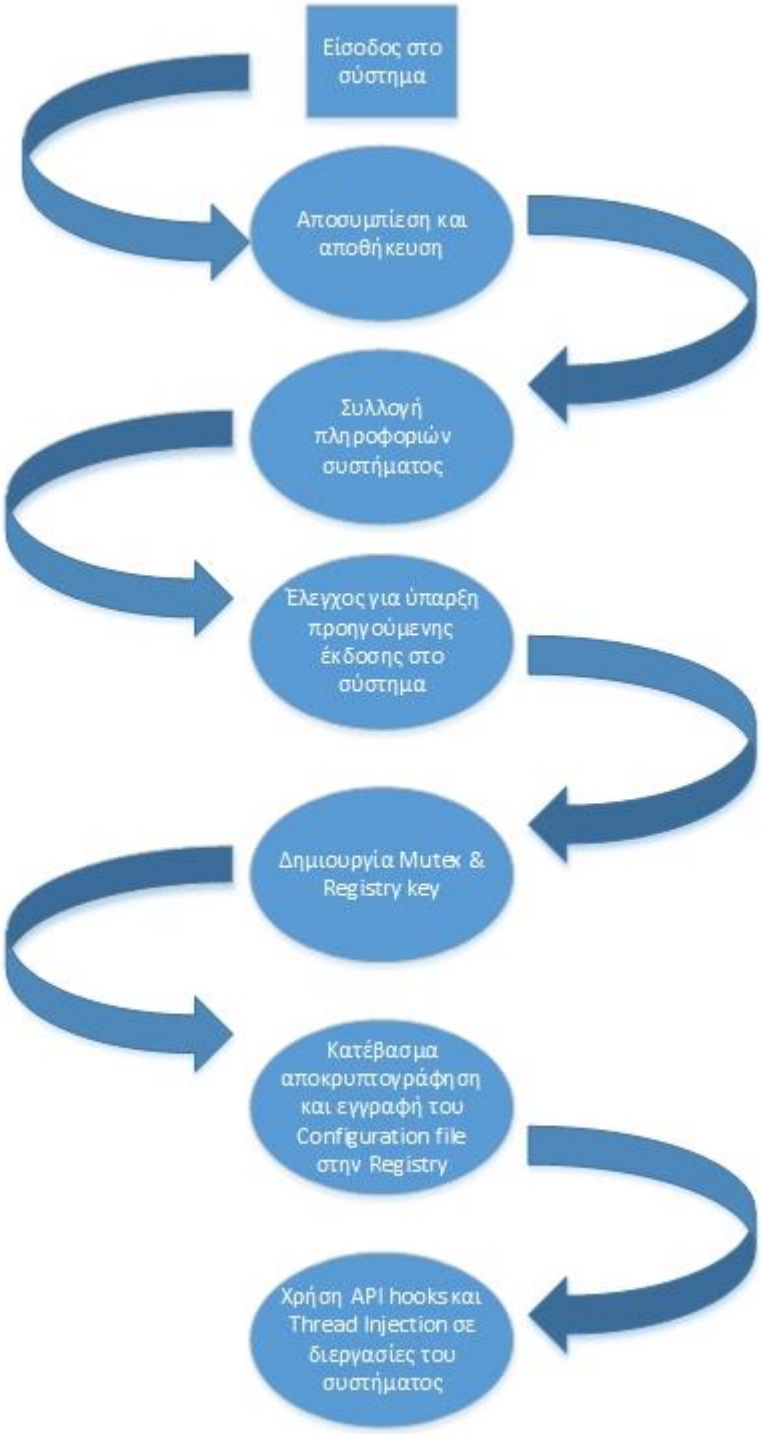
4.1.2 Τρόπος Λειτουργίας

Όλες οι εκδόσεις του Zeus, αρχικά, για λόγους απόκρυψης, γίνονται pack με την βοήθεια κάποιου packer σε κάποια γνωστή και φαινομενικά ακίνδυνη μορφή αρχείου όπως zip, rar, jpg, pdf. Ο packer μεταξύ άλλων συμπιέζει το μέγεθος του αρχείο για ευκολότερη

διανομή, όπως επίσης και για απόκρυψη καθώς τα συμπιεσμένα αρχεία είναι δυσκολότερο να ελεγχθούν.

Η πρώτη ενέργεια του Zeus σε ένα σύστημα είναι, αφού αποσυμπιεστεί και αποθηκευτεί, να συλλέξει πληροφορίες για το σύστημα, όπως την έκδοση του λειτουργικού, το process ID και το access level που έχει. Στην συνέχεια, θα προχωρήσει στην εκτέλεση των εντολών του. Μια από τις πρώτες εντολές είναι ο καθορισμός για το εάν πρόκειται για το πρώτο αρχείο στο σύστημα και χρειάζεται να αντιγραφεί στο %AppData% ή εάν βρίσκεται ήδη εκεί και πρέπει να προχωρήσει στις επόμενες ενέργειες. Για να το πετύχει αυτό, θα αντιγράψει τον εκτελέσιμο κώδικα του στην μνήμη RAM και θα βρει την πρώτη εγγραφή με το όνομα .data αντιγράφοντας 0x200 bytes από αυτή την τοποθεσία στον σωρό των δεδομένων. Στην συνέχεια αποκρυπτογραφεί αυτά τα bytes με χρήση RC4 και το τελευταίο dword από τα αποκρυπτογραφημένα δεδομένα χρησιμοποιείται ως flag για να προσδιορίσει εάν πρόκειται για το πρώτο δείγμα κακόβουλου λογισμικού στο σύστημα ή εάν έχει ήδη αντιγράψει τον εαυτό του.

Υποθέτοντας ότι είναι το πρώτο δείγμα στο εν λόγω σύστημα, το Zeus θα δημιουργήσει ένα mutex, δηλαδή ένα mutual exclusion, ώστε να επιτραπεί η χρήση κοινών resources από τα διάφορα threads του. Το όνομα του mutex θα προέρχεται από την κρυπτογράφηση του αρχείου, ώστε να είναι μοναδικό, αποκλείοντας έτσι την λειτουργία πολλών αντιγράφων του συγκεκριμένου κακόβουλου λογισμικού στο εν λόγω σύστημα. Η χρήση mutex επιτρέπει όμως την λειτουργία άλλων εκδόσεων του ίδιου κακόβουλου λογισμικού. Μέσω του mutex αποκρυπτογραφείται η διεργασία, η οποία θα αντιγράψει το κακόβουλο λογισμικό στον φάκελο του συστήματος για τον οποίο προορίζεται. Η μέθοδος κρυπτογράφησης που χρησιμοποιείται είναι byte-wise XOR με key μεγέθους 4 bytes. Το κλειδί και το μέγεθος των δεδομένων που θα αποκρυπτογραφηθούν ορίζονται στα 0x200 bytes, τα οποία είχαν αποκρυπτογραφηθεί νωρίτερα και πρόκειται να αντικατασταθούν στο αντιγραμμένο αρχείο όπως αναλύσαμε νωρίτερα, ώστε αυτό να μην έχει την δυνατότητα να ακολουθήσει την ίδια διαδικασία δημιουργώντας ένα loop.



Εικόνα 4.3 Λειτουργία του Zeus

Στην συνέχεια δημιουργείται ένα τυχαίο κλειδί κάτω από το κλειδί HKCU\Software\Microsoft του μητρώου συστήματος (registry). Σε αυτό το σημείο αποθηκεύονται και τα δεδομένα του configuration ενώ δημιουργείται ένα νέο block

δεδομένων μεγέθους 0x200 bytes προς αντικατάσταση του παλαιού. Το block αυτό θα περιέχει σημαντικές πληροφορίες τις οποίες θα χρειαστεί το αντιγραμμένο αρχείο για να λειτουργήσει. Αυτές είναι οι πληροφορίες του μολυσμένου μηχανήματος (computer name, έκδοση OS, ημερομηνία εγκατάστασης OS, OS product ID), ένα GUID το οποίο θα αναγνωρίζει τον δίσκο στον οποίο αντιγράφηκε το αρχείο, το RC4 encryption key για την αποκρυπτογράφηση του configuration file και την κρυπτογράφηση των υποκλαπέντων στοιχείων, την διαδρομή στο %AppData% όπου έχει αντιγραφεί το αρχείο καθώς και το όνομα του κλειδιού που δημιουργήθηκε στην registry. Τέλος το block αυτό γράφεται πάνω από το παλιό block δεδομένων και το αρχείο αποθηκεύεται στον τυχαίο φάκελο κάτω από το %AppData% με ημερομηνία και ώρα του παρελθόντος ενώ το αντιγραμμένο αρχείο εκτελείται διαγράφοντας το αρχικό με την βοήθεια ενός batch file.

Το τελικό αρχείο του Zeus θα κάνει αυτοέλεγχο για το εάν πρόκειται όντως για το τελικό αρχείο ή εάν πρέπει να αντιγραφεί. Αυτό θα συμβεί αποκρυπτογραφώντας τα πρώτα 0x1e6 bytes από το ενσωματωμένο, όπως είδαμε προηγουμένως, block των 0x200 bytes. Αυτή την φορά όμως το κλειδί RC4 θα είναι διαφορετικό από αυτό που χρησιμοποιήθηκε για τον έλεγχο του πρώτου αρχείου που έχει εισαχθεί στο σύστημα και πριν αυτό αντιγραφεί. (Εικ. 4.3)

Dropper	Droppee
Decrypt 0x200 byte block	Decrypt 0x200 byte block
Check DWORD value at offset 0x1E6	Check DWORD value at offset 0x1E6
Decrypt file creation routine	Decrypt first 0x1E6 bytes of block using different key
Write and execute new file	Verify block running on same system as created on
Write and execute self-deletion batch script	Inject into other processes
End	Continue ...

Εικόνα 4.4 Αποκρυπτογράφηση Zeus [17]

Στην συνέχεια γίνεται έλεγχος του GUID του δίσκου στον οποίο είναι αποθηκευμένο το αρχείο, όπως και στην διαδρομή, ώστε να εξασφαλισθεί ότι στο σύστημα θα τρέχει μόνο ένα instance. Εάν κάποιος από τους παραπάνω ελέγχους αποτύχει τότε το πρόγραμμα θα τερματιστεί.

Το επόμενο βήμα είναι η εισαγωγή ενός νέου thread σε κάθε διεργασία στην οποία έχει πρόσβαση το κακόβουλο λογισμικό. Με χρήση API's, τα οποία θα ενσωματώσει στις διεργασίες τις οποίες έχει πρόσβαση το κακόβουλο λογισμικό θα αποκτήσει την δυνατότητα να υποκλέπτει δεδομένα όσο αυτά διακινούνται στο μολυσμένο σύστημα. Κάποιες από τις διεργασίες που χρησιμοποιεί το Zeus είναι οι:

- dwm.exe
- taskhost.exe
- taskeng.exe
- wscntfy.exe
- ctfmon.exe
- rdpclip.exe
- explorer.exe

Εάν γίνει injection ενός κακόβουλου thread σε κάποιες από τις παραπάνω διεργασίες τότε εκτελούνται επιπλέον threads, τα οποία μπορούν να «ακούν» διάφορα ports, να κατεβάζουν και να επεξεργάζονται το configuration file, όπως και να ελέγχουν αν το runkey του κακόβουλου λογισμικού στην registry βρίσκεται στην θέση του, αλλά και να το επαναφέρουν αν έχει διαγραφεί.

Η επεξεργασία του configuration file είναι μια επίσης σημαντική διεργασία, η οποία εκτελείται. Το URL του configuration file καθώς και το RC4 encryption key για την αποκρυπτογράφηση του είναι κρυπτογραφημένα και ενσωματωμένα στο binary αρχείο του Zeus. Μόλις, η διεργασία, η οποία είναι υπεύθυνη για το κατέβασμα του configuration file, ενεργοποιηθεί θα ξεκινήσει την αποκρυπτογράφηση του URL και του decryption key του και θα κατεβάσει το configuration file από το συγκεκριμένο URL. Στην συνέχεια θα

αποκρυπτογραφήσει το configuration file και θα συγκρίνει το MD5 hash του με μια τιμή στο header του file ώστε να βεβαιωθεί ότι αυτό δεν έχει αλλάξει. [17]

Στην συνέχεια τα δεδομένα του configuration file κρυπτογραφούνται πάλι και εγγράφονται στο τυχαίο registry key που δημιουργήθηκε στα προηγούμενα βήματα. Το encryption key για αυτή την κρυπτογράφηση είναι διαφορετικό από αυτό που χρησιμοποιήθηκε για την αποκρυπτογράφηση του configuration file μετά το κατέβασμα του και είναι αυτό το οποίο βρίσκεται στο block δεδομένων, το οποίο εγγράφηκε στα πρώτα βήματα λειτουργίας, κατά την αντιγραφή του κακόβουλου λογισμικού στο σύστημα.

Η όλη διαδικασία περιγράφεται συνοπτικά από τα παρακάτω βήματα:

- Αποκρυπτογράφηση συγκεκριμένης περιοχής του binary
- Εξαγωγή URL και κλειδιού RC4 No1 από τα αποκρυπτογραφημένα δεδομένα
- Κατέβασμα του configuration file και χρήση του κλειδιού No1 για την αποκρυπτογράφηση του
- Σύγκριση MD5 τιμών μεταξύ κρυπτογραφημένων και μη δεδομένων
- Χρήση του κλειδιού No1 για την αποκρυπτογράφηση του block δεδομένων, που εγγράφηκε αρχικά από το πρώτο αρχείο του κακόβουλου λογισμικού στο σύστημα
- Εξαγωγή του κλειδιού αποκρυπτογράφησης No2 από τα αποκρυπτογραφημένα δεδομένα
- Κρυπτογράφηση δεδομένων με χρήση του κλειδιού No2 και εγγραφή στην registry

Το 2ο κλειδί RC4 χρησιμοποιείται επίσης για την κρυπτογράφηση των υποκλαπέντων δεδομένων κατά την προσωρινή αποθήκευση τους στο μολυσμένο σύστημα και την αποστολή τους στον χρήστη του κακόβουλου λογισμικού.

Μετά την εγγραφή του configuration file στην registry, η ίδια διεργασία θα συνεχίσει με την αναζήτηση οποιουδήποτε εκτελέσιμου αρχείου το οποίο μπορεί να υποδεικνύει το configuration file, όπως επίσης θα αναζητήσει τυχόν ενημερωμένες εκδόσεις του configuration εφόσον αυτές υποδεικνύονται με σχετικό URL στο κατεβασμένο αρχείο.

Μετά από τα παραπάνω βήματα θεωρούμε ότι ένα σύστημα έχει μολυνθεί και σε εκείνο το σημείο ξεκινάει η πραγματική λειτουργία του Zeus, η οποία είναι η υποκλοπή δεδομένων.

Η λειτουργία αυτή επιτυγχάνεται με την χρήση API hooks εντός διεργασιών του συστήματος όπως οι παρακάτω:

DLL	**API**
ntdll.dll	NtCreateThread (pre Vista)
ntdll.dll	NtCreateUserProcess (Vista and later)
ntdll.dll	LdrLoadDll
kernel32.dll	GetFileAttributesExW
wininet.dll	HttpSendRequest
wininet.dll	HttpSendRequestEx
wininet.dll	InternetCloseHandle
wininet.dll	InternetReadFile
wininet.dll	InternetReadFileEx
wininet.dll	InternetQueryDataAvailable
wininet.dll	HttpQueryInfo
ws2_32.dll	closesocket
ws2_32.dll	send
ws2_32.dll	WSASend
user32.dll	OpenInputDesktop
user32.dll	SwitchDesktop
user32.dll	DefWindowProc
user32.dll	DefDlgProc
user32.dll	DefFrameProc
user32.dll	DefMDIChildProc
user32.dll	CallWindowProc
user32.dll	RegisterClass
user32.dll	RegisterClassEx
user32.dll	BeginPaint
user32.dll	EndPaint

user32.dll	GetDCEx
user32.dll	GetDC
user32.dll	GetWindowDC
user32.dll	ReleaseDC
user32.dll	GetUpdateRect
user32.dll	GetUpdateRgn
user32.dll	GetMessagePos
user32.dll	GetCursorPos
user32.dll	SetCursorPos
user32.dll	SetCapture
user32.dll	ReleaseCapture
user32.dll	GetCapture
user32.dll	GetMessage
user32.dll	PeekMessage
user32.dll	TranslateMessage
user32.dll	GetClipboardData
crypt32.dll	PFXImportCertStore
nspr4.dll	PR_OpenTCPSocket
nspr4.dll	PR_Close
nspr4.dll	PR_Read
nspr4.dll	PR_Write

Πίνακας 4.1 Αρχεία dll και API που χρησιμοποιούνται από το Zeus [17]

Στον Πίνακα 4.1 βλέπουμε ένα μικρό δείγμα των πρακτικά ανεξάντλητων δυνατοτήτων του συγκεκριμένου λογισμικού το οποίο είναι εξαιρετικά επικίνδυνο ιδιαίτερα σε ότι αφορά τις ηλεκτρονικές συναλλαγές.

4.2 Hesperbot

Ένας νέος τύπος κακόβουλου λογισμικού, το οποίο στοχεύει σε τραπεζικές συναλλαγές, είναι το Hesperbot. Το συγκεκριμένο κακόβουλο λογισμικό ανακαλύφθηκε να στοχεύει τραπεζικά συστήματα σε Τουρκία, Τσεχία, Πορτογαλία και Ηνωμένο Βασίλειο. Το ενδιαφέρον ήταν ότι κατά την ανακάλυψη του αποκαλύφθηκε ότι χρησιμοποιούσε σαν host ένα domain το οποίο φαινομενικά άνηκε στα Τσέχικα ταχυδρομεία. [18]

Η ανάλυση του εν λόγω κακόβουλου λογισμικού αποκάλυψε ότι ενώ είχε παρόμοια λειτουργία με το Zeus και το SpyEye, παρουσίαζε σημαντικές διαφορές στην υλοποίηση, με αποτέλεσμα να θεωρηθεί ως τελείως νέο και όχι ως παραλλαγή των ήδη γνωστών

κακόβουλων λογισμικών. Παρά την νεότητα του αποδείχθηκε ότι το Hesperbot είναι πολύ ικανό, εφοδιασμένο με δυνατότητες keystroke logging, δημιουργίας screenshots, λήψης video αλλά και δημιουργίας απομακρυσμένου proxy. Πέραν αυτών, περιείχε και ιδιαίτερα εξελιγμένες δυνατότητες, όπως την δημιουργία ενός κρυφού VNC server στο μολυσμένο σύστημα, την υποκλοπή δεδομένων που διακινούνται στο δίκτυο αλλά και δυνατότητα HTML injection. Στόχος των χρηστών του κακόβουλου λογισμικού ήταν η υποκλοπή των στοιχείων αυθεντικοποίησης των τραπεζικών λογαριασμών των θυμάτων αλλά επίσης και η εγκατάσταση κακόβουλου λογισμικού στις κινητές τους συσκευές, όπως Symbian, BlackBerry και Android.

Το Hesperbot χρησιμοποιεί το domain www.ceskaposta.net το οποίο είναι παρόμοιο με το www.ceskaposta.cz των Τσέχικων ταχυδρομείων. Για την εγκατάσταση του στα συστήματα των θυμάτων χρησιμοποιεί emails με συνημμένα exe, κρυμμένα σε αρχεία pdf, με την ονομασία "zasilka" το οποίο σημαίνει αλληλογραφία στα τσέχικα. Με αυτό τον τρόπο, το θύμα, βλέποντας ότι το mail προέρχεται από ιστοσελίδα παρόμοια με αυτή των ταχυδρομείων, εύκολα παρασύρεται στο να ανοίξει το συνημμένο αρχείο. Παρ' όλη την επίσημη ενημέρωση, η οποία πραγματοποιήθηκε από τα Τσέχικα ταχυδρομεία μετά την αποκάλυψη του κακόβουλου λογισμικού, ήδη αρκετά συστήματα είχαν μολυνθεί. Στην Τουρκία χρησιμοποιήθηκε παρόμοια μέθοδος με αυτή της Τσεχίας. Εκεί το phishing mail είχε την μορφή τιμολογίου προερχόμενο από τον μεγαλύτερο ISP της χώρας όπως και στην Πορτογαλία, όπου τα κακόβουλα αρχεία είχαν την μορφή τιμολογίου της Portugal Telecom.

Στην πορεία της έρευνας για το Hesperbot αποκαλύφθηκε επίσης ότι χρησιμοποιούσε ένα επιπλέον κακόβουλο λογισμικό ως δομικό στοιχείο, το Win32/Spy.Agent.OEC, με σκοπό την υποκλοπή διευθύνσεων emails από τα μολυσμένα συστήματα για την διεύρυνση των πιθανών στόχων. Επίσης διαπιστώθηκε ότι το configuration file του κακόβουλου λογισμικού περιείχε ρητά τα τραπεζικά συστήματα κάθε χώρας, τα οποία θα στόχευε για υποκλοπή HTTP επικοινωνίας και HTML injection. Τα web injects με χρήση HTML, στην Τουρκία και στην Πορτογαλία, είχαν ως σκοπό την αλλοίωση των ιστοσελίδων των συγκεκριμένων τραπεζικών ιδρυμάτων στα μολυσμένα συστήματα, ενώ αντίθετα στην Τσεχία δεν χρησιμοποιήθηκαν web injects αλλά πιθανότατα key loggers ή κάποια μέθοδος form grabbing.

Σύμφωνα με το ESET Livegrid [18], δεκάδες συστήματα μολύνθηκαν στην Τσεχία, ενώ εκατοντάδες στην Τουρκία και στην Πορτογαλία.

4.2.1 Ανάλυση

Όπως και στα περισσότερα κακόβουλα λογισμικά το Heresbot είναι ένα modular λογισμικό το οποίο περιλαμβάνει διαφορετικά δομικά στοιχεία. Αρχικό στοιχείο του αποτελεί το πρώτο αρχείο το οποίο θα εισαχθεί σε ένα σύστημα και θα ξεκινήσει την μόλυνση του. Όπως είδαμε και στο προηγούμενο κεφάλαιο, με το Zeus, το αρχικό αρχείο αναλαμβάνει την εγκατάσταση και εξάπλωση του κακόβουλου λογισμικού στο σύστημα και είναι συνήθως δημιουργημένο με την χρήση κάποιου packer σε κάποιο γνωστό filetype, ώστε να αποκρύπτεται ο ρόλος του.

4.2.1.1 Δομικά Στοιχεία

Στην περίπτωση του Hesperbot, το αρχικό αρχείο το οποίο εισάγεται σε ένα σύστημα κάνει injection τον πυρήνα του στην διεργασία explorer.exe ενώ, στην συνέχεια, ο πυρήνας αυτός αναλαμβάνει την εγκατάσταση των υπολοίπων modules και plug-ins, τα οποία θα χρησιμοποιηθούν για κακόβουλες ενέργειες. [18]

Για το injection του πυρήνα στο explorer.exe ακολουθείται μια από τις παρακάτω διαδικασίες:

- Εκκίνηση μιας νέας διεργασίας explorer.exe και χρήση του NtGetContextThread στην εκκίνηση ώστε να χρησιμοποιηθεί διαφορετικός κώδικας

- Injection σε υπάρχουσα διεργασία του explorer.exe χρησιμοποιώντας την διαδικασία Shell_TrayWnd/SetWindowLong/SendNotifyMessage, η οποία είναι γνωστή από το PowerLoader και άλλα κακόβουλα λογισμικά

- Injection στο explorer.exe με χρήση του CreateRemoteThread

Το ποια από τις παραπάνω μεθόδους θα χρησιμοποιηθεί καθορίζεται εάν στο σύστημα βρίσκονται τα αρχεία cmdguarg.sys ή klif.sys, τα οποία είναι drivers των προγραμμάτων προστασίας Comodo και Kaspersky αντίστοιχα. Ο πυρήνας του κακόβουλου λογισμικού εφόσον έχει πια ενεργοποιηθεί στο explorer.exe χειρίζεται την επικοινωνία με τον Command & Control (C&C) server του κακόβουλου λογισμικού, όπως επίσης και την εγγραφή των απαραίτητων κλειδιών στην registry του συστήματος. Για την επικοινωνία με τον C&C server χρησιμοποιεί κάποιο ενσωματωμένο URL ή δημιουργεί νέο URL με την χρήση αλγόριθμου στην περίπτωση που το αρχικό URL δεν είναι διαθέσιμο. Στην συνέχεια στέλνει τις παρακάτω πληροφορίες:

- Το όνομα του βασισμένο στο computer name του μολυσμένου συστήματος
- Το όνομα του Botnet στο οποίο ανήκει με βάση την χώρα εξάπλωσης
- Τις διευθύνσεις IP των καρτών δικτύου
- Ονόματα ενεργών smart cards
- Πληροφορίες για τα εγκατεστημένα κακόβουλα plugins

Ο C&C server αντίστοιχα μπορεί να απαντήσει με:

- Το configuration file
- Plugin modules
- Ένα εκτελέσιμο αρχείο
- Ή μια νέα έκδοση κακόβουλου λογισμικού προς εγκατάσταση

Το ενδιαφέρον στο συγκεκριμένο κακόβουλο λογισμικό είναι η δυνατότητα του να απαριθμεί τις smart cards, οι οποίες είναι παρούσες στο σύστημα, με χρήση συγκεκριμένων API's χωρίς να επεμβαίνει στην λειτουργία τους.

Στην συνέχεια ο πυρήνας κρυπτογραφεί τα δεδομένα που έχει λάβει από τον C&C server με χρήση του αλγορίθμου Twofish και με κλειδί μήκους 256bit του οποίου η παραγωγή βασίζεται στο όνομα του μολυσμένου συστήματος, στην ημερομηνία εγκατάστασης του λειτουργικού, στην έκδοση λειτουργικού, στην αρχιτεκτονική του επεξεργαστή καθώς και στα MachineGuid και DigitaProductId τα οποία λαμβάνει από την registry. Τα δεδομένα αυτά αποθηκεύονται σε τυχαία δημιουργημένο φάκελο εντός του %AppData%.

Τέλος ο πυρήνας έχει την δυνατότητα να κάνει inject τον εαυτό του σε οποιαδήποτε διεργασία εκτελείται.

4.2.2 Τρόπος Λειτουργίας

Εκτός από την λειτουργία του αρχικού αρχείου και του core, ο οποίος γίνεται inject από το Hesperbot, τα διάφορα plugins και modules που εγκαθίστανται από τον core είναι τα πλέον σημαντικά και αυτά τα οποία εκτελούν τον τελικό σκοπό του κακόβουλου λογισμικού, δηλαδή την υποκλοπή στοιχείων. Παρακάτω θα προσπαθήσουμε να αναλύσουμε τα σημαντικότερα από αυτά.

4.2.2.1 Υποκλοπή Δεδομένων Δικτύου και Web-Injects

Η δυνατότητα υποκλοπής δεδομένων τα οποία διακινούνται στο δίκτυο είναι μια από τις πιο ενδιαφέρουσες δυνατότητες του Hesperbot. Το ενδιαφέρον αυτό προκύπτει από το γεγονός ότι το Hesperbot σε αντίθεση με άλλα γνωστά κακόβουλα λογισμικά, όπως το Zeus, δεν χρησιμοποιεί την μέθοδο Man-in-the-Browser, τροποποιώντας πακέτα HTTP και HTTPS για να υποκλέψει δεδομένα εντός του μολυσμένου browser αλλά ξεχωριστά plugins.

Τα plugins αυτά είναι τα Nethk, Httphk και Httpi, τα οποία συνεργάζονται μεταξύ τους ώστε να επιτύχουν το επιθυμητό αποτέλεσμα.

Συγκεκριμένα το Nethk δημιουργεί έναν proxy σε τυχαίο port στην IP 127.0.1.1 και ενσωματώνει τις ακόλουθες λειτουργίες στο αρχείο mswock.dll

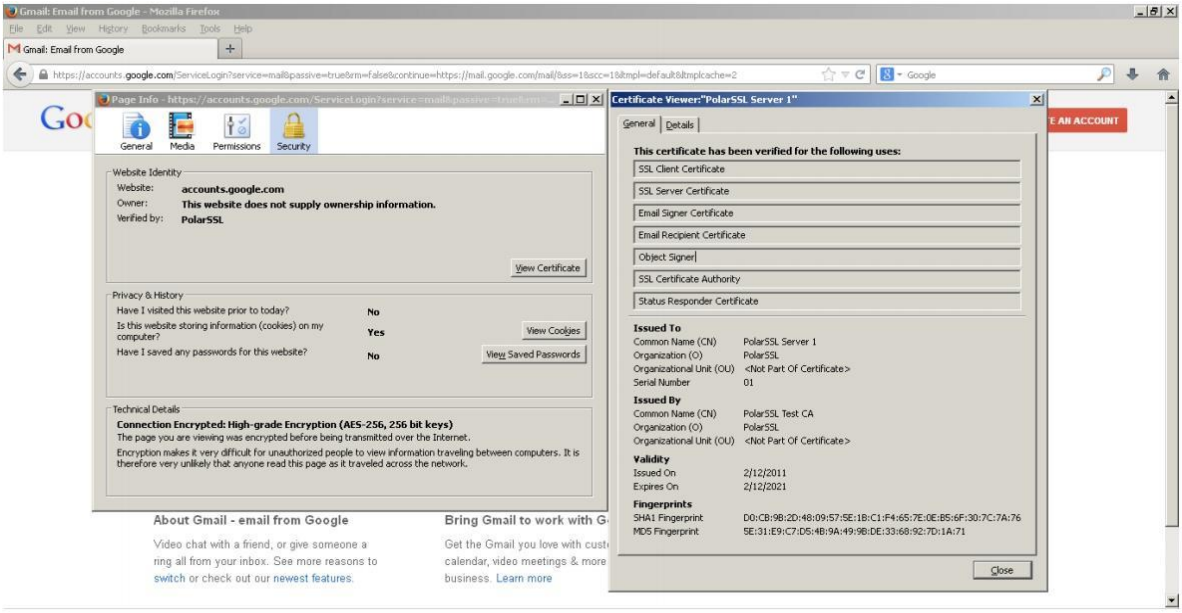
- WSPSocket
- WSPloctl
- WSPConnect
- WSPCloseSocket

Στην Εικόνα 4.4, βλέπουμε πως ο browser ενώνεται με τον proxy ο οποίος έχει δημιουργηθεί από το κακόβουλο λογισμικό αντί με το website του τραπεζικού οργανισμού μέσω της ενσωμάτωσης κακόβουλου κώδικα στο API WSPConnect.

Εικόνα 4.5 Σύνδεση με Hesperbot proxy [18]

Κάθε φορά που ο proxy ανιχνεύει ένα request από τον browser, καλεί το plugin httphk πριν στείλει το request στον πραγματικό του προορισμό, όπως και αντίστοιχα ενεργοποιεί το httphk όταν ο πραγματικός server στείλει απάντηση. Το plugin httphk είναι

ουσιαστικά υπεύθυνο για την διαχείριση της δικτυακής κίνησης που υποκλέπτεται. Αρχικά διαχωρίζονται τα HTTP με τα HTTPS πακέτα. Στην περίπτωση των πρώτων το nethk απλά στέλνει τα δεδομένα στο httphk plugin. Στην περίπτωση όμως των HTTPS πακέτων που αποστέλλονται από τον browser, το nethk τα αποκρυπτογραφεί και στην συνέχεια τα στέλνει στο httphk το οποίο τα κρυπτογραφεί και πάλι. Για αυτή την διαδικασία το nethk χρησιμοποιεί ένα πλαστό ψηφιακό πιστοποιητικό για SSL δημιουργημένο από τον μολυσμένο browser ενώ το httpk το κρυπτογραφεί με το πραγματικό πιστοποιητικό για SSL το οποίο χρησιμοποιείται από τον web server του τραπεζικού ιδρύματος και το προωθεί στον πραγματικό προορισμό. Όταν λαμβάνονται πακέτα από τον webserver του τραπεζικού ιδρύματος ακολουθείται η αντίστροφη διαδικασία χρησιμοποιώντας στην πραγματικότητα μια μέθοδο Man-in-the Middle. Για την αποφυγή αποκάλυψης του κακόβουλου λογισμικού λόγω λανθασμένου πιστοποιητικού SSL, το nethk φέρει δικά του self-signed πιστοποιητικά SSL με τα οποία αντικαθιστά τα πραγματικά πιστοποιητικά (Εικ 4.5).



Εικόνα 4.6 Αντικατάσταση SSL Certificate της Google με κατασκευασμένο από το Hesperbot [18]

Για την αποφυγή οποιουδήποτε μηνύματος στον browser για ψεύτικο πιστοποιητικό ασφαλείας, το Hesperbot τροποποιεί τις διεργασίες οι οποίες είναι υπεύθυνες για την επαλήθευση των πιστοποιητικών. Στην Εικόνα 4.6 βλέπουμε τους browsers που υποστηρίζονται και τις διεργασίες που τροποποιούνται.

Browser process	Hooked functions
iexplore.exe maxthon.exe avant.exe sleipnir.exe webkit2webprocess.exe browser.exe chrome.exe deepnet.exe	CertVerifyCertificateChainPolicy and CertGetCertificateChain in crypt32.dll
firefox.exe seamonkey.exe k-meleon.exe	CERT_VerifyCertificate, CERT_VerifyCert, CERT_VerifyCertificateNow, CERT_VerifyCertNow and CERT_VerifyCertName in nss3.dll
opera.exe	Function in opera.dll

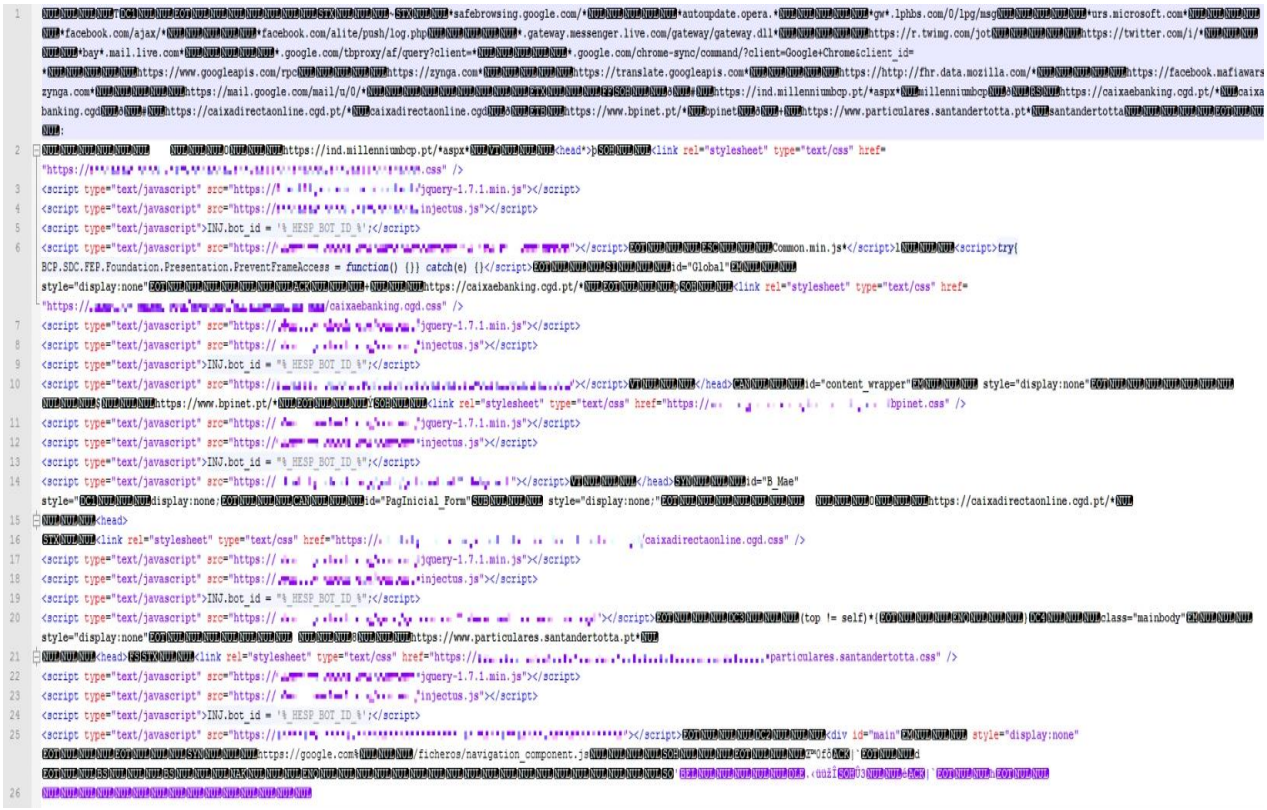
Εικόνα 4.7 Hesperbot Browser Hooks [18]

Μια ενδιαφέρουσα πτυχή είναι η χρήση hash αντί ονομάτων διεργασιών browsers, ώστε να αποφευχθεί ανίχνευση του κακόβουλου λογισμικού από μεθόδους αντιμετώπισης βασισμένες στις ψηφιακές υπογραφές όπως φαίνεται στην Εικόνα 4.7.

Εικόνα 4.8 Χρήση Hash από το Hesperbot [18]

Το module httpi είναι αυτό το οποίο σύμφωνα με τις οδηγίες του configuration file μεταβάλλει τα δεδομένα HTTP. Όταν καλείται η διεργασία httpi_request-callback, το module httpi ξεκινάει την ανάγνωση του configuration file και ελέγχει το URL, εφόσον στο configuration file περιέχονται εντολές περί λήψης screenshots και video τότε ξεκινάει την δημιουργία τους. Στην συνέχεια ελέγχει εάν υπάρχει ένα POST request και εάν το περιεχόμενο είναι είτε "application/ x-www-form-urlencoded" είτε "text/plain". Εφόσον ισχύουν οι προηγούμενες συνθήκες είναι πιθανό ο χρήστης να συμπλήρωσε και να απέστειλε τα στοιχεία αυθεντικοποίησης του και τα δεδομένα υποκλέπτονται και αποθηκεύονται σε ένα log, εφόσον στο configuration file ορίζεται ότι το συγκεκριμένο URL πρέπει να παρακολουθείται.

Στην συνέχεια καλείται η διεργασία httpi_response_callback, η οποία ελέγχει αν η απάντηση HTTP έχει τον κωδικό 200, έπειτα διαβάζει το configuration file για να βρει εάν υπάρχουν στοιχεία web inject για την συγκεκριμένη ιστοσελίδα, η οποία απάντησε, και τα εισάγει στον κώδικα HTML. Στην Εικόνα 4.8 βλέπουμε το configuration file, το οποίο χρησιμοποιήθηκε στην έκδοση του κακόβουλου λογισμικού η οποία αφορούσε την Πορτογαλία. Στο συγκεκριμένο αρχείο βλέπουμε στην αρχή URL σχετικά με μέσα κοινωνική δικτύωσης και ηλεκτρονικού ταχυδρομείου τα οποία γενικά αγνοούνται από το httpi module διότι δεν παρουσιάζουν ενδιαφέρον για τους χρήστες του κακόβουλου λογισμικού. Αντίθετα παρακάτω παρουσιάζονται διευθύνσεις ιστοσελίδων τραπεζικών ιδρυμάτων όπως και ο κακόβουλος κώδικας HTML ο οποίος πρόκειται να γίνει inject.



Εικόνα 4.9 Hesperbot Configuration File [18]

Μία εξαιρετικά ενδιαφέρουσα λειτουργία του Hesperbot είναι αυτή της εκμετάλλευσης των κινητών συσκευών των θυμάτων, ώστε να υπερπηδήσει τις σύγχρονες μεθόδους αυθεντικοποίησης των τραπεζικών ιδρυμάτων όπως τα Mobile Transaction Authentication Number. Συγκεκριμένα, το Hesperbot έχει την δυνατότητα, μέσω των web injects, να κατευθύνει τον χρήστη στην εγκατάσταση κακόβουλης εφαρμογής στο κινητό του τηλέφωνο, μάλιστα του ζητείται η εισαγωγή του τηλεφωνικού του αριθμού και του μοντέλου του τηλεφώνου για να του αποσταλεί στην συνέχεια link με την εφαρμογή που υποτιθέμενα χρειάζεται για να εκτελέσει τραπεζικές συναλλαγές. Οι εφαρμογές αυτές έχουν παρόμοια λειτουργία στα λειτουργικά συστήματα που υποστηρίζουν, χρησιμοποιούν μια διαδικασία ενεργοποίησης με activation number, ο οποίος παρέχεται μέσω του web injected browser του χρήστη και εν συνεχεία του ζητείται από την εφαρμογή του τηλεφώνου. Η εφαρμογή τηλεφώνου, μετά την εισαγωγή του activation number, παρέχει έναν επιπλέον αριθμό μέσω σχετικού αλγορίθμου ο οποίος ζητείται από τον browser, έτσι ώστε να ταυτοποιηθεί ότι ο ίδιος χρήστης έχει εγκαταστήσει το κακόβουλο λογισμικό και στο τηλέφωνό του. Η κακόβουλη εφαρμογή έχει την δυνατότητα από εκεί και πέρα να προωθεί

οποιοδήποτε SMS λαμβάνεται από το τηλέφωνο στον κακόβουλο χρήστη, έτσι ώστε να έχει πρόσβαση σε πιθανά mTAN τα οποία αποστέλλονται από τραπεζικό ίδρυμα. Τέλος, η κακόβουλη εφαρμογή επιτρέπει τον απομακρυσμένο έλεγχο της συσκευής μέσω της αποστολής εντολών με SMS.

Μια ακόμη λειτουργία του Hesperbot είναι η δυνατότητα keylogging. Αυτό επιτυγχάνεται με την ενσωμάτωση των διεργασιών GetMessage και TranslateMessage στο αρχείο user32.dll. Στην συνέχεια τα δεδομένα εγγράφονται σε ένα log file μαζί με το όνομα της διεργασίας που τα υπέκλεψε και αποστέλλονται στον C&C server.

Η λειτουργία δημιουργίας screenshots και εγγραφής video αναφέρθηκε προηγουμένως ότι εκτελείται από το module httpi, εφόσον περιλαμβάνεται στο configuration file. Δεν αποτελεί μια πρωτότυπη λειτουργία καθώς παρέχεται και από άλλα ήδη κακόβουλου λογισμικού για την παροχή πληροφοριών σχετικά με το τι συμβαίνει στην οθόνη του θύματος. Στην συγκεκριμένη περίπτωση χρησιμοποιεί τις διεργασίες AVIFileCreateStream, AVIFileMakeCompressedStream, AVIStreamWrite, τις οποίες ενσωματώνει στο αρχείο Avifil32.dll για το video ενώ χρησιμοποιεί τις BitBit και GetDIbits τις οποίες ενσωματώνει στο Gdi32.dll για τα screenshots.

Τέλος θα πρέπει να αναφερθεί και η δυνατότητα του Hesperbot να εγκαταστήσει κρυφά έναν VNC server στο μολυσμένο μηχάνημα, δίνοντας την δυνατότητα απομακρυσμένου ελέγχου ταυτόχρονα με τον νόμιμο χρήστη, χωρίς μάλιστα να γίνεται αντιληπτό. Η συγκεκριμένη λειτουργία δίνει επίσης στον κακόβουλο χρήστη δικαιώματα εκτέλεσης των browsers, που βρίσκονται εγκατεστημένοι στο μολυσμένο σύστημα, παρέχοντας του πρόσβαση σε όλα τα δεδομένα των browsers.

Το Hesperbot παρουσιάζει μια άλλη άποψη των τεχνικών δυνατοτήτων των σύγχρονων κακόβουλων λογισμικών, παρ' όλες τις ομοιότητες του με παλαιότερα λογισμικά, καθιστώντας το ιδιαίτερα ικανό και επικίνδυνο.

4.3 Buhtrap

Το Buhtrap είναι ένα νέο κακόβουλο λογισμικό, το οποίο παρότι είναι ενεργό από το 2014 υπήρξε έξαρση στην χρήση του κατά την διάρκεια του 2015 και του 2016 [19]. Σε αντίθεση με άλλα κακόβουλα λογισμικά, το συγκεκριμένο δεν στοχεύει πελάτες τραπεζικών ιδρυμάτων αλλά την υποδομή των ίδιων των ιδρυμάτων και συγκεκριμένα τραπεζικών ιδρυμάτων που εδρεύουν στη Ρωσία και την Ουκρανία. Η δυνατότητα του να εξαπλώνεται εντός των δικτύων των τραπεζικών ιδρυμάτων έχει ως αποτέλεσμα σημαντικά κομμάτια της υποδομής τους να παραμένουν ανενεργά έως ότου απομακρυνθεί το κακόβουλο λογισμικό, επιβάλλοντας επιπλέον οικονομικές απώλειες και καθυστερήσεις στις συναλλαγές. Συγκεκριμένα, από τον Αύγουστο του 2015 μέχρι τον Φεβρουάριο του 2016, αποδείχθηκε ότι χρησιμοποιήθηκε για 16 πετυχημένες επιθέσεις ενάντια σε ρωσικά τραπεζικά ιδρύματα με αποτέλεσμα την απώλεια 1.8δις ρουβλιών. [20]

Ο κύριος τρόπος εξάπλωσης του είναι τα phishing emails, τα οποία εμφανίζονται να προέρχονται από κάποιο τραπεζικό ίδρυμα. Στα συγκεκριμένα emails υπάρχει συνημμένο αρχείο MS Office, το οποίο περιέχει μακροεντολές και οδηγίες για την ενεργοποίηση των μακροεντολών. Σε περίπτωση που ο χρήστης ακολουθήσει τις οδηγίες ένα script εγκαθίσταται στο σύστημα. Το συγκεκριμένο script έχει την ικανότητα να ελέγχει αν το σύστημα έχει αποθηκεμένα links από συστήματα online banking, τραπεζικό λογισμικό εγκατεστημένο ή επισκέψεις σε τραπεζικά ιδρύματα αποθηκευμένες στο ιστορικό του browser. Εφόσον ανακαλύψει κάτι από τα προηγούμενα τότε προχωράει στο κατέβασμα του κακόβουλου λογισμικού και στη εγκατάσταση του.

Μια άλλη μέθοδος εξάπλωσης είναι αυτή με χρήση exploit kit, κατά την οποία οι χρήστες γίνονται redirect από ιστοσελίδες, οι οποίες έχουν μολυνθεί με κακόβουλο λογισμικό, σε servers οι οποίοι φιλοξενούν exploit kit, το οποίο εκμεταλλευόμενο ευπάθειες σε browsers κατεβάζει και εγκαθιστά το κακόβουλο λογισμικό.

Η τελευταία και ίσως πλέον επικίνδυνη μέθοδος εξάπλωσης του Buhtrap είναι με την χρήση νόμιμου λογισμικού. Τον Οκτώβρη του 2016 αποκαλύφθηκε ότι η ιστοσελίδα της εταιρείας Ammyy, η οποία αναπτύσσει λογισμικό απομακρυσμένου ελέγχου, είχε μολυνθεί

και το λογισμικό που είχαν την δυνατότητα να κατεβάσουν οι χρήστες περιείχε το Buhtrap. [19]

Ειδικότερα στην χρήση του Buhtrap ενάντια σε τραπεζικά ιδρύματα, οι χρήστες του προέβαιναν στις παρακάτω ενέργειες:

- Μετά την αρχική εγκατάσταση σε σύστημα τραπεζικού ιδρύματος ενεργοποιούσαν λογισμικό απομακρυσμένου ελέγχου για την εκτέλεση του module, το οποίο ήταν υπεύθυνο για την εξάπλωση του κακόβουλου λογισμικού εντός του δικτύου

- Συνέλεγαν στοιχεία αυθεντικοποίησης των domain accounts

- Το κακόβουλο λογισμικό έκανε έρευνα για το τραπεζικό λογισμικό AWS CBC (Automated Working Station of the Central Bank Client)

- Έπαιρνε το έλεγχο του AWS CBC και αντικαθιστούσε έγγραφα σχετικά με πληρωμές προς την Κεντρική Τράπεζα, τα οποία και επεξεργαζόταν

- Απενεργοποιούσε τα μολυσμένα συστήματα, ώστε να κάνει δυσκολότερη την συλλογή αποδεικτικών στοιχείων

4.3.1 Τρόπος Λειτουργίας

Πέρα από την εισαγωγή του Buhtrap σε συστήματα μέσω phishing emails, ιδιαίτερο ενδιαφέρον παρουσιάζει η μέθοδος με χρήση exploit kit. Για αυτή την μέθοδο χρησιμοποιούνται websites, τα οποία έχουν αλλοιωθεί με στόχο οι επισκέπτες να κατεβάζουν κακόβουλο λογισμικό χωρίς να το γνωρίζουν. Τα sites τα οποία χρησιμοποιήθηκαν για τις επιθέσεις σε Ρωσία και Ουκρανία ήταν το eurolab.ua, ένα δημοφιλές website για θέματα υγείας με πάνω από μισό εκατομμύριο επισκέπτες από Ουκρανία και Ρωσία τον μήνα. [20] Από το eurolab.ua, οι χρήστες ανακατευθύνονταν στην ιστοσελίδα rozhlas.site, η οποία περιείχε ένα browser exploit. Το συγκεκριμένο exploit ήταν σε θέση να επηρεάσει τον Microsoft Internet Explorer στις εκδόσεις 9 εως 11. Μετά την επιτυχημένη εκμετάλλευση της ευπάθειας του browser, το Buhtrap έτρεχε ένα

ενσωματωμένο Powershell script, το οποίο στην συνέχεια κατέβαζε το κακόβουλο λογισμικό.

Αξίζει εδώ να σημειωθεί ότι το Buhtrap λειτουργεί σε 2 στάδια με το πρώτο να αφορά τον έλεγχο του συστήματος για το εάν αποτελεί επιθυμητό στόχο και το δεύτερο στάδιο να αφορά την κύρια επίθεση κατά του συστήματος.

4.3.1.1 Πρώτο Στάδιο

Κατά το πρώτο στάδιο λειτουργίας, το Buhtrap έχει ως κύριο σκοπό τον έλεγχο του συστήματος. Για να το επιτύχει αυτό ελέγχει για την ύπαρξη τραπεζικού λογισμικού και ιστορικού browser και καταγράφει τις πληροφορίες στα αρχεία c:\Loginfo.txt και C:\WINDOWS\Debug\UserMode\userenv.txt. Εάν οι έλεγχοι του είναι επιτυχημένοι, τότε προχωράει στο κατέβασμα του αρχείου το οποίο ενεργοποιεί το δεύτερο στάδιο ενεργειών από το link hxxp://rozhlas[.]site/news/business/debug.bin. Σε περίπτωση που το μολυσμένο σύστημα δεν παρουσιάζει ενδιαφέρον, ως στόχος, τότε κατεβάζει ένα αρχείο με καλοήθες περιεχόμενο από το hxxp://rozhlas[.]site/news/business/release.bin, το οποίο έχει ως σκοπό την αποτροπή της ανάλυσης του κακόβουλου λογισμικού [20]. Το κακόβουλο λογισμικό αποθηκεύεται στο %appdata%\..\ssl_bapi.exe ή στο %tmp%\ssl_bapi.exe ανάλογα την έκδοση του λειτουργικού συστήματος. Από την ονομασία των αρχείων, εύκολα καταλαβαίνουμε ότι προσπαθεί να χρησιμοποιήσει κάποια πλαστά πιστοποιητικά ασφαλείας για να υπερπηδήσει πιθανούς ελέγχους.

Στην συνέχεια, διενεργείται ο έλεγχος του συστήματος για το εάν αποτελεί καλό στόχο. Η διαδικασία αυτή ξεκινάει με το WMI query “SELECT Name FROM Win32_Process”, ώστε το κακόβουλο λογισμικό να διαπιστώσει τις διεργασίες που εκτελούνται στο σύστημα. Εάν το συγκεκριμένο query αποτύχει τότε πραγματοποιεί και πάλι έλεγχο χρησιμοποιώντας τα APIs Process32First και Process32Next. Στην Εικόνα 4.9 βλέπουμε το WMI query.

```
if ( byte_4086B5 )
{
  v24 = 0;
  v4 = sub_4033B9((int)"SELECT Name FROM Win32_Process", 30);
  if ( (unsigned __int8)sub_4039AD((LPCSTR)v4, (int)&v24) && v24 > 0 )
  {
    v5 = strtok_s(0, ",", &Context);
    v23 = v5;
    while ( 1 )
    {
      v6 = 0;
      v22 = 0;
      if ( v1 )
        break;
LABEL_10:
      v5 = strtok_s(0, ",", &Context);
      v23 = v5;
      if ( !v5 )
        goto LABEL_18;
    }
    while ( lstrcmpi(v5, *(&v26 + v6)) )
    {
      v5 = v23;
      v6 = v22 + 1;
      v22 = v6;
      if ( v6 >= v1 )
        goto LABEL_10;
    }
    v21 = 1;
LABEL_18:
    free(0);
  }
}
```

Εικόνα 4.10 Buhtrap WMI query [20]

Οι διεργασίες για τις οποίες αναζητεί το κακόβουλο λογισμικό σχετίζονται με τραπεζικά ιδρύματα της Ρωσίας και απεικονίζονται στην Εικόνα 4.10 παρακάτω.

ip-client.exe	pkimonitor.exe	BC_Loader.exe	CbShell.exe	Bankline.EXE
prclient.exe	pmodule.exe	Client2008.exe	clb.exe	GeminiClientStation.exe
rclient.exe	pn.exe	IbcRemote31.exe	CliBank.exe	_ClientBank.exe
saclient.exe	postmove.exe	_ftcgpk.exe	CliBankOnlineEn.exe	ISClient.exe
SRCLBClient.exe	productprototype.exe	scardsvr.exe	CliBankOnlineRu.exe	cws.exe
twawebclient.exe	quickpay.exe	CL_1070002.exe	CliBankOnlineUa.exe	CLBANK.EXE
vegaClient.exe	rclaunch.exe	intpro.exe	client2.exe	IMBLink32.exe
dsstart.exe	retail.exe	UpMaster.exe	client6.exe	cbsmain.dll
dtpaydesk.exe	retail32.exe	SGBClient.exe	clientbk.exe	GpbClientSftcws.exe
eelclnt.exe	translink.exe	el_cli.exe	clntstr.exe	Run.exe
elbank.exe	unistream.exe	MWClient32.exe	clntw32.exe	SGBClient.exe
etprops.exe	uralprom.exe	Adirect.exe	contactng.exe	sx_Doc_ni.exe
eTSrv.exe	w32mkde.exe	Bclient.exe	Core.exe	icb_c.exe
ibconsole.exe	wclnt.exe	bc.exe	cshell.exe	Client32.exe
kb_cli.exe	wfinist.exe	ant.exe	cyberterm.exe	BankCl.exe
KLBS.exe	winpost.exe	arm.exe	client.exe	ICLTransportSystem.exe
KlientBnk.exe	wupostagent.exe	arm_mt.exe	cncclient.exe	GPBClient.exe
lfcpaymentais.exe	Zvit1DF.exe	ARMSH95.EXE	bbclient.exe	CLMAIN.exe
loadmain.exe	budget.exe	asbank_lite.exe	EximClient.exe	ONCBCLI.exe
lpbos.exe	CB.exe	bank.exe	fcclient.exe	CLBank3.exe
mebiusbankxp.exe	cb193w.exe	bank32.exe	iscc.exe	rmclient.exe
mmbank.exe	cbank.exe	bbms.exe	kabinet.exe	FcolseOW.exe
pcbank.exe	cbmain.ex	bk.exe	SrCLBStart.exe	RkcLoader.exe
pinpayr.exe	CBSMAIN.exe	BK_KW32.EXE	srcbclient.exe	uarm.exe
Pionner.exe		bnk.exe	Upp_4.exe	nlnotes.exe

Εικόνα 4.11 Τραπεζικές διεργασίες που αναζητεί το Buhtrap [21]

Στην συνέχεια το κακόβουλο λογισμικό ελέγχει για εγκατεστημένο λογισμικό το οποίο έχει σχέση με τραπεζικά ιδρύματα. Αυτό επιτυγχάνεται με τον κώδικα που παρουσιάζεται στην Εικόνα 4.11.

```
v21 = func_StringDecrypt((int)"%PROFILE%", 9);
v22 = func_StringDecrypt((int)"iBank2", 6);
v23 = func_StringDecrypt((int)"%APPDATA%", 9);
v24 = func_StringDecrypt((int)"amicon,bifit,*bss,*ibank", 24);
v25 = func_StringDecrypt((int)"%PROGRAMFILES32%", 16);
v26 = func_StringDecrypt(
        (int)"*gpb,inist,mdm,bifit,Aladdin,Amicon,*bss,Signal-COM,iBank2,*\\bc.exe,*\\*\\intpro.exe,*cft,agava,*
        112);
v27 = func_StringDecrypt((int)"%PROGRAMFILES64%", 16);
v28 = func_StringDecrypt(
        (int)"*gpb,inist,mdm,bifit,Aladdin,Amicon,*bss,Signal-COM,iBank2,*\\bc.exe,*\\*\\intpro.exe,*cft,agava,*
        112);
v29 = func_StringDecrypt((int)"%SYSTEMDRIVE%", 13);
v30 = func_StringDecrypt((int)"*SFT,*Agava,*Clnt,*CLUNION.0QT,*5NT,*BS,*ELBA,*Bank,ICB_C,*sped,*gpb", 68);
v31 = func_StringDecrypt((int)"%DESKTOP%", 9);
v32 = func_StringDecrypt((int)"*ELBA,*ELBRUS", 13);
if ( fun_EnumFiles((LPCSTR *)&v21)
```

Εικόνα 4.12 Έλεγχος λογισμικού τραπεζικών ιδρυμάτων από το Buhtrap [20]

Όπως φαίνεται στην Εικόνα 4.12 η αναζήτηση γίνεται σε συγκεκριμένες τοποθεσίες και για συγκεκριμένα λογισμικά πράγμα που αποδεικνύει την εκ προτέρων γνώση του τραπεζικού συστήματος.

```
%PROFILE% : “iBank2”
%APPDATA% : “amicon,bifit,*bss,*ibank”
%PROGRAMFILES32% : “*gpb,inist,mdm,bifit,Aladdin,Amicon,*bss,Signal-
COM,iBank2,*\\bc.exe,*\\*\\intpro.exe,*cft,agava,*R-Style,*AKB Perm”
%PROGRAMFILES64% : “*gpb,inist,mdm,bifit,Aladdin,Amicon,*bss,Signal-
COM,iBank2,*\\bc.exe,*\\*\\intpro.exe,*cft,agava,*R-Style,*AKB Perm”
%SYSTEMDRIVE% : “*SFT,*Agava,*Clnt,*CLUNION.0QT,*5NT,*BS,*ELBA,*Bank,ICB_C,*sped,*gpb”
%Desktop%: “*ELBA,*ELBRUS”
```

Εικόνα 4.13 Πιθανές τοποθεσίες τραπεζικών εφαρμογών [20]

Το επόμενο βήμα του κακόβουλου λογισμικού είναι η αναζήτηση στην cache των browsers του μολυσμένου συστήματος. Αυτό το επιτυγχάνει χρησιμοποιώντας το API FindFirstUrlCacheAntryA μέσω του οποίου κάνει αναζήτηση στις παρακάτω τοποθεσίες, ερευνώντας για συγκεκριμένα strings τα οποία σχετίζονται με ρωσικές τράπεζες, όπως αυτά στον Πίνακα 4.2 παρακάτω:

Strings σχετιζόμενα με τράπεζες	Τοποθεσίες έρευνας
ICPortalSSL	%localappdata%\Google\Chrome\User Data\Default\History
isfront.priovtb.com	%appdata%\Google\Chrome\User Data\Default\History
PortalSSL	%appdata%\Mozilla\Firefox\Profiles
beta.mcb.ru	%appdata%\Opera\Opera\global_history.dat
ibank	
ibrs	
iclient	
e-plat.mdmbank.com	
sberweb.zubsb.ru	
ibc	
elbrus	
i-elba	
clbank.minbank.ru	
chelindbank.ru/online/	
uwagb	
wwwbank	
dbo	
ib.	

Πίνακας 4.2 Strings τραπεζών και τοποθεσίες browser cache [20]

4.3.1.2 Δεύτερο Στάδιο

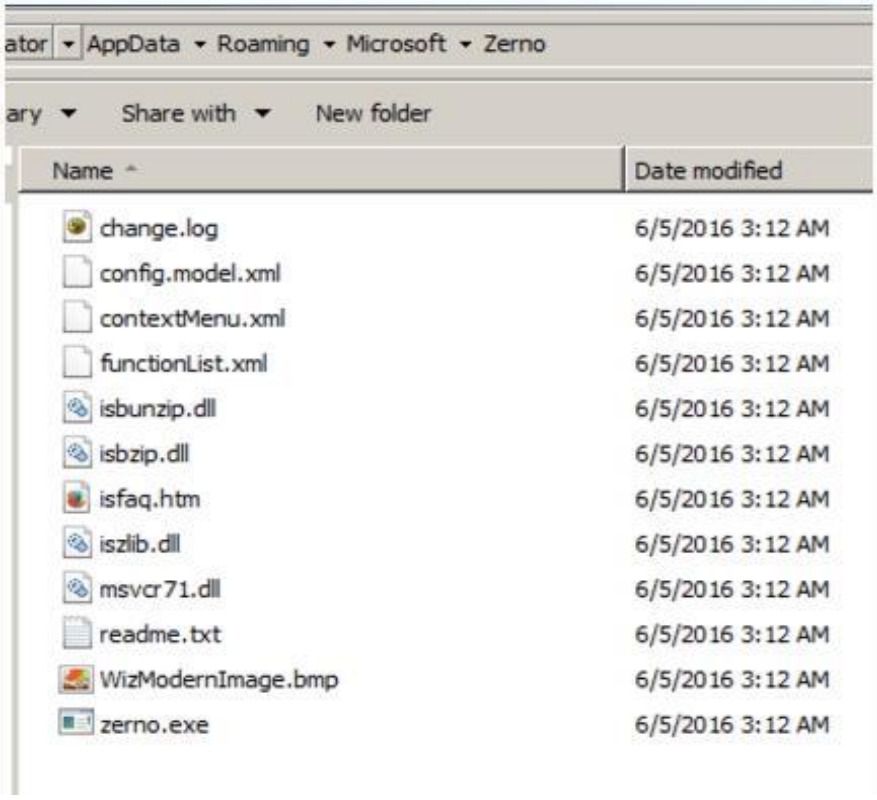
Κατά το δεύτερο στάδιο λειτουργίας το Buhtrap, εφόσον ικανοποιηθούν επιτυχώς οι συνθήκες λειτουργίας του πρώτου σταδίου κατεβάζει ένα αρχείο compiled με το NSIS (Nullsoft Scriptable Install System). Χρησιμοποιώντας το NSIS, το οποίο είναι ένα open source λογισμικό που χρησιμοποιείται ευρέως, προσπαθεί να αποφύγει τον εντοπισμό από κάποιο λογισμικό Antivirus. Το αρχείο αυτό επίσης φέρει ψηφιακή υπογραφή και περιλαμβάνει ιδιότητες από προηγούμενες εκδόσεις, σε μια προσπάθεια να φαίνεται όσο το δυνατό πιο ακίνδυνο. [22]

Εντός του NSIS αρχείου περιλαμβάνεται ένα συμπιεσμένο 7zip αρχείο, το οποίο είναι και προστατευμένο με password το οποίο περιέχεται στο NSIS αρχείο. Εντός του 7zip υπάρχει σχετική εντολή για την αποσυμπίεση των περιεχομένων του και την αλλαγή του timestamp του. Στην συνέχεια ελέγχει εάν η γλώσσα του μολυσμένου συστήματος είναι η ρωσική μέσω του API GetSystemDefaultLangID, εάν δεν είναι η ρωσική σταματά την

λειτουργία του. Εφόσον επιβεβαιώσει όμως, την ρωσική γλώσσα ως την γλώσσα συστήματος εκτελεί τις παρακάτω εντολές και διεργασίες:

- **ATTRIB -H -S -R "C:\USERS\ADMINISTRATOR\APPDATA\ROAMING\MICROSOFT\ZERNO"**
- **7ZA.EXE X -P2DP9ENV5BK INSTALL.DAT DEV2055.TMP -AOA**
- **7ZA.EXE X -P2DP9ENV5BK DEV2055.TMP -AOA -O8992023.TMP**
- **7ZA.EXE X -P2DP9ENV5BK INSTALL.DAT FILETOUCH.EXE -AOA**
- **C:\USERS\ADMINISTRATOR\APPDATA\ROAMING\MICROSOFT\ZERNO\ZERNO.EXE**

Όπως φαίνεται και παραπάνω το κακόβουλο λογισμικό αποθηκεύεται στον φάκελο %AppData%\Microsoft\Zerno μετά την αποσυμπίεση του και εκτελείται από το αρχείο zerno.exe. Τα αρχεία τα οποία περιέχονται στον φάκελο Zerno παρουσιάζονται στην Εικόνα 4.13.



Εικόνα 4.14 Τοποθεσία αρχείων του buhtrap [22]

Από αυτά τα αρχεία μόνο το zerno.exe και msvcr71.dll είναι κακόβουλα, τα υπόλοιπα είναι μέρη του λογισμικού Notepad++ και έχουν σκοπό την παρεμπόδιση αποκάλυψης του κακόβουλου λογισμικού. Τέλος δημιουργείται και μια συντόμευση του zerno.exe στον φάκελο startup του μολυσμένου συστήματος, ώστε το κακόβουλο λογισμικό να εκτελείται σε κάθε εκκίνηση του συστήματος.

Η εκτέλεση του zerno.exe έχει ως αποτέλεσμα την κλήση της βιβλιοθήκης msvcr71.dll, η οποία πραγματοποιεί όλη την υπόλοιπη κακόβουλη λειτουργία. Μια από αυτές είναι αυτή του keylogger, η διεργασία του keylogger δημιουργεί ένα αρχείο με την ονομασία uninstall.log μέσα στον φάκελο %temp% στο οποίο καταγράφει όλα τα δεδομένα. Στην Εικόνα 4.14 φαίνεται η υλοποίηση του keylogger.

```
while ( !RegisterClassExW(&v10) );
if ( CreateWindowExW(0, v10.lpszClassName, 0, 0, 0, 0, 0, 0, HWND_MESSAGE, 0, v10.hInstance, 0) )
{
  v6 = (CHAR *)GlobalAlloc(0x40u, 0x1000u);
  wMsgFilterMax = sub_10015DFC(a1, 0);
  sub_10015BA1((int)&v12, (int)&wMsgFilterMax);
  sub_10015D5E(&v13, 64, "%Y-%m-%d %X", (int)&v12);
  nNumberOfBytesToWrite = wsprintfA(v6, "%s\t%s: entering msg loop\n", &v13, "KeyLoggerThread@767");
  OutputDebugStringA(v6);
  GetTempPathA(0x104u, &Buffer);
  lstrcatA(&Buffer, "uninstall.log");
  v7 = CreateFileA(&Buffer, 4u, 1u, 0, 4u, 6u, 0);
  WriteFile(v7, v6, nNumberOfBytesToWrite, &nNumberOfBytesToWrite, 0);
  CloseHandle(v7);
  GlobalFree(v6);
  for ( result = (HGLOBAL)GetMessageW(&Msg, 0, 0, 0); (signed int)result > 0; result = (HGLOBAL)GetMessageW(
                                                                                 &Msg,
                                                                                 0,
                                                                                 0,
                                                                                 0) )
  {
    TranslateMessage(&Msg);
    DispatchMessageW(&Msg);
```

Εικόνα 4.15 Keylogger του Buhtrap [22]

Η άλλη λειτουργία του msvcr71.dll είναι αυτή του Smart Card reader, για την οποία χρησιμοποιεί το API WinSCard.dll για να διαπιστώσει την ύπαρξη ή όχι Smart Cards στο σύστημα και να καταγράψει την κατάσταση του στο αρχείο uninstall.log.

Μία ακόμα διαθέσιμη λειτουργία είναι αυτή του κατεβάσματος επιπλέον κακόβουλου λογισμικού από τον Command & Control Server, με το μεγάλο πλεονέκτημα ότι μπορεί να το πραγματοποιήσει και χωρίς να χρησιμοποιήσει τον δίσκο του συστήματος αλλά φορτώνοντας το κακόβουλο λογισμικό απευθείας στην μνήμη. Στην Εικόνα 4.15 βλέπουμε

πως επιτυγχάνεται το κατέβασμα του κακόβουλου λογισμικού στον δίσκο ενώ στην Εικόνα 4.16 βλέπουμε τον κώδικα της φόρτωσης του κακόβουλου λογισμικού στην μνήμη.

Εικόνα 4.16 Κατέβασμα του buhtrap στον δίσκο [22]

Εικόνα 4.17 Φόρτωση του buhtrap στην μνήμη [22]

Έρευνα της διεύθυνσης IP του site rozhlas.site, το οποίο χρησιμοποιεί το κακόβουλο λογισμικό Buhtrap έδειξε τα αποτελέσματα που φαίνονται παρακάτω στην Εικόνα 4.17.

Domain	Last Resolved
getadobe.org	5/10/2016
chromelabs.org	5/13/2016
adobelabs.org	5/14/2016
canvaslabs.org	5/22/2016
57569b378f3fb.archive.getadobe.org	6/7/2016
chrome.services	7/2/2016
get.adobelabs.org	7/2/2016
safechrome.services	7/11/2016
www.safechrome.services	7/28/2016
cdn.lidovky.site	8/9/2016
rozhlas.site	8/17/2016
getcanvas.org	9/14/2016
medioca-room02.org	9/28/2016

Εικόνα 4.18 IP που χρησιμοποιούνται από το Buhtrap [20]

Από τα αποτελέσματα της Εικόνας 4.17 συμπεραίνουμε ότι το κακόβουλο λογισμικό Buhtrap ήταν σε χρήση κατά το μεγαλύτερο μέρος του 2016, όπως επίσης ότι χρησιμοποιεί παραλλαγές γνωστών ιστοσελίδων στην προσπάθεια του να αποκρύψει την πραγματική του ιδιότητα.

Το Buhtrap γενικά είναι κακόβουλο λογισμικό, το οποίο χρησιμοποιεί ψηφιακά πιστοποιητικά για να αποφύγει τυχόν ελέγχους, σε συνδυασμό με χρήση παραλλαγμένων

γνωστών ιστοσελίδων και open source installer, με προστατευμένο με κωδικό περιεχόμενο, ενώ επιλέγει προσεκτικά τα συστήματα-θύματα του καταφέρνοντας να μην γίνεται ευρέως αντιληπτό. Ο συνδυασμός αυτός είναι που το καθιστά πολύ επικίνδυνο ειδικά για τα τραπεζικά ιδρύματα.

4.4 Corkow

Το κακόβουλο λογισμικό Corkow βρίσκεται σε κυκλοφορία από το 2011 και είναι συνεχώς εξελισσόμενο και μεταδιδόμενο μέχρι και σήμερα. Είναι ρωσικής κατασκευής και αποτελεί μέλος της ευρύτερης οικογένειας των κακόβουλων λογισμικών, τα οποία στοχεύουν τραπεζικές συναλλαγές. Μοιράζεται αρκετά κοινά χαρακτηριστικά με κακόβουλα λογισμικά, τα οποία αναλύσαμε στην παρούσα διατριβή (π.χ. Zeus, Hesperbot), όπως την καταγραφή των smart cards που υπάρχουν στο σύστημα και την στόχευση συγκεκριμένων τραπεζικών ιστοσελίδων και λογισμικών. Επίσης όπως και τα υπόλοιπα λογισμικά της οικογένειας είναι σπονδυλωτό αποτελούμενο από διαφορετικά modules και plugins τα οποία χρησιμοποιούνται ανάλογα την περίσταση.

Συγκεκριμένα το Corkow χρησιμοποιεί βιβλιοθήκες DLL (Dynamic Link Library) για τα modules του χρησιμοποιώντας ένα αρχείο DLL ως βάση στο οποίο ενσωματώνουν τις λειτουργίες τους τα υπόλοιπα. Πολλά από τα plugins είναι ήδη ενσωματωμένα στο βασικό αρχείο ενώ άλλα κατεβαίνουν από τον Command & Control Server, ενώ στην συνέχεια ενεργοποιούνται και ενσωματώνονται σε διεργασίες του μολυσμένου συστήματος από το βασικό αρχείο DLL. Στην Εικόνα 4.18 βλέπουμε τα διάφορα modules του Corkow και τις λειτουργίες τους. [23]

Module	Description
Core	Main module responsible for injecting other modules into corresponding processes and for C&C communication. Also takes screenshots, enumerates smart cards and can block applications from running.
MON	Collects information about the system (list of running processes, user name, SID, last user input) and sends it to the C&C.
FG	Web-injections and form-grabbing module based on the leaked Zeus source-code. Corkow mainly uses the form-grabbing functionality to capture data.
KLG	Keylogger
HVNC	Hidden VNC connection that enables the attacker to connect remotely to the victim's machine.
PG	PuTTY logger for the `putty.exe` process. This is able to capture server logon credentials, which are valuable to cybercriminals.
PONY	Launches the "3rd party" universal password stealer Pony. ESET detects this trojan as Win32/PSW.Fareit.
IB2	Targets iBank2, a Russian banking application.
SBRF	Targets standalone Windows banking applications of Sberbank, the third-largest bank in Europe.
DC	Searches for finance-related text strings in browser history, installed and last used applications and running processes.

Εικόνα 4.19 Τα modules του Corkow [23]

Παρόλο που το βασικό αρχείο DLL του Corkow είναι υπεύθυνο για το κατέβασμα και την λειτουργία των plugins, αυτά περιλαμβάνουν στα δεδομένα τους την διεύθυνση του C&C server καθώς και την δυνατότητα την συλλογής και αποστολής των πληροφοριών που υποκλέπτουν αυτόνομα. Στην Εικόνα 4.18 φαίνεται ότι το Corkow έχει όλες τις δυνατότητες ενός τυπικού κακόβουλου λογισμικού το οποίο στοχεύει τραπεζικές συναλλαγές, όπως keylogging και δυνατότητα απομακρυσμένου ελέγχου, επιπλέον όμως έχει και κάποιες εξειδικευμένες δυνατότητες οι οποίες αφορούν την στοχοποίηση συστημάτων ρωσικών τραπεζών αλλά και την συλλογή πληροφοριών σχετικά με τις οικονομικές συναλλαγές του θέματος.

4.4.1 Τρόπος Λειτουργίας

Το Corkow χρησιμοποιεί εκτελέσιμο αρχείο ως μέθοδο εισόδου σε ένα σύστημα, όταν το αρχείο αυτό εκτελεστεί τότε γίνεται αποκρυπτογράφηση του βασικού αρχείου DLL, το οποίο είναι ενσωματωμένο στο εκτελέσιμο αρχείο και καλείται η διεργασία DLLMain δίνοντας της ως παράμετρο τον φάκελο στον οποίο θα εγκατασταθεί το κακόβουλο λογισμικό. Η τοποθεσία εγκατάστασης εξαρτάται άμεσα από το αν ο user account στον οποίο εκτελείται είναι απλού χρήστη ή administrator.

Στην συνέχεια ενεργοποιείται το βασικό DLL αρχείο το οποίο αναζητεί ένα μη κακόβουλο αρχείο DLL για να το χρησιμοποιήσει ως host αποφεύγοντας έτσι την ανίχνευση. Η αναζήτηση γίνεται στον φάκελο %SystemRoot%\System32 και αφορά αρχεία DLL τα οποία δεν είναι προστατευμένα ως αρχεία συστήματος. Εφόσον βρεθεί το κατάλληλο αρχείο DLL τότε το DLL του Corkow κρυπτογραφείτε και ενσωματώνει την κρυπτογραφημένη έκδοση του στο DLL του μολυσμένου συστήματος. Για την αποκρυπτογράφηση και την εκτέλεση του, το βασικό DLL γράφει τις σχετικές πληροφορίες σαν μια νέα διεργασία εξόδου στο μολυσμένο DLL και αυτό με την σειρά του αποθηκεύεται στον φάκελο εγκατάστασης του κακόβουλου λογισμικού ενώ το αντίγραφο του στον φάκελο του συστήματος παραμένει αναλλοίωτο στην αρχική του θέση. Τέλος δημιουργείται μια εγγραφή στην registry, ώστε να μονιμοποιηθεί η εγκατάσταση του κακόβουλου λογισμικού στο σύστημα. Στην Εικόνα 4.19 βλέπουμε τους πιθανούς φακέλους εγκατάστασης του Corkow καθώς, τις ανάλογες εγγραφές στην registry και τα DLL που χρησιμοποιούνται. [23]

Path	Registry entry	DLL export
%CommonProgramFiles%\microsoft shared\DW*$random$*\	[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters] ServiceDll = *$path_to_malware$*	ServiceMain
%AppData%\DAO*$random$*\	[HKEY_CURRENT_USER\Software\Classes\CLSID\{35CEC8A3-2BE6-11D2-8773-92E220524153}\InprocServer32] (Default) = *$path_to_malware$*	DllGetClassObject
%AppData%\Microsoft Corporation\	[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] NvCplWow64 = *$path_to_rundll32.exe$ "$path_to_malware$"*, Control_RunDLL	Control_RunDLL

Εικόνα 4.20 Φάκελοι εγκατάστασης του Corkow [23]

Όπως προαναφέρθηκε το αρχικό DLL του Corkow εγγράφει τα δεδομένα του εντός αρχείου DLL του μολυσμένου συστήματος σε κρυπτογραφημένη μορφή, για αυτή την κρυπτογράφηση χρησιμοποιεί την μέθοδο XOR και το κλειδί που δημιουργείται εξαρτάται από τον σειριακό αριθμό του σκληρού δίσκου του συστήματος. Με αυτό τον τρόπο εξασφαλίζεται ότι το κακόβουλο λογισμικό δεν θα εκτελεστεί σε άλλο υπολογιστή όπως επίσης δυσκολεύει και η ανάλυση του.

Το βασικό module του Corkow είναι υπεύθυνο για την εξαγωγή και την ενσωμάτωση, των υπολοίπων modules τα οποία φέρει, στις κατάλληλες διεργασίες του μολυσμένου συστήματος, όπως επίσης και για την επικοινωνία με τον Command & Control server. Για την επικοινωνία του με τον C&C server περιέχει κάποια URL με τα οποία δοκιμάζει να συνδεθεί, τα αρχικά αιτήματα HTTP περιέχουν πληροφορίες σχετικές με το σύστημα, τις εκδόσεις των modules του κακόβουλου λογισμικού και τον αριθμό ID του. Στην συνέχεια, η επικοινωνία με τον C&C server κρυπτογραφείτε, με το κλειδί κρυπτογράφησης να συντίθεται από το URL του C&C server και το ID του κακόβουλου λογισμικού.

Μετά την επίτευξη επικοινωνίας μεταξύ του Corkow και του C&C server, ο δεύτερος απαντά με συγκεκριμένες εντολές όπως:

- Reboot
- Κατέβασε και εκτέλεσε συγκεκριμένο αρχείο
- Εκτέλεσε ενημέρωση του κακόβουλου λογισμικού
- Κατέβασε το configuration file ή επιπλέον modules
- Διέγραψε αρχεία του συστήματος και γράψε τυχαία δεδομένα στην θέση του
- Κάνε απεγκατάσταση ή αυτοκαταστρέψου

Όπως συμπεραίνουμε από τις παραπάνω εντολές το Corkow μπορεί να αποβεί καταστροφικό για ένα σύστημα, το οποίο έχει μολύνει, διαγράφοντας αρχεία κατά βούληση. Επίσης, η εντολή απεγκατάστασης μπορεί να σταλεί με συγκεκριμένες παραμέτρους έτσι ώστε να διαγράψει παράλληλα σημαντικά αρχεία του συστήματος αλλά και να επανεγγράψει το MBR (Master Boot Record) του συστήματος με τυχαία δεδομένα καθιστώντας το άχρηστο.

Το βασικό module έχει επίσης την δυνατότητα να λαμβάνει screenshots, να εμποδίζει την εκτέλεση συγκεκριμένων εφαρμογών και να προσμετρά τις smartcards, οι οποίες

υπάρχουν στο σύστημα. Η ικανότητα να εμποδίζει την εκτέλεση εφαρμογών καθορίζεται από τις ρυθμίσεις του module, γενικά όμως το Corkow ελέγχει συνεχώς τις διεργασίες που εκτελούνται και όταν συναντήσει την επιθυμητή ονομασία προσπαθεί να την τερματίσει. Η ύπαρξη της παραπάνω δυνατότητας έχει κυρίως σημασία για την αποτροπή του χρήστη από το να εκτελέσει οποιοδήποτε τραπεζικό λογισμικό θα του επιτρέψει τον έλεγχο των λογαριασμών του. Σε αντίθεση με άλλα λογισμικά το Corkcow δεν αλληλεπιδρά με τις Smart Cards παρά μόνο τις προσμετρά χρησιμοποιώντας το API SetupDI και ψάχνοντας για συγκεκριμένα ονόματα συσκευών.

Το Corkow στοχεύει συγκεκριμένα τραπεζικά λογισμικά, ιδιαίτερα όμως ο τρόπος με τον οποίο αλληλεπιδρά με το τραπεζικό λογισμικό iBank2 παρουσιάζει ενδιαφέρον. Το Corkow ενσωματώνει κακόβουλο κώδικα στην Java με την οποία είναι γραμμένο το iBank2 και για να το πετύχει αυτό αρχικά ενσωματώνει με την μέθοδο του inject κακόβουλο κώδικα Java σε κάθε διεργασία Java, η οποία εκκινεί στο σύστημα. Ο ενσωματωμένος κώδικας χρησιμοποιεί το Java Native Interface ώστε να τοποθετήσει έναν pointer στο Java Virtual Machine και με αυτό τον τρόπο να ενσωματώσει τον κακόβουλο κώδικα και εκεί. Ο κακόβουλος αυτός κώδικας σε Java έχει την δυνατότητα να υποκλέπτει το υπόλοιπο του τραπεζικού λογαριασμού του θύματος, να εξάγει screenshots αλλά και να αντιγράφει του κωδικούς αυθεντικοποίησης, τους οποίους χρησιμοποιεί το θύμα. Οι γλώσσες του λογισμικού iBank2 τις οποίες υποστηρίζει είναι τα Αγγλικά, τα Ρωσικά και τα Ουκρανικά.

Ένα ακόμα ενσωματωμένο module του Corkcow είναι το DC, η κύρια λειτουργία του οποίου είναι ο έλεγχος της δραστηριότητας του θύματος. Συγκεκριμένα ψάχνει για διεργασίες που εκτελούνται, το ιστορικό των browsers, το εγκατεστημένο λογισμικό και το λογισμικό που έχει χρησιμοποιηθεί πρόσφατα. Το αξιοπερίεργο στο συγκεκριμένο module είναι ότι δεν στέλνει όλες τις πληροφορίες που συλλέγει στον C&C server, αλλά μόνο αυτές οι οποίες αφορούν λογισμικό οικονομικών συναλλαγών, βάση συγκεκριμένης λίστας η οποία περιέχεται στο module. Παρακάτω στην Εικόνα 4.20 [23] βλέπουμε την λίστα με τα τραπεζικά ιδρύματα το λογισμικό των οποίων στοχεύει το Corkow.

iBank2	First Ukrainian International Bank (ПУМБ)	Interactive Brokers
WebMoney Keeper	AKB Privatbank	Ameritrade
Yandex.Money	Zuger Kantonalbank	Schwab
Sberbank	Credit Suisse	E*Trade
Alfa-Bank	CIM Banque	Fidelity
Contact NG	HSBC	5trade
Western Union	Hypo Landesbank Vorarlberg	**Digital currencies**
Xpress Money	Loyal Bank	Liberty Reserve
Trans-Fast	Valartis Bank	BTC-e
MoneyGram	Danske Bank	Mt.Gox
Promsvyazbank Cyprus	Jyske Bank	BitStamp
Avangard Bank	BEC	50BTC
Hellenic Bank	Raiffeisen	Bitcoin-Qt
Russian Commercial Bank Cyprus	Forum Bank	MultiBit
Alpha Bank Cyprus	**Electronic trading platforms, stock brokerages**	Electrum
Bank of Cyprus	Finam Direct II	Bitcoin Armory
Cyprus Popular Bank (Laiki)	Blackwood Pro	Litecoin
DBS Bank	Scottrade	**Other**
United Overseas Bank	ScottradeELITE	Google Play developer activity
OCBC Bank	MBT Desktop Pro	PuTTY
ABLV Bank	QuoteTracker	WinSCP
Baltikums Bank	eSignal	LightSpeed
Norvik Bank	Ensign	QIWI payment system
Snoras Bank	TraderBytes	*various unidentified PoS systems*
Rietumu Bank	ROX	

Εικόνα 4.21 Τράπεζες που στοχεύει το Corkow [23]

Το Corkow αν και αποτελεί ένα μείγμα λειτουργιών άλλων κακόβουλων λογισμικών, είναι εμπλουτισμένο με νέες ικανότητες και δεν παύει να αποτελεί έναν υπολογίσιμο κίνδυνο ειδικά για τα τραπεζικά περιβάλλοντα.

Κεφάλαιο 5

Σύγκριση Κακόβουλων Λογισμικών

5.1 Εισαγωγή

Απαραίτητη προϋπόθεση, ώστε να καταφέρουμε να ταξινομήσουμε τα κακόβουλα λογισμικά τα οποία αναλύσαμε στο προηγούμενο κεφάλαιο, είναι να συγκρίνουμε τον τρόπο λειτουργίας τους και τα εγγενή τους χαρακτηριστικά. Ξεκινώντας από τις μεθόδους εισόδου, τις οποίες χρησιμοποιούν, στο παρόν κεφάλαιο θα αντιπαραβάλουμε λεπτομερώς τον τρόπο λειτουργίας τους, ώστε να διευκολύνουμε την ταξινόμηση τους.

5.2 Μέθοδοι Εισόδου

Συγκεκριμένα ως προς την μέθοδο εισόδου στο σύστημα παρατηρήσαμε στις αναλύσεις ότι το κακόβουλο λογισμικό Zeus χρησιμοποιεί εκτελέσιμα αρχεία, τα οποία αποθηκεύει σε συγκεκριμένους φακέλους του συστήματος. Βάσει της μεθόδου κατασκευής του Zeus το εκτελέσιμο αρχείο είναι μοναδικό για κάθε χρήστη, καθώς περιλαμβάνει μοναδικό encryption key. Το εκτελέσιμο αρχείο στην συνέχεια συμπιέζεται και μετατρέπεται με την βοήθεια κάποιου packer, σε φαινομενικά ακίνδυνο τύπο αρχείου, με σκοπό να διευκολυνθεί η είσοδος του στο σύστημα. Στις τελευταίες εκδόσεις του Zeus το εκτελέσιμο αρχείο αντιγράφεται σε τυχαίο φάκελο εντός του Application Data και στην συνέχεια εισάγει στην registry το configuration file του, αφού πρώτα διενεργήσει έλεγχο συστήματος και συλλέξει κρίσιμες πληροφορίες.

Το κακόβουλο λογισμικό Hesperbot χρησιμοποιεί παρόμοια, αρχικά, μέθοδο μόλυνσης ενός συστήματος με το Zeus, βασιζόμενο στην είσοδο εκτελέσιμου αρχείου στο σύστημα. Στην συνέχεια όμως το αρχικό αρχείο κάνει inject κώδικα στην διεργασία explorer.exe και μέσω αυτής προχωρά στην εγκατάσταση των διαφόρων modules του και στην λήψη του configuration file από τον C&C server.

Σε αντίθεση με το Zeus και το Hesperbot το κακόβουλο λογισμικό Buhtrap χρησιμοποιεί εναλλακτικές μεθόδους εισόδου σε ένα σύστημα. Συγκεκριμένα χρησιμοποιεί μακροεντολές σε έγγραφα MS Office σε συνδυασμό με scipts, τα οποία ελέγχουν για πιθανή χρήση του συστήματος για τραπεζικούς σκοπούς αλλά και προχωρούν στο κατέβασμα και εγκατάσταση του κακόβουλου λογισμικού. Πέραν τούτου χρησιμοποιεί επίσης exploit kits, τα οποία εκμεταλλεύονται ευπάθειες σε browsers προκειμένου να κατεβάσουν και να εγκαταστήσουν τον Buhtrap. Όπως και στην περίπτωση των Zeus και Hesperbot χρησιμοποιείται και εδώ εκτελέσιμο αρχείο, το οποίο όμως έχει ως κύριο λόγο τον έλεγχο για το κατά πόσο το εν λόγω σύστημα χρησιμοποιείται για τραπεζικές συναλλαγές και όχι για την εξάπλωση του κακόβουλου λογισμικού στο σύστημα.

Τέλος το κακόβουλο λογισμικό Corkow χρησιμοποιεί και αυτό εκτελέσιμο αρχείο για την είσοδο στο σύστημα με την σημαντική διαφορά όμως της χρήσης αρχείων DLL για την εξάπλωση και λειτουργία του. Συγκεκριμένα, το εκτελέσιμο αρχείο αποκρυπτογραφεί το βασικό αρχείο DLL, το οποίο στην συνέχεια επιμολύνει κάποιο DLL του συστήματος.

Παρατηρούμε παραπάνω ότι τα κακόβουλα λογισμικά Zeus, Hesperbot και Corkow παρουσιάζουν σημαντικές διαφορές ως προς την είσοδο σε ένα σύστημα σε σχέση με το Buhtrap καθώς απαιτούν την χειροκίνητη εισαγωγή ενός εκτελέσιμου αρχείου στο σύστημα, ενώ το Buhtrap χρησιμοποιεί αυτοματοποιημένη διαδικασία μέσω scripts.

5.3 Μέθοδοι Εξάπλωσης

Όλα τα κακόβουλα λογισμικά τα οποία αναλύσαμε διαφέρουν ως προς τον τρόπο εξάπλωσης στο σύστημα μετά την αρχική εισαγωγή τους, χρησιμοποιώντας είτε εγγραφές

στην registry όπως το Zeus, είτε inject σε διεργασίες του συστήματος όπως το Hesperbot, είτε αρχεία DLL όπως το Corkow, είτε τέλος open source λογισμικό όπως το Buhtrap.

5.4 Τρόποι Λειτουργίας

Στις αναλύσεις κακόβουλων λογισμικών στο προηγούμενο κεφάλαιο παρατηρούμε κάποιες βασικές ομοιότητες σε ότι αφορά των τρόπο λειτουργίας τους. Συγκεκριμένα, η χρήση ενός βασικού αρχείου το οποίο με την σειρά του χρησιμοποιεί configuration file είναι μια λειτουργία την οποία συναντάμε στα κακόβουλα λογισμικά Zeus, Hesperbot και Corkow. Αντίθετα, το Buhtrap διαφέρει και πάλι ως προς την λειτουργία του, καθώς λόγω των προελέγχων που εκτελεί, της σπονδυλωτής δομής τους και της λειτουργίας του σε στάδια φαίνεται ότι δεν χρειάζεται να χρησιμοποιήσει κάποιο configuration file.

Μια ακόμα σημαντική ομοιότητα, ως προς την λειτουργία των κακόβουλων λογισμικών, είναι η χρήση APIs, τα οποία ενσωματώνονται στις διεργασίες του συστήματος και εκτελούν κακόβουλες ενέργειες. Χρήση APIs κάνει το Zeus, το Hesperbot, το Buhtrap αλλά και το Corkow. Παρατηρούμε εδώ ότι τα APIs αποτελούν ένα σημαντικό δομικό στοιχείο των κακόβουλων λογισμικών καθώς είναι αυτά τα οποία και εκτελούν τις κακόβουλες ενέργειες.

Τέλος ένα ακόμα χαρακτηριστικό στοιχείο λειτουργίας των κακόβουλων λογισμικών που αναλύθηκαν είναι η χρήση αρχείων DLL. Παρατηρούμε ότι τόσο το Zeus όσο και τα Hesperbot, Buhtrap και Corkow κάνουν ευρεία χρήση των αρχείων DLL είτε για να ενσωματώσουν τα διάφορα APIs είτε σαν βασικό αρχείο στην περίπτωση του Corkow.

5.5 Δυνατότητες Κακόβουλων Λογισμικών

Όπως αναφέρθηκε, τα σύγχρονα κακόβουλα λογισμικά έχουν πλέον την δυνατότητα να εναλλάσσουν τις δυνατότητες κακόβουλων ενεργειών τους, διατηρώντας τον βασικό

κομμάτι τους ίδιο και εναλλάσσοντας modules ή διαφοροποιώντας το configuration file, προσθέτοντας ή αφαιρώντας με αυτόν τον τρόπο δυνατότητες ανάλογα τους σκοπούς του χρήστη. Παρόλα αυτά δεν παύουν να υπάρχουν κάποιες εγγενείς δυνατότητες, τις οποίες και θα καταγράψουμε για τα κακόβουλα λογισμικά τα οποία παρουσιάσαμε παραπάνω.

Το κακόβουλο λογισμικό Zeus έχει ως κύρια αποστολή την υποκλοπή τραπεζικών στοιχείων και στοιχείων αυθεντικοποίησης. Ο σκοπός του επιτυγχάνεται με διάφορες μεθόδους μερικές από τις οποίες είναι:

- Η παρακολούθηση συγκεκριμένων ports
- Η μέθοδος web inject σε HTML
- Η χρήση API hooks σε διεργασίες του συστήματος

Από τα παραπάνω παρατηρούμε ότι οι δυνατότητες υποκλοπής στοιχείων είναι πρακτικά ανεξάντλητες και εξαρτώνται καθαρά από τις διαθέσεις του κακόβουλου χρήστη. Κύριος σκοπός του Zeus όμως παραμένει, σε κάθε περίπτωση, η υποκλοπή στοιχείων τα οποία μπορούν να χρησιμοποιηθούν σε απάτη συναλλαγών.

Το Hesperbot από την άλλη θεωρείται ένα ιδιαίτερα εξελιγμένο, ως προς τις δυνατότητες του, κακόβουλο λογισμικό το οποίο είναι ικανό να:

- Πραγματοποιήσει keystroke logging
- Δημιουργήσει screenshots
- Πραγματοποιήσει λήψη video
- Δημιουργήσει remote proxy
- Δημιουργήσει VNC server για απομακρυσμένο έλεγχο
- Υποκλέψει δεδομένα από το δίκτυο
- Πραγματοποιήσει HTML injection
- Εγκαταστήσει κακόβουλο λογισμικό σε mobile συσκευές μέσω των web injects
- Στοχεύσει συγκεκριμένα τραπεζικά ιδρύματα για HTML injection και network spoofing
- Εφαρμόσει Form grabbing

Σε αντίθεση με το Zeus, δεν χρησιμοποιεί τη μέθοδο Man-in-the-Browser αλλά ξεχωριστά plugins για την υποκλοπή στοιχείων από browsers. Η κατά βούληση χρήση τέτοιων plugins αλλά και οι περισσότερες δυνατότητες που αυτά του δίνουν είναι η κύρια διαφοροποίηση του με το Zeus.

Το Buhtrap ως νέο κακόβουλο λογισμικό εισάγει την καινοτομία της επιλεκτικής προσβολής συστημάτων, με σκοπό την μόλυνση συγκεκριμένων συστημάτων τα οποία ανήκουν σε τραπεζικά ιδρύματα. Ακόμα και η χρήση μακροεντολών του MS Office σε συνδυασμό με scripts είναι δυνατότητες που δεν συναντούμε στα υπόλοιπα λογισμικά που παρουσιάσαμε. Επιπλέον το Buhtrap δεν αποσκοπεί στην υποκλοπή μόνο τραπεζικών στοιχείων αλλά και domain accounts με σκοπό την πλήρη διείσδυση σε τραπεζικούς οργανισμούς, ενώ χρησιμοποιεί open source λογισμικό για να αποκρύψει την ταυτότητα του. Τα παραπάνω ενισχύουν την αποτελεσματικότητα του, παρόλα αυτά σε ότι αφορά την κύρια λειτουργία του, η οποία αφορά υποκλοπή τραπεζικών στοιχείων, χρησιμοποιεί παρόμοιες μεθόδους με τα υπόλοιπα κακόβουλα λογισμικά, οι οποίες αλλάζουν ανάλογα με τα modules τα οποία χρησιμοποιούνται.

Τέλος το Corkow ανήκει και αυτό στα κακόβουλα λογισμικά τα οποία στοχεύουν συγκεκριμένους τραπεζικούς οργανισμούς, με την βοήθεια όμως περισσότερο συμβατικών μεθόδων σε σχέση με το Buhtrap. Η δυνατότητα, η οποία το διαφοροποιεί από τα υπόλοιπα κακόβουλα λογισμικά τα οποία παρουσιάστηκαν, είναι αυτή του πλήρους ελέγχου των διεργασιών που εκτελούνται σε ένα σύστημα. Με αυτόν τον τρόπο, μπορεί και απαγορεύει την εκτέλεση διεργασιών, οι οποίες θα αποκαλύψουν την ύπαρξη του στο σύστημα ή θα προϊδεάσουν τον χρήστη για την τραπεζική απάτη που λαμβάνει χώρα. Ειδικότερα, η δυνατότητα αποτροπής εκτέλεσης τραπεζικού λογισμικού, όπως επίσης και η δυνατότητα αλλαγής των στοιχείων που παρουσιάζονται στον χρήστη από μια τραπεζική εφαρμογή, είναι αυτές οι οποίες θεωρούνται οι πλέον επικίνδυνες, σε συνδυασμό με την υποκλοπή των στοιχείων αυθεντικοποίησης του χρήστη, μέσω της μολυσμένης τραπεζικής εφαρμογής. Επίσης σημαντική είναι η δυνατότητα του να στέλνει στον κακόβουλο χρήστη μόνο τα στοιχεία εκείνα που συλλέγει και αφορούν αποκλειστικά οικονομικές συναλλαγές. Βέβαια με δεδομένο ότι το Corkow είναι modular, όπως και τα υπόλοιπα, κατανοούμε ότι οι δυνατότητες τους είναι πολύ περισσότερες και καταλαμβάνουν ένα ευρύ φάσμα.

Κεφάλαιο 6

Ταξινόμηση Κακόβουλων Λογισμικών και Συμπεράσματα

6.1 Μέθοδος Ταξινόμησης

Σε προηγούμενο κεφάλαιο δείξαμε την δυσκολία στην ονοματοδοσία και ταξινομία των κακόβουλων λογισμικών. Αυτή έγκειται κυρίως στην έλλειψη παγκοσμίως αποδεκτού κοινού προτύπου αλλά και στην δυσκολία ταξινόμησης των κακόβουλων λογισμικών βάσει της συμπεριφοράς τους. Η δυνατότητα εναλλαξιμότητας κακόβουλου κώδικα, όπως επίσης και το γεγονός ότι τα σύγχρονα κακόβουλα λογισμικά μπορούν να είναι modular και να χρησιμοποιούν διαφορετικά modules ανάλογα την περίσταση και το σκοπό τους, είναι το μεγαλύτερο πρόβλημα που καλούμαστε να αντιμετωπίσουμε σε μια ταξινομία κακόβουλων λογισμικών.

Πιθανή λύση στο συγκεκριμένο πρόβλημα έρχεται να δώσει η Joanna Rutkowska με την πρόταση της [16], η οποία υιοθετεί μια άλλη προσέγγιση στην ταξινόμηση κακόβουλου λογισμικού χρησιμοποιώντας μια παραλλαγή του ορισμού κακόβουλο λογισμικό, ως εξής:

«Κακόβουλο λογισμικό είναι ένα κομμάτι κώδικα το οποίο αλλάζει την συμπεριφορά του πυρήνα ενός λειτουργικού συστήματος ή μιας εφαρμογής χωρίς την άδεια του χρήστη με τέτοιον τρόπο ώστε να μην μπορεί να γίνει αντιληπτή.» [16]

Όπως παρατηρούμε, η φράση κλειδί στον παραπάνω ορισμό είναι η αλλαγή της συμπεριφοράς του λειτουργικού συστήματος ή κάποιας εφαρμογής. Βασιζόμενοι στο

αποτέλεσμα της δράσης και όχι στις ιδιότητες του ίδιου του κακόβουλου λογισμικού, η ταξινόμηση του στις προτεινόμενες από την Rutkowska κατηγορίες ή ακόμα και η οριοθέτηση νέων κατηγοριών κάνει πλέον την ταξινομία ευκολότερη και αποδοτικότερη. Για τους παραπάνω λόγους, στην παρούσα διατριβή θα επιχειρήσουμε την ταξινόμηση των κακόβουλων λογισμικών που αναλύθηκαν με χρήση της προτεινόμενης μεθόδου από την Rutkowska.

6.1.1 Ταξινόμηση και αντιμετώπιση

Η Rutkowska στην πρόταση της διαχωρίζει τα κακόβουλα λογισμικά σε 4 κατηγορίες-τύπους (Types) [16], οι οποίοι διαχωρίζονται ανάλογα με την επίδραση του κακόβουλου λογισμικού στο λειτουργικό σύστημα. Στην Type 0 ταξινομούνται τα κακόβολα λογισμικά τα οποία δεν αλληλοεπιδρούν με το λειτουργικό σύστημα με οποιαδήποτε άγνωστη μέθοδο, όπως επίσης και δεν αλλάζουν την συμπεριφορά του πυρήνα ή των διεργασιών του. Μπορούν ωστόσο να εκτελούν τις δικές τους αυτόνομες διεργασίες στο πλαίσιο του λειτουργικού συστήματος μέσω των οποίων εκτελούν τις κακόβουλες ενέργειες τους. Ο τύπος αυτός κακόβουλων λογισμικών είναι και ο πιο εύκολος να ανιχνευθεί με τις γνωστές μεθόδους ανίχνευσης όπως τον έλεγχο συμπεριφοράς, το sandboxing, τα AI heuristics ή τον έλεγχο βάση ψηφιακής υπογραφής. Λόγω της πολυπλοκότητας των κακόβουλων λογισμικών, τα οποία στοχεύουν τραπεζικές συναλλαγές και της ανάγκης εκτέλεσης πολυδιάστατων και πολύπλοκων ενεργειών ώστε να παρακάμψουν τα συστήματα ασφαλείας, η ταξινόμηση τους σε αυτή την κατηγορία παρατηρούμε ότι είναι εκ των πραγμάτων απίθανη.

Στην Type I [16] κατηγορία ταξινομούνται κατά την Rutkowska τα κακόβουλα λογισμικά τα οποία επιδρούν και αλλάζουν στοιχεία του λειτουργικού συστήματος τα οποία θεωρούνται σταθερά και αμετάβλητα. Τέτοια είναι τα εκτελέσιμα αρχεία, κομμάτια κώδικα στην μνήμη, ο κώδικας του BIOS η και η μνήμη των εγκατεστημένων στο σύστημα PCI συσκευών. Τα κακόβουλα λογισμικά τα οποία ανήκουν στην κατηγορία αυτή είναι δύσκολο να ανιχνευθούν με τις υπάρχουσες μεθόδους, καθώς έχουν την δυνατότητα να δρουν σε επίπεδο γλώσσας μηχανής, όπως π.χ. ένας keylogger ο οποίος επεμβαίνει στον κώδικα του Keyboard Interrupt Handler, η ακόμα και στον πυρήνα του λειτουργικού συστήματος όπως αρχεία DLL και SYS. Ωστόσο αν και το να επιβεβαιώνουμε την ακεραιότητα του πυρήνα του λειτουργικού συστήματος, του κώδικα των διεργασιών που βρίσκεται στην μνήμη και

γενικά των αρχείων συστήματος, είναι σχετικά εύκολο, προκύπτει ένα επιπλέον πρόβλημα. Μέσω της μεθόδου αυτής μπορούν να προκύψουν false alarms, λόγω της ιδιότητας κάποιον καλόβουλων λογισμικών όπως των software Firewalls, των IPS (Intrusion Prevention System) και Antivirus να επεμβαίνουν στο λειτουργικό σύστημα. Το γεγονός δε ότι αυτά τα λογισμικά πολλές φορές δεν έχουν ψηφιακά υπογεγραμμένα το σύνολο τον αρχείων τους, καθιστά το πρόβλημα ακόμα μεγαλύτερο.

Στην κατηγορία Type II [16] ταξινομούνται τα κακόβουλα λογισμικά τα οποία δεν επιδρούν σε κανένα από τα σταθερά στοιχεία του λειτουργικού συστήματος, αλλά στις μεταβλητές αυτού. Συγκεκριμένα επιδρούν στα δεδομένα τα οποία αντλούν οι διεργασίες του λειτουργικού συστήματος ή αλλιώς hooking points, προσθέτοντας τους κατάλληλους pointers, έτσι ώστε να εκτελείτε ο κακόβουλος κώδικας αντί αυτός του λειτουργικού. Τα προβλήματα στην αντιμετώπιση τους εδώ, έγκεινται στο γεγονός ότι τα δεδομένα αυτά δεν είναι σταθερά, όπως επίσης πολλές φορές δεν είναι και προσβάσιμα για λόγους πνευματικών δικαιωμάτων των κατασκευαστών, επομένως είναι φύση αδύνατον να εφαρμόζεται έλεγχος ακεραιότητας. Μια πιθανή λύση θα ήταν η εκ των προτέρων ρητή αναφορά των ευαίσθητων δομών δεδομένων και hooking points από τους κατασκευαστές λογισμικού, ώστε να καταστεί εφικτή η κατασκευή μιας διαδικασίας αυτοματοποιημένου ελέγχου ακεραιότητας.

Στην κατηγορία Type III [16] ταξινομούνται τα κακόβουλα λογισμικά τα οποία έχουν την ικανότητα ελέγχου ολόκληρου του λειτουργικού συστήματος, χωρίς να είναι απαραίτητη η μεταβολή οποιουδήποτε στοιχείου τους. Τα εν λόγω κακόβουλα λογισμικά δεν χρησιμοποιούν hooking points για να επηρεάσουν την λειτουργία του λειτουργικού συστήματος αλλά μπορούν να έχουν την μορφή τυχαίων δεδομένων στην μνήμη χρησιμοποιώντας τις τεχνολογίες virtualization των σύγχρονων επεξεργαστών. Η εν λόγω δυνατότητα είναι πολύ σημαντική και χρησιμοποιείται από τα λεγόμενα VMBR (Virtual Machine Based Rootkits), τα οποία καταφέρνουν να αλλάζουν την σειρά εκκίνησης του συστήματος και ουσιαστικά να λειτουργούν σαν hosts των λειτουργικών συστημάτων που προσβάλλουν. Μια πιθανή λύση στο πρόβλημα της αντιμετώπισης των εν λόγω κακόβουλων λογισμικών, θα ήταν η υλοποίηση ενός συστήματος hypervisor εικονικών μηχανημάτων ή αλλιώς virtual machines, ο οποίος θα απαγορεύει την λειτουργία των κακόβουλων λογισμικών της κατηγορίας Type III.

Κακόβουλο Λογισμικό	Ιδιότητες
Zeus	• Επεμβαίνει στην registry • Επεμβαίνει σε αρχεία dll • Χρησιμοποιεί API hooks • Επεμβαίνει σε εκτελέσιμα αρχεία του συστήματος
Hesperbot	• Επεμβαίνει στην διεργασία explorer.exe • Επεμβαίνει στην registry • Επεμβαίνει σε αρχεία dll • Επεμβαίνει σε διεργασίες οι οποίες εκτελούνται εκείνη την στιγμή
Buhtrap	• Δεν επεμβαίνει σε αρχεία του συστήματος • Χρησιμοποιεί δικά του dll • Χρησιμοποιεί WMI query • Χρησιμοποιεί API hooks • Χρησιμοποιεί 3nd party open source λογισμικό
Corkow	• Χρησιμοποιεί αλλά δεν μεταβάλλει αρχεία dll του συστήματος • Ανακατευθύνει κλήσεις dll του συστήματος προς δικά του dll

Πίνακας 6.1 Ιδιότητες Λειτουργίας Κακόβουλων Λογισμικών

6.1.1.1 Ταξινόμηση του Zeus

Έχουμε ήδη δει στην ανάλυση του ότι το κακόβουλο λογισμικό Zeus επεμβαίνει και αλλάζει δομικά στοιχεία του λειτουργικού συστήματος, όπως την registry και αρχεία dll. Στην μεν πρώτη έχει την δυνατότητα να εισάγει νέα κλειδιά, τα οποία περιέχουν τόσο δεδομένα του configuration file, όσο και εντολές εκτέλεσης του σε κάθε εκκίνηση του συστήματος. Στα δε αρχεία dll ενσωματώνει API hooks μέσω των οποίων υποκλέπτει

δεδομένα. Επίσης έχει την δυνατότητα επέμβασης σε εκτελέσιμα αρχεία του συστήματος όπως π.χ. το taskhost.exe, explorer.exe κ.α. στα οποία κάνει inject κακόβουλα threads κώδικα έτσι ώστε να γίνεται κατέβασμα και επεξεργασία του configuration file αλλά και έλεγχος καλής λειτουργίας του κακόβουλου λογισμικού.

Οι παραπάνω δυνατότητες και ιδιαίτερα η δυνατότητα αλλαγής του κώδικα των εκτελέσιμων αρχείων του συστήματος, αφορούν ξεκάθαρες επεμβάσεις στον πηγαίο κώδικα του λειτουργικού ο οποίος υπό προϋποθέσεις παραμένει αμετάβλητος. Έτσι σύμφωνα με την προτεινόμενη ταξινομία της Rutkowska, το κακόβουλο λογισμικό Zeus δεν θα μπορούσε παρά να κατηγοριοποιηθεί στην κατηγορία Type I.

6.1.1.2 Ταξινόμηση του Hesperbot

Το κακόβουλο λογισμικό Hesperbot με την σειρά του έχει την ιδιότητα να κάνει inject τον πυρήνα του στην διεργασία explorer.exe με χρήση κακόβουλων threads. Στην συνέχεια μέσω αυτής της διεργασίας, γίνεται η επικοινωνία με τον Command & Control Server και η εγγραφή κλειδιών απαραίτητων για την λειτουργία του στην registry. Μετά την αρχική προσβολή του συστήματος και το injection πυρήνα στην διεργασία explorer.exe, το Hesperbot έχει την δυνατότητα να κάνει inject τον πυρήνα του πρακτικά σε οποιαδήποτε διεργασία εκτελείται εκείνη την στιγμή. Επίσης το Hesperbot έχει την δυνατότητα ενσωμάτωσης διεργασιών στο user32.dll μέσω των οποίων εκτελεί εργασίες keylogging, καθώς και στα avidil32.dll και Gdi32.dll για την λήψη βίντεο και screenshots. Τέλος μέσω των plugins του, έχει την ικανότητα ενσωμάτωσης επιπλέον λειτουργιών στο αρχείο mswock.dll, μέσω των οποίων παρεμβάλλεται μέσω proxy στην διαδικτυακή επικοινωνία των browsers.

Το Hesperbot όπως και το Zeus επιχειρεί την άμεση επέμβαση τόσο σε εκτελέσιμα αρχεία του συστήματος, όσο και στον κώδικα αρχείων dll ώστε να επιτύχει τους κακόβουλους σκοπούς του. Η δυνατότητες αυτές είναι κι εκείνες που το κατατάσσουν στην κατηγορία Type I όπως αυτή ορίζεται στην πρόταση της Rutkowska.

6.1.1.3 Ταξινόμηση του Buhtrap

Το κακόβουλο λογισμικό Buhtrap αποτελεί μια περίπτωση κακόβουλου λογισμικού, η οποία δεν επεμβαίνει στον κώδικα του λειτουργικού συστήματος αλλάζοντας το προς όφελος του. Αντίθετα χρησιμοποιεί δικά του αρχεία είτε εκτελέσιμα είτε dll μέσω των οποίων συλλέγει πληροφορίες από το σύστημα. Συγκεκριμένα πραγματοποιεί έλεγχο των διεργασιών του συστήματος μέσω WMI query, όπως επίσης και έλεγχο της cache των browsers για τραπεζικά strings μέσω APIs. Στην κύρια λειτουργία του χρησιμοποιεί 3nd party λογισμικό για λόγους απόκρυψης, μέσω του οποίου ελέγχει την γλώσσα του λειτουργικού συστήματος και εκτελείται η εγκατάσταση του εκτελέσιμου αρχείου το οποίο αναλαμβάνει και την υπόλοιπη κακόβουλη λειτουργία. Το εκτελέσιμο αυτό αρχείο με το όνομα zerno.exe, αποκρύπτεται με την σειρά του εντός τρίτου καλόβουλου λογισμικού και καλεί την βιβλιοθήκη msvcr71.dll, η οποία αποτελεί μέρος του Buhtrap και περιλαμβάνει με την σειρά της τον κακόβουλο κώδικα.

Στην περίπτωση του Buhtrap παρατηρούμε ότι δεν επηρεάζεται άμεσα το λειτουργικό σύστημα αλλά το κακόβουλο λογισμικό ενεργεί αυτόνομα μέσω ξεχωριστού εκτελέσιμου αρχείου και βιβλιοθήκης dll. Σύμφωνα με την ταξινομία της Rutkowska τα κακόβουλα λογισμικά τα οποία δεν επηρεάζουν άμεσα το λειτουργικό σύστημα όπως στην περίπτωση μας το Buhtrap κατατάσσονται στην κατηγορία Type 0.

6.1.1.4 Ταξινόμηση του Corkow

Στην ανάλυση του κακόβουλου λογισμικού Corkow παρατηρήσαμε την χρήση βιβλιοθήκης dll σε συνδυασμό με εκτελέσιμο αρχείο για την εγκατάσταση του σε ένα σύστημα. Η βασική βιβλιοθήκη dll αποκρυπτογραφείται και αντιγράφεται στο σύστημα με χρήση της διεργασίας DLLMain, ενώ στην συνέχεια για λόγους απόκρυψης χρησιμοποιεί βιβλιοθήκες dll του συστήματος. Συγκεκριμένα αναζητά εντός τους φακέλου %SystemRoot%\System32 μη προστατευμένα αρχεία dll ώστε να ενσωματώσει τον κρυπτογραφημένο κώδικα του, ενώ παράλληλα ενσωματώνει και την κατάλληλη διεργασία για την αποκρυπτογράφηση και εκτέλεση του κακόβουλου κώδικα. Στην συνέχεια το αλλοιωμένο αρχείο dll αποθηκεύεται στον φάκελο του κακόβουλου λογισμικού, ενώ το

αρχικό παραμένει αναλλοίωτο. Οι δυνατότητες του εξαρτώνται από τα διάφορα modules τα οποία θα εγκατασταθούν μετά από επικοινωνία με τον C&C Server, παρόλα αυτά όμως ο αρχικός πυρήνας του Corkow είναι ικανός να εκτελέσει κακόβουλες ενέργειες όπως π.χ. την λήψη screenshots.

Στην περίπτωση του Corkow παρατηρούμε ότι ο κώδικας του λειτουργικού συστήματος παραμένει αναλλοίωτος ,καθώς το αρχείο dll που χρησιμοποιείται αρχικά παραμένει τελικά αμετάβλητο και απλά αντικαθίσταται η λειτουργία του με αυτό το οποίο δημιουργεί το Corkow. Από ότι φαίνεται το Corkow δεν επεμβαίνει άμεσα στο λειτουργικό σύστημα αλλά αντικαθιστά την κλήση συγκεκριμένης βιβλιοθήκης dll του συστήματος με αυτή που ο ίδιο έχει δημιουργήσει. Σύμφωνα με την ταξινομία της Rutkowska το Corkow θα μπορούσε να ενταχθεί στην κατηγορία Type II, καθώς δημιουργεί τέτοιες συνθήκες ώστε το λειτουργικό σύστημα να παραμείνει αναλλοίωτο και να καλείται δυναμικά η δική του βιβλιοθήκη DLL αντί αυτή του λειτουργικού.

Κακόβουλο Λογισμικό	**Κατηγορία**
Zeus	Type I
Hesperbot	Type I
Buhtrap	Type 0
Corkow	Type II

Πίνακας 6.2 Ταξινόμηση Κακόβουλων Λογισμικών

6.2 Συμπεράσματα

Στην παρούσα διατριβή, ορίσαμε την έννοια του κακόβουλου λογισμικού, μελετήσαμε τα κακόβουλα λογισμικά τα οποία χρησιμοποιούνται σε απάτες συναλλαγών, όπως επίσης και τον αντίκτυπο τους στην καθημερινότητα. Επίσης προχωρήσαμε στην ανάλυση του διεθνούς προβλήματος ονοματοδοσίας-ταξινομίας των κακόβουλων λογισμικών αλλά και στην παρουσίαση αναλύσεων κάποιων από τα πλέον σημαντικά κακόβουλα λογισμικά που χρησιμοποιούνται για απάτες συναλλαγών. Τέλος έγινε μια

προσπάθεια ταξινόμησης τους μέσω μιας πρωτότυπης πρότασης ταξινομίας από την Joanna Rutkowska.

Με βάση τα παραπάνω γίνεται κατανοητό ότι η χρήση κακόβουλου λογισμικού για απάτες συναλλαγών έχει σημαντικό αντίκτυπο στην παγκόσμια οικονομία και τα μέτρα αντιμετώπισης τους είναι πιο επιτακτικά από ποτέ. Η ανάλυση στην παρούσα διατριβή μερικών από τα πλέον επικίνδυνα εξ αυτών καταδεικνύει όχι μόνο την επικινδυνότητα τους αλλά ίσως και τα σημαντικότερα τους χαρακτηριστικά, όπως την ευκολία παραγωγής και παραμετροποίησης, την δυνατότητα εύκολης αλλαγής τρόπου δράσης με χρήση διαφορετικών modules κατά προτίμηση, την δυσκολία στην ανίχνευση αλλά και την δυνατότητα της επιλογής των συστημάτων-στόχων. Τα παραπάνω κάνουν επίσης δύσκολη την διαδικασία της ταξινόμησης τους με του συνήθεις τρόπους για αυτό και επιλέχθηκε η εναλλακτική πρόταση της Rutkowska, η οποία δεν βασίζεται τόσο στα χαρακτηριστικά του ίδιου του κακόβουλου λογισμικού αλλά περισσότερο στο τι ακριβώς επηρεάζει σε ένα μολυσμένο σύστημα.

Συμπερασματικά, οι προτάσεις της Rutkowska φαίνεται να μπορούν να ταξινομήσουν τα διαρκώς μεταβαλλόμενα κακόβουλα λογισμικά αυτού του είδους με σχετική ακρίβεια, ενώ παράλληλα οι προτάσεις της για ενδεχόμενους τρόπους προστασίας από το κακόβουλο λογισμικό κάθε κατηγορίας μπορούν να θεωρηθούν εύλογες. Το μεγάλο πρόβλημα το οποίο αντιμετωπίζει και αυτή η πρόταση όμως είναι η αντοχή της σε μελλοντικές μετεξελίξεις του κακόβουλου λογισμικού, αλλά και η ανάγκη για παγκόσμια αποδοχή προτύπων σε ότι αφορά τους την προστασία από κάθε κατηγορία κακόβουλου λογισμικού.

Κλείνοντας την παρούσα διατριβή πρέπει να τονιστεί ότι το μεγαλύτερο πρόβλημα στην αντιμετώπιση του κακόβουλου λογισμικού, ειδικότερα σε ότι αφορά τις απάτες συναλλαγών δεν αποτελεί το ίδιο το κακόβουλο λογισμικό αλλά η μη ύπαρξη καθιερωμένων και ευρέως αποδεκτών προτύπων τόσο σε ότι αφορά την ταξινομία τους όσο και την τους τρόπους αντιμετώπισης τους. Είναι εμφανές ότι η σωστή ταξινομία των κακόβουλων λογισμικών καταρχήν είναι αυτή η οποία μπορεί να οδηγήσει στην αποτελεσματικότερη λήψη μέτρων αντιμετώπισης τους και ότι τώρα περισσότερο από ποτέ είναι αναγκαία η

καθιέρωση διεθνών προτύπων, μια καλή βάση προς συζήτηση των οποίων θα μπορούσε να αποτελεί η πρόταση της Rutkowska.

Βιβλιογραφία

[1] "Is $35 Billion in Card Fraud the 'Cost of Doing Business'? | Chip Shield." [Online]. Available: http://www.chipshield.com/is-35-billion-in-card-fraud-the-cost-of-doing-business/.

[2] M. Souppaya and K. Scarfone, "Guide To Malware Incident Prevention and Handling for Desktops and Laptops," *Int. J. Comput. Res.*, vol. 20, no. 4, pp. 417–458, 2013.

[3] A. P. and A. Carblanc, "Malicious Software (Malware): A security Threat to the Internet Economy," *OECD*, pp. 1–106, 2008.

[4] K. Julisch, "Risk-Based Payment Fraud Detection," *IBM Res.*, vol. 32, no. 4, pp. 888–896, 2002.

[5] R. J. Sullivan, "Controlling Security Risk and Fraud in Payment Systems," *Fed. Reserv. Bank Kansas City Econ. Rev.*, vol. Third Quar, pp. 47–78, 2014.

[6] J. D. Aycock, *Computer viruses and malware*. 2006.

[7] J. Kałużny and M. Olejarka, "Script-based malware detection in Online Banking Security Overview," *SecuRing*, 2010.

[8] FBI, "FBI — International Blackshades Malware Takedown," 2014. [Online]. Available: https://www.fbi.gov/news/stories/2014/may/international-blackshades-malware-takedown/international-blackshades-malware-takedown.

[9] B. H. Michael Hale Ligh, Steven Adair and Artstein, *Malware Analyst's Cookbook*. 2011.

[10] M. A. M. and A. H. O. Ammar Ahmed E. Elhadi, "Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph Ammar Ahmed E. Elhadi, Mohd Aizaini Maarof and Ahmed Hamza Osman Information Assurance and

Security Research Group, Faculty of Computer Science and Infor," *Journal, Am. Sci. Appl. Publ. Sci.*, vol. 9, no. 3, pp. 283–288, 2012.

[11] M. Sikorski and A. Honig, "Practical Malware Analysis," *No Starch*, vol. 53, no. 9, pp. 1689–1699, 2013.

[12] A. Sanabria, "Malware Analysis: Environment Design and Artitecture," *SANS Inst. InfoSec Read. Room*, 2007.

[13] Kaspersky Lab, "Securelist-Clasification." [Online]. Available: https://securelist.social-kaspersky.com/en/descriptions/Trojan.Win32.AutoRun.gen.

[14] V. Bontchev, "Current Status of the CARO Malware Naming Scheme," 2005.

[15] I. Kirillov, D. Beck, P. Chase, and R. Martin, "Malware Attribute Enumeration and Characterization," 2011.

[16] J. Rutkowska, "Introducing Stealth Malware Taxonomy," *COSEINC Adv. Malware Labs*, no. November, pp. 1–9, 2006.

[17] J. Wyke, "What is Zeus?" no. May, pp. 1–17, 2011.

[18] A. Cherepanov and R. Lipovsky, "Hesperbot – A New, Advanced Banking Trojan in the Wild," 2013.

[19] Group-IB, "Buhtrap - The Evolution of Targeted Attacks Against Financial Institutions," 2016.

[20] Dhruval Gandhi, "Banking Malware Buhtrap Caught in Action – Cyphort," 2016. [Online]. Available: https://www.cyphort.com/banking-malware-buhtrap-caught-action/. [Accessed: 08-Mar-2017].

[21] JEAN-IAN BOUTIN, “Operation Buhtrap, the trap for Russian accountants,” 2015. [Online]. Available: http://www.welivesecurity.com/2015/04/09/operation-buhtrap/. [Accessed: 09-Mar-2017].

[22] Paul Kimayong, “Buhtrap Malware: What Every Bank’s Security Team Needs To Know – Cyphort,” 2016. [Online]. Available: https://www.cyphort.com/buhtrap-malware-every-banks-security-team-needs-know/. [Accessed: 09-Mar-2017].

[23] R. Lipovsky, “Corkow: Analysis of a business-oriented banking Trojan,” 2014. [Online]. Available: http://www.welivesecurity.com/2014/02/27/corkow-analysis-of-a-business-oriented-banking-trojan/. [Accessed: 10-Mar-2017].