ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας΄»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Δυναμική Ανάλυση Κακόβουλου Λογισμικού και χρήση των LOLBAS/LOLBINs** **Dynamic malware analysis and the abuse of LOLBAS/LOLBINs**
Ονοματεπώνυμο Φοιτητή	**Καρακατσάνης Ζώης**
Πατρώνυμο	**Ελευθέριος**
Αριθμός Μητρώου	**ΜΠΚΣΑ20018**
Επιβλέπων	**Κωνσταντίνος Πατσάκης, Αναπληρωτής Καθηγητής**

Ημερομηνία Παράδοσης **Νοέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Πατσάκης Κωνσταντίνος Αναπληρωτής Καθηγητής	Αλέπης Ευθύμιος Αναπληρωτής Καθηγητής	Σακκόπουλος Ευάγγελος Αναπληρωτής Καθηγητής

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πολύ τον επιβλέποντα καθηγητή μου Κωνσταντίνο Πατσάκη, για την καθοδήγηση του και τις πολύτιμες συμβουλές του καθ' όλη την διάρκεια της συγγραφής της μεταπτυχιακής μου διατριβής.

Επιπλέον, αφιερώνω αυτή την δουλειά στην οικογένεια μου, που με στηρίζει και πιστεύει σε μένα σε κάθε μου προσπάθεια.

Περίληψη

Στην παρούσα μεταπτυχιακή διατριβή δείξαμε τον τρόπο δράσης επιθέσεων από κακόβουλα λογισμικά Ransomware. Αυτό το δείξαμε δημιουργώντας εικονικά μηχανήματα μέσω του Virtual Box και κάνοντας χρήση του Cuckoo Sandbox. Πριν ξεκινήσουμε τις αναλύσεις κάναμε μερική παραμετροποίηση του εικονικού μηχανήματος Windows 10 με τους καλύτερους δυνατούς τρόπους ώστε να μην μπορούν να το ανιχνεύσουν τα Ransomware και να έχουμε καλύτερα αποτελέσματα όσον αφορά την λειτουργικότητά τους. Μέσω δυναμικής ανάλυσης εξετάσαμε τα δέκα πιο πετυχημένα είδη Ransomware. Συνολικά αναλύσαμε διακόσια δείγματα, δηλαδή είκοσι δείγματα από το κάθε ένα και παρουσιάσαμε τους μηχανισμούς λειτουργίας και τους μηχανισμούς για την αποφυγή εντοπισμού και ανάλυσης από ερευνητές ασφαλείας. Επίσης εντοπίσαμε μέσω των Json αποτελεσμάτων τα LOLBAS αρχεία, και παρουσιάσαμε τα είδη των κρυπτογραφικών μεθόδων καθώς και τα μηνύματα που αφήνουν στον χρήστη για την πληρωμή λύτρων. Επίσης παρουσιάσαμε μια πληθώρα τεχνικών και ιδιαιτεροτήτων του κάθε Ransomware. Δείξαμε επίσης γραφήματα συνολικών στατιστικών στοιχείων για την καλύτερη κατανόηση των αναλύσεών μας, την επίδειξη των δυνατοτήτων της εικονικής μηχανής μας, αλλά και την βαρύτητα μιας τέτοιας επίθεσης σε οποιοδήποτε σύστημα μολύνουν. Η συγκεκριμένη διατριβή μπορεί να φανεί πολύ χρήσιμη σε ερευνητές ασφαλείας ώστε να ξέρουν τι θα εντοπίσουν ακολουθώντας την σχετική υλοποίηση αλλά και πληροφοριακά για την δημιουργία αμυντικών μηχανισμών των συστημάτων τους.

Λέξεις-Κλειδιά: Ιομορφικό Λογισμικό, Κακόβουλο Λογισμικό, Ransomware, Cuckoo, Sandbox, LOLBAS, Δυναμική Ανάλυση, Αποφυγή Ανάλυσης, Κρυπτογράφηση, Εικονικό μηχάνημα

Abstract

In this master's thesis we have shown how malicious software's named Ransomware attacks work. We demonstrated this by creating virtual machines through Virtual Box and using Cuckoo Sandbox. Before starting the analyses we partially configured the Windows 10 virtual machine in the best possible ways so that it cannot be detected by Ransomware and we get better results in terms of their functionality. Through dynamic analysis we examined the ten most successful types of Ransomware. In total we analyzed two hundred samples, that is twenty samples of each and presented the mechanisms of operation and the mechanisms to avoid detection and analysis by security researchers. We also identified through the Json results the LOLBAS files, and presented the types of cryptographic methods as well as the messages they leave for the user to pay the ransom. We also presented a multitude of techniques and peculiarities of each Ransomware. We have also shown graphs of overall statistics to better understand our analysis, our virtual machine capabilities and also the severity of such an attack on any system they infect. This specific thesis can be very useful to security researchers to know what they will detect following the relevant implementation but also informative for the creation of defense mechanisms of their systems.

Keywords: Iomorphic Software, Malware, Ransomware, Cuckoo, Sandbox, LOLBAS, Dynamic Analysis, Analysis Evasion, Encryption, Virtual Machine

Περιεχόμενα

Περίληψη 3
Abstract 3
Πίνακας Εικόνων 6
1 Εισαγωγή 8
1.1 Στόχοι και Περιγραφή της Έρευνας 8
1.2 Δημιουργία Εργαστηρίου και Μεθοδολογία 10
1.3 Ανάλυση Αποτελεσμάτων για τον εντοπισμό Living Off the Land Binaries, Scripts and Libraries 16
2 Conti 18
2.1 Living Off the Land Binaries, Scripts and Libraries 18
2.2 Μηχανισμοί Λειτουργίας 26
2.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 32
2.4 Κρυπτογράφηση 37
3 Netwalker(Mailto) 40
3.1 Living Off the Land Binaries, Scripts and Libraries 40
3.2 Μηχανισμοί Λειτουργίας 42
3.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 43
3.4 Κρυπτογράφηση 44
4 Locky 46
4.1 Living Off the Land Binaries, Scripts and Libraries 46
4.2 Μηχανισμοί Λειτουργίας 47
4.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 50
4.4 Κρυπτογράφηση 51
5 Revil/Sodinokibi 53
5.1 Living Off the Land Binaries, Scripts and Libraries 53
5.2 Μηχανισμοί Λειτουργίας 55
5.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 55
5.4 Κρυπτογράφηση 56
6 DarkSide 58
6.1 Living Off the Land Binaries, Scripts and Libraries 58
6.2 Μηχανισμοί Λειτουργίας 60
6.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 60
6.4 Κρυπτογράφηση 61
7 Ryuk 62
7.1 Living Off the Land Binaries, Scripts and Libraries 62
7.2 Μηχανισμοί Λειτουργίας 64
7.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 65
7.4 Κρυπτογράφηση 65
8 RagnarLocker 67

8.1 Living Off the Land Binaries, Scripts and Libraries 67
8.2 Μηχανισμοί Λειτουργίας 68
8.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 68
8.4 Κρυπτογράφηση 69
9 MountLocker 70
9.1 Living Off the Land Binaries, Scripts and Libraries 70
9.2 Μηχανισμοί Λειτουργίας 72
9.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 73
9.4 Κρυπτογράφηση 73
10 BlackMatter 75
10.1 Living Off the Land Binaries, Scripts and Libraries 75
10.2 Μηχανισμοί Λειτουργίας 77
10.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 77
10.4 Κρυπτογράφηση 78
11 Egregor 79
11.1 Living Off the Land Binaries, Scripts and Libraries 79
11.2 Μηχανισμοί Λειτουργίας 81
11.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques) 81
11.4 Κρυπτογράφηση 82
12 Στατιστικά Στοιχεία Αναλύσεων 83
13 Στατιστικά Στοιχεία Επιθέσεων Ransomware 90
14 Συμπεράσματα 93
Χρήσιμοι σύνδεσμοι 94
Βιβλιογραφικές Αναφορές 94

Πίνακας Εικόνων

Εικόνα 1: Ανίχνευση Περιβάλλοντος 11
Εικόνα 2: Pafish 1....12
Εικόνα 3: Pafish 2....13
Εικόνα 4: Cuckoo Environment 14
Εικόνα 5 Ανατομία επίθεσης Ransomware....14
Εικόνα 6: Εσωτερική Ανατομία Επίθεσης Ransomware....15
Εικόνα 7: Σύγκριση μεθόδων Malware με και χωρίς LOLBAS τεχνικές 16
Εικόνα 8: Explorer.exe 18
Εικόνα 9: Process32NextW....19
Εικόνα 10: Processing Tree 20
Εικόνα 11: Dllhost.exe....22
Εικόνα 12: Svchost.exe....22
Εικόνα 13: Cmd.exe 23
Εικόνα 14: Cmd.exe Activity....23
Εικόνα 15: Delete Shadow Copies with Vssadmin 24
Εικόνα 16: Delete Shadow Copies with WMIC.exe 25
Εικόνα 17: Delete Shadow Copies with WMIC.exe 2 26
Εικόνα 18: Process Injection 26
Εικόνα 19: PDB Path....27
Εικόνα 20: Process Crashed....27
Εικόνα 21: Cab File....27
Εικόνα 22: Suspicious File 28
Εικόνα 23: Chinese Language 28
Εικόνα 24: Host With No DNS Query 29
Εικόνα 25: Kernel Module Without a Name....29
Εικόνα 26: Vssadmin Fail....29
Εικόνα 27: Allocated Memory....30
Εικόνα 28: Dead Host....30
Εικόνα 29: General Behavior....31
Εικόνα 30: PPID 31
Εικόνα 31: Entropy 32
Εικόνα 32: Thread Handles in Other Processes....33
Εικόνα 33: Malfind Detection....33
Εικόνα 34: PEB Modified....33
Εικόνα 35: Stopped Application Layer Gateway Service....33
Εικόνα 36: Stopped Firewall Service....34
Εικόνα 37: Resumed Suspended Thread....34
Εικόνα 38: Unknown PE Section....34
Εικόνα 39: Known Packer 35
Εικόνα 40: Debugged by a debugger 35
Εικόνα 41: Delay Analysis....36
Εικόνα 42: Searches Running Processes 36
Εικόνα 43: Known Packer 36
Εικόνα 44: Cryptsp.dll....37
Εικόνα 45: SSL....37
Εικόνα 46: Cryptsvc....38
Εικόνα 47: CryptAcqireContextA 38
Εικόνα 48: Winrar....38
Εικόνα 49: Conti Ransom Message 39
Εικόνα 50: Rundll32.exe....41
Εικόνα 51: PPID of Inject-x86.exe....41
Εικόνα 52: Analyzer.py....42
Εικόνα 53: Rundll32.exe Call 42
Εικόνα 54: Vssadmin....43
Εικόνα 55: SSL Protocol....44
Εικόνα 56: Cryptsp.dll....44

Εικόνα 57: Netwalker(Mailto) Ransom Message 45
Εικόνα 58: Vssadmin Delete Shadow Copies 48
Εικόνα 59: Creates Office Documents 48
Εικόνα 60: Word Macros 48
Εικόνα 61: Office Documents 48
Εικόνα 62: Office Documents 2 49
Εικόνα 63: Encrypted Office Documents 49
Εικόνα 64: Attack through malicious MS Office documents 49
Εικόνα 65: Structure of typical Word and Excel documents 50
Εικόνα 66: RSA Machine Keys 51
Εικόνα 67: CryptEncrypt 51
Εικόνα 68: Locky Ransom Message 52
Εικόνα 69: Locky Ransom message from Json file 52
Εικόνα 70: Powershell.exe version 1 53
Εικόνα 71: Sleep to Delay 54
Εικόνα 72: MsMpEng.exe and Volatility 54
Εικόνα 73: Present and Enabled 54
Εικόνα 74: Crypto\\RSA 56
Εικόνα 75: Readme.txt in Buffer 56
Εικόνα 76: Revil/Sodinokibi Ransom Message 57
Εικόνα 77: Too Few Signatures 59
Εικόνα 78: Obfuscated Powershell Command 60
Εικόνα 79: De-Obfuscated PowerShell Command 60
Εικόνα 80: DarkSide Ransom Message 61
Εικόνα 81: Wake on Lan 63
Εικόνα 82: File Access with icalcs.exe 63
Εικόνα 83: Persistence Command 63
Εικόνα 84: SeDebugPriviledge, AdjustTokenPriviledges 63
Εικόνα 85: Ryuk Ransom Message in Json 66
Εικόνα 86: Ryuk Encryption 66
Εικόνα 87: Ryuk Ransom Message 66
Εικόνα 88: RagnarLocker Ransom Message 69
Εικόνα 89: MountLocker Ransom Message 74
Εικόνα 90: System Info and Process Kill 74
Εικόνα 91: SeTokenPriviledge 76
Εικόνα 92: BlackMatter Ransom Message 78
Εικόνα 93: RDP Services Stopped 81
Εικόνα 94: Egregor Ransom Message 82
Εικόνα 95: Ransomware Operations 86
Εικόνα 96: Ransomware Evasion Techniques 87
Εικόνα 97: Percentage of Ransomware Using Operations 88
Εικόνα 98: Percentage of Ransomware Using Operations 2 88
Εικόνα 99: Percentage of Ransomware Using Evasion Techniques 89
Εικόνα 100: Difference in Percentages 89
Εικόνα 101: Προτιμώμενοι στόχοι Ransomware επιθέσεων 90
Εικόνα 102: Στατιστικά κατά προσέγγιση των επιθέσεων και του κόστους 91

1 Εισαγωγή

Στην σύγχρονη εποχή που διανύουμε είναι γνωστό πως οι περισσότερες εταιρίες και οργανισμοί χρησιμοποιούν την επιστήμη της πληροφορικής ως κύριο εργαλείο της λειτουργίας τους και των επαγγελματικών τους δραστηριοτήτων. Παρατηρείται ότι καθώς αυξάνεται η εξάρτησή στις ψηφιακές τεχνολογίες αυξάνονται και οι κίνδυνοι που επιφυλάσσουν. Το αποτέλεσμα αυτών των δεδομένων είναι τα πληροφοριακά συστήματα να αποκτούν μεγάλη αξία για τον οργανισμό αλλά και για άλλους επιτήδειους που σκοπεύουν στην εκμετάλλευση τους. Έτσι λοιπόν βλέπουμε πως η ασφάλεια των συστημάτων βάλλεται διαρκώς από νεοεμφανιζόμενα ιομορφικά λογισμικά και άλλες εξελιγμένες τεχνικές, ώστε η ανάγκη για αμυντικά συστήματα να είναι μεγαλύτερη από ποτέ. Η πλευρά των επιτιθέμενων αποτελείται από συγγραφείς κακόβουλων λογισμικών και εξαιρετικούς γνώστες επιθετικών εργαλείων πληροφορικής, ενώ η πλευρά των αμυνόμενων αποτελείται από ερευνητές ασφαλείας, αυθεντίες στον εντοπισμό απειλών και στην αποτροπή αυτών. Είναι γνωστό ότι τα κακόβουλα λογισμικά υπάρχουν εδώ και αρκετό καιρό. Παρά το γεγονός ότι οι πρώτες υλοποιήσεις συστημάτων πληροφοριών ήταν πολύ πιο ευάλωτες, οι κακόβουλες επιθέσεις δεν αποτελούσαν ανησυχία και αυτό γιατί κατά τη διάρκεια εκείνων των πρώτων ετών, πολύ λίγοι άνθρωποι ήταν σε θέση να κατανοήσουν σε βάθος πώς λειτουργούν τα συστήματα πληροφοριών και επομένως να τα εκμεταλλευτούν.

Στις μέρες μας το πλήθος των διαδικτυακών απειλών είναι αδιαμφησβήτητα μεγάλο. Ωστόσο σε αυτήν την διατριβή δεν θα εξετάσουμε όλες τις απειλές των υπολογιστικών συστημάτων, αλλά θα παρουσιάσουμε την μεγαλύτερη εξ αυτών που είναι τα επονομαζόμενα Ransomware. Αν και τα Ransomware υπάρχουν από τις πρώτες μέρες των προσωπικών υπολογιστών, η πολυπλοκότητά και η επιθετικότητα τους έχουν αυξηθεί σημαντικά με τα χρόνια.

Το Ransomware, είναι ένα είδος κακόβουλου λογισμικού που κρυπτογραφεί τα αρχεία του θύματος ώστε να το εκβιάσει, με στόχο την απολαβή μεγάλου χρηματικού ποσού για την αποκρυπτογράφηση των αρχείων, και την μη έκθεση των πληροφοριών που περιείχαν στις ιστοσελίδες του επιτιθέμενου. Το Ransomware έχει σχεδιαστεί για να εξαπλώνεται σε πολλαπλά συστήματα σε ένα δίκτυο και να στοχεύει βάσεις δεδομένων και διακομιστές αρχείων, και έτσι μπορεί να παραλύσει γρήγορα έναν ολόκληρο οργανισμό. Είναι μια αυξανόμενη απειλή, που οδηγεί σε πληρωμές υπέρογκων ποσών σε εγκληματίες του κυβερνοχώρου και προκαλεί σημαντικές ζημιές και έξοδα σε επιχειρήσεις και κυβερνητικούς οργανισμούς. Έτσι όλο και περισσότεροι οργανισμοί ζητούν ερευνητές για την προστασία των συστημάτων τους καθώς οι συμβατικές λύσεις ασφάλειας δεν καλύπτουν επαρκώς αυτή την ανάγκη.

1.1 Στόχοι και Περιγραφή της Έρευνας

Η έρευνα που ακολουθήσαμε σε αυτήν την εργασία εξετάζει τις δυνατότητες ιών που προσβάλουν ηλεκτρονικά συστήματα με στόχο τον πλήρες έλεγχό τους. Η οικογένεια ή κατηγορία κακόβουλων λογισμικών που αναλύθηκε είναι τα λεγόμενα Ransomware που ως κύριο στόχο έχουν την αδυναμία λειτουργίας των συστημάτων μας κρυπτογραφώντας τα αρχεία ώστε να μην μπορούμε να τα ανακτήσουμε, εκτός αν υποκύψουμε στον οικονομικό εκβιασμό που απαιτείται από τους θήτες. Για την διεξαγωγή της έρευνας αναζητήσαμε πληροφορίες που βρήκαμε στην ιστοσελίδα https://ransomwhe.re// όπου εντοπίσαμε τα πιο δημοφιλή ransomware με χρονικό περιθώριο all time τα οποία είναι το Conti, Netwalker(Mailto), Locky, REvil/Sodinokibi, DarkSide, Ryuk, RagnarLocker, MountLocker, BlackMatter, Egregor. Από τα αναγραφόμενα βρήκαμε δείγματα μέσω της ιστοσελίδας https://bazaar.abuse.ch/ και κατεβάσαμε διακόσια δείγματα είκοσι για το κάθε ένα για να τα εξετάσουμε.

Στόχος μας είναι να κάνουμε αυτόματη δυναμική ανάλυση των Ransomware χρησιμοποιώντας το εργαλείο Cuckoo Sandbox μέσω ενός εικονικού μηχανήματος με λειτουργικό Windows 10. Στην συνέχεια λαμβάνοντας τα αποτελέσματα από είκοσι δείγματα του κάθε Ransomware παρουσιάζουμε:

- Τα LOLBAS που χρησιμοποιήθηκαν και την λειτουργικότητά τους
- Τους μηχανισμούς λειτουργίας τους, και εμβάθυνση στους πιο ενδιαφέρων

- Τους μηχανισμούς λειτουργίας για την αποφυγή ανίχνευσης των Ransomware από μηχανισμούς ασφαλείας αλλά και ανάλυσης(Evasion Techniques)
- Τα είδη των κρυπτογραφικών μεθόδων που χρησιμοποιεί το κάθε Ransomware
- Τα μηνύματα πληρωμής λύτρων
- Γραφήματα συνόλων και συγκρίσεων

Το Cuckoo Sandbox είναι ένα εργαλείο ανοικτού κώδικα [1] που χρησιμοποιείται για την εκκίνηση κακόβουλου λογισμικού σε ένα ασφαλές και απομονωμένο περιβάλλον και μέσω των μηχανισμών του είναι ικανό:

- Αναλύει πολλούς διαφορετικούς τύπους αρχείων όπως εκτελέσιμα, αρχεία Office, pdf, και λοιπά, καθώς και μολυσμένες ιστοσελίδες μέσα σε εικονικά περιβάλλοντα Windows, Linux, MacOS and Android.
- Καταγράφει API Calls και την γενική συμπεριφορά του αρχείου και παρέχει τις πληροφορίες και τα signatures με τρόπο που μπορούν να γίνουν κατανοητά.
- Κάνει ανάλυση της δικτυακής κινητικότητας και την δίνει σε εξαγόμενο αρχείο PCAP
- Εκτελεί προηγμένη ανάλυση μνήμης του μολυσμένου συστήματος μέσω του Volatility.

Η βασική ιδέα είναι ότι το Sandbox ξεγελάει το κακόβουλο λογισμικό και νομίζει ότι έχει μολύνει έναν γνήσιο κεντρικό υπολογιστή. Στην μεθοδολογία παρακάτω θα δούμε πως μπορούμε να βελτιώσουμε αυτήν την ιδέα και να κρύψουμε ακόμα καλύτερα το εικονικό μας μηχάνημα από τα Ransomware.

Η προσοχή μας σε αυτή την έρευνα εστιάζεται και στα LOLBAS (Living Off The Land Binaries, Scripts and Libraries) τα οποία αναφέρονται λεπτομερέστερα στην ιστοσελίδα https://lolbas-project.github.io/ στους μηχανισμούς που χρησιμοποιούν, την κρυπτογράφηση, τις τεχνικές για την αποφυγή της ανάλυσης (anti-analysis techniques) και τι κοινά στοιχεία υπάρχουν μεταξύ των Json αποτελεσμάτων. Για να θεωρηθεί ένα αρχείο ως LOLBAS πρέπει να πληροί ορισμένες προδιαγραφές οι οποίες είναι :

- Να είναι αρχείο υπογεγραμμένο από τη Microsoft, είτε εγγενές στο λειτουργικό σύστημα είτε έχει ληφθεί από τη Microsoft.
- Έχουν επιπλέον «απροσδόκητη» λειτουργικότητα. Δεν είναι ενδιαφέρον να τεκμηριώνονται περιπτώσεις χρήσης για την οποία προορίζονται.
    - Εξαιρέσεις αποτελούν οι παρακάμψεις στη λίστα επιτρεπόμενων εφαρμογών
- Να έχει λειτουργικότητα που θα ήταν χρήσιμη σε μια κόκκινη ομάδα (Red Team) η APT(Advanced Persistent Threat) ομάδα.

Ενδιαφέρουσα λειτουργικότητα μπορεί να περιλαμβάνει:

- Εκτέλεση κώδικα
    - Αυθαίρετη εκτέλεση κώδικα
    - Εκτέλεση μεταβίβασης άλλων προγραμμάτων (χωρίς υπογραφή) ή σεναρίων (μέσω LOLBin)
- Μεταγλώττιση κώδικα
- Λειτουργίες αρχείων
    - Λήψη
    - Μεταφόρτωση
    - Αντιγραφή
- Επιμονή (Persistence)
    - Επιμονή διέλευσης με χρήση υπάρχοντος LOLBin
    - Επιμονή (π.χ. απόκρυψη δεδομένων στο ADS, εκτέλεση κατά τη σύνδεση)
- Παράκαμψη UAC (User Account Control)
- Κλοπή διαπιστευτηρίων
- Αποθήκευση των διεργασιών της μνήμης
- Παρακολούθηση (π.χ. keylogger, δικτύου)

- Διαφυγή/τροποποίηση των log αρχείων
- Πλάγια φόρτωση/hijacking DLL χωρίς την εγκατάσταση του αλλού στο σύστημα αρχείων.

Τα LOLBAS χρησιμοποιούνται από κακόβουλα λογισμικά αφού έχουν πάρει πρόσβαση στο σύστημα-θύμα για όλες τις παραπάνω λειτουργίες που αναφέρουμε και πάντα έχοντας ως στόχο να μην γίνουν αντιληπτά από οποιοδήποτε μηχανισμό ασφαλείας όπως Antivirus, IPS, IDS, EDR, DFIR, EPM. Αναμειγνύονται με τις υπόλοιπες υπογεγραμμένες από την Microsoft διεργασίες ώστε ο εντοπισμός μιας δραστηριότητας με χρήση LOLBAS καθίσταται μια πολύ δύσκολη διαδικασία.

Τους υπόλοιπους στόχους μας όπως τους μηχανισμούς λειτουργίας, αποφυγής ανάλυσης, τις κρυπτογραφικές μεθόδους, τα μηνύματα λύτρων και τα γραφήματα θα τα δούμε αναλυτικά σε όλη την έκταση της διατριβής.

1.2 Δημιουργία Εργαστηρίου και Μεθοδολογία

Η υλοποίηση μας ξεκινάει έχοντας στην διάθεσή μας έναν υπολογιστή με λειτουργικό σύστημα Windows 10 αρκετή χωρητικότητα μνήμης Ram 32GB, επεξεργαστή με έξι πυρήνες και δώδεκα εικονικούς και αρκετό χώρο στον σκληρό δίσκο μας 1TB. Στην συνέχεια προχωρήσαμε στην εγκατάσταση του λειτουργικού VirtualBox της Oracle και δημιουργήσαμε μια εικονική μηχανή με λειτουργικό σύστημα Ubuntu 21.10 δίνοντας τους απαραίτητους πόρους για την λειτουργία του συστήματος. Ακολούθως εγκαταστήσαμε το Cuckoo με τα προ απαιτούμενα που χρειάζονται για την ομαλή λειτουργία του. Το Cuckoo χρειάζεται επίσης ένα εικονικό μηχάνημα όπου εκεί θα εκτελεστεί το Ransomware. Έτσι εγκαταστήσαμε το Virtual Box μέσα στο Ubuntu για να κάνουμε ένα nesting virtualization δηλαδή ένα εικονικό μηχάνημα που τρέχει μέσα σε ένα άλλο εικονικό μηχάνημα. Άρα το τελικό στήσιμο των μηχανημάτων έχει ένα Windows 10 που τρέχει μέσα σε Ubuntu και αυτό μέσα σε Windows 10 πάλι.

Συνεχίζοντας κάναμε ορισμένες παραμετροποιήσεις στο δίκτυο, στα settings του virtual box στο Ubuntu αλλά και στα αρχεία του Cuckoo για να ορίσουμε τις επιθυμητές λειτουργίες. Στα settings των Windows 10 που είναι μέσα στο Ubuntu ορίσαμε στην κάρτα δικτύου να είναι Host Only. Στην συνέχεια εκτελέσαμε εντολές μέσω iptables για να μπορέσουμε να δώσουμε δίκτυο στο εσωτερικό Windows μηχάνημα μέσω του Ubuntu ώστε να περνάει η δικτυακή κίνηση του μέσα από αυτό. Οι εντολές είναι :

1. sudo apt-get install -y iptables-persistent
2. sudo iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.56.0/24 -j MASQUERADE
3. sudo iptables -P FORWARD DROP
4. sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
5. sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT
6. sudo iptables -A FORWARD -s 192.168.56.0/24 -d 192.168.56.0/24 -j ACCEPT
7. sudo iptables -A FORWARD -j LOG
8. sudo su
9. iptables-save > /etc/iptables/rules.v4

Ακολούθως μέσα στον φάκελο .cuckoo/conf στο αρχείο cuckoo conf, βάζουμε

- machinery = virtualboxmemory_dump = yes
- resultserver ip = 192.168.56.1

στο auxiliary.conf

- ενεργοποιούμε το sniffer enabled = yes

στο virtualbox.conf

- βάζουμε mode = gui
- machines = cuckoo1
- label = cuckoo1
- platform = windows
- ip = 192.168.56.101
- snapshot = Snapshot 1

στο processing.conf βάζουμε το memory

- enabled = yes

στο memory.conf για την χρήση του Volatility βάζουμε προφίλ

- guest_profile =Win10x64_17763

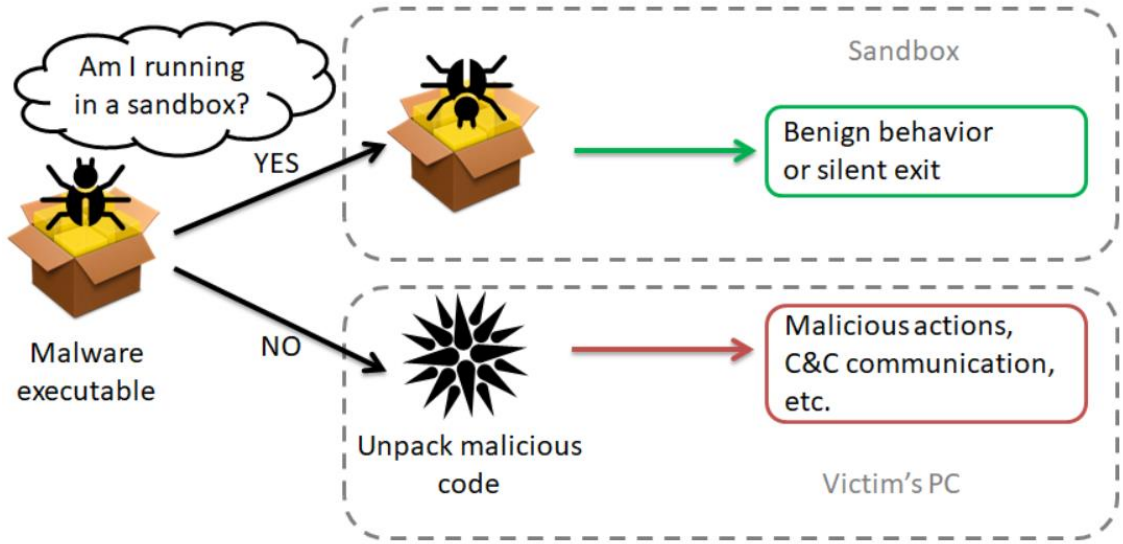
στο reporting.conf ενεργοποιούμε το html report το Json το PDF και mongoDB

- enabled = yes

Στην συνέχεια κάνουμε update τα cuckoo signatures με την εντολή στο terminal cuckoo community και ενεργοποιούμε το Cuckoo με τις παρακάτω εντολές :

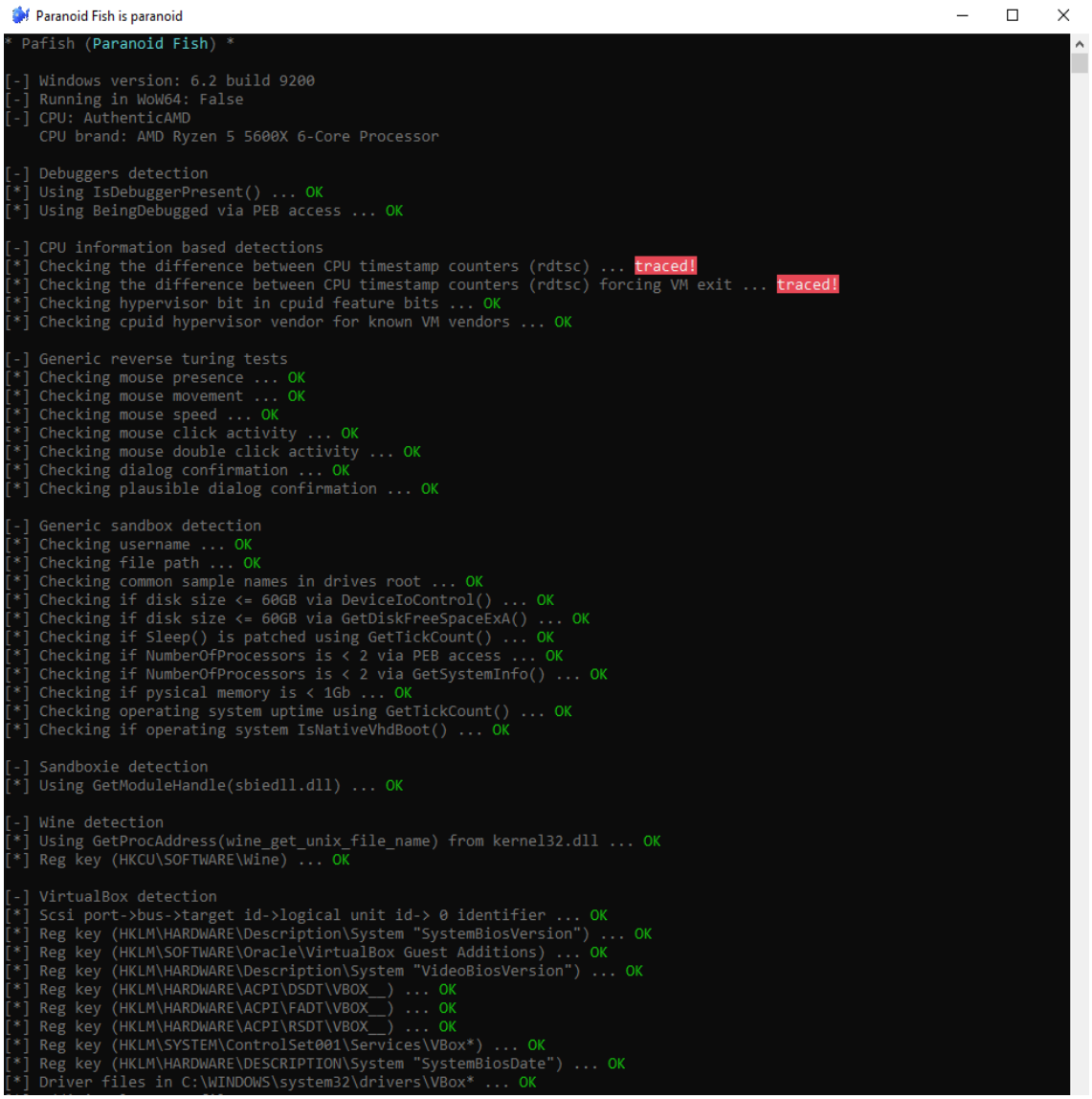
1. sudo cuckoo rooter -g cuckoo
2. Ξεκινάμε το εικονικό μας μηχάνημα
3. cuckoo
4. cuckoo web --host 127.0.0.1 --port 8080
5. 127.0.0.1:8080

Πριν ξεκινήσουμε όμως το Cuckoo θα κάνουμε μερικές ενέργειες ώστε να βελτιώσουμε το εικονικό μας μηχάνημα ώστε να μην φαίνεται τόσο ότι είναι εικονικό ώστε να αποφύγουμε μερικούς μηχανισμούς ασφαλείας που έχουν τα Ransomware και καταλαβαίνουν το περιβάλλον που εκτελούνται ώστε να μην εμφανίσουν όλα τους τα χαρακτηριστικά. Παρακάτω βλέπουμε μια εικόνα [2] που δείχνει πως αντιδρά ένα κακόβουλο λογισμικό στο περιβάλλον του όπου με τον ίδιο τρόπο λειτουργούν και τα Ransomware.



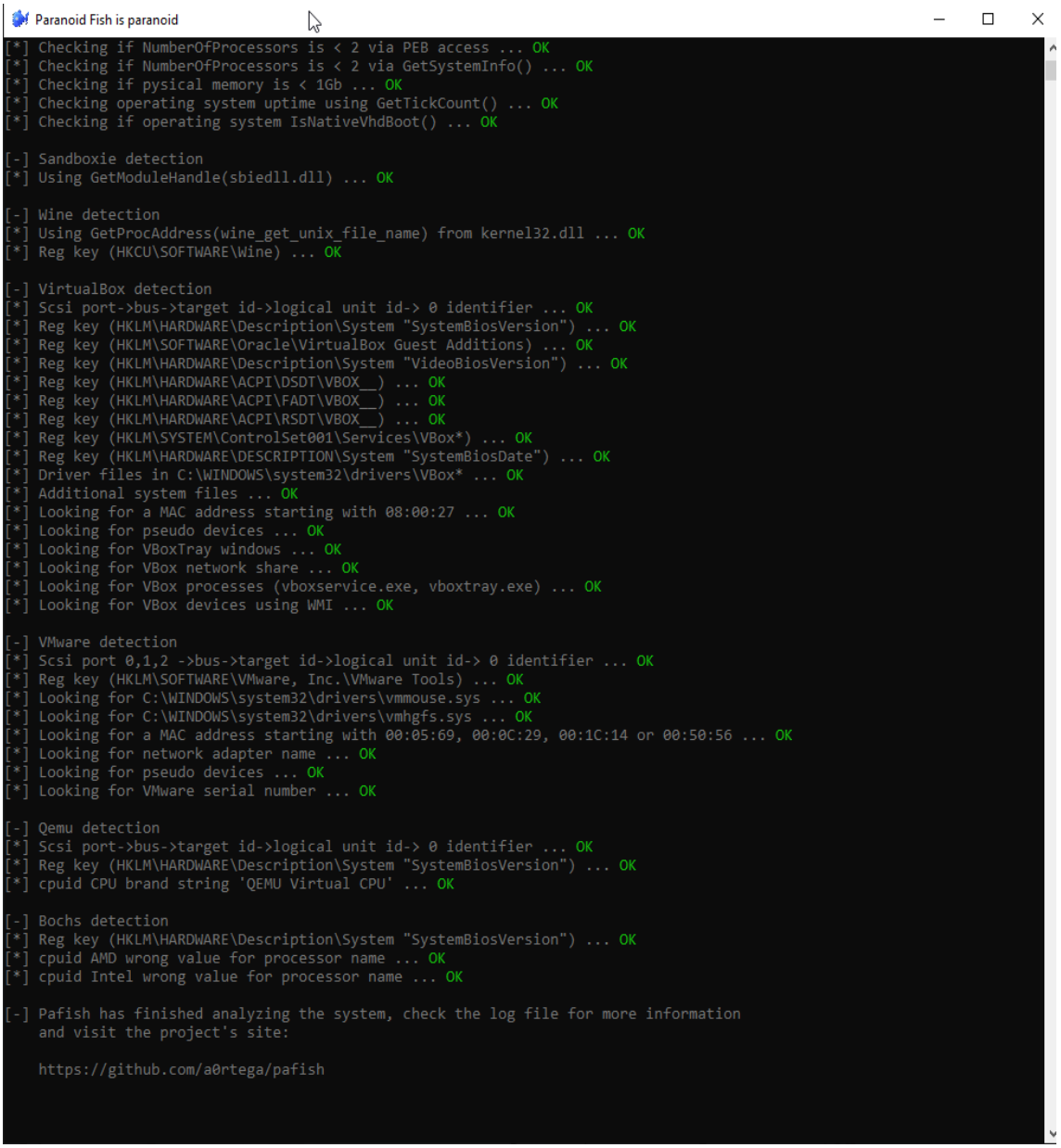
Εικόνα 1: Ανίχνευση Περιβάλλοντος

Οι ενέργειες που κάναμε για την απόκρυψη του εικονικού μας μηχανήματος ήταν αρχικά να μην χρησιμοποιήσουμε το pack Guest Additions του Virtual Box καθώς κάνουν το εικονικό μας μηχάνημα ευάλωτο ως προς την αναγνώριση του από το Ransomware. Έπειτα από τον σύνδεσμο https://github.com/d4rksystem/VBoxCloak κατεβάσαμε το VBoxCloak που μας βοήθησε στο να κρύψουμε το εικονικό μας μηχάνημα. Στην συνέχεια από τον σύνδεσμο https://github.com/a0rtega/pafish κατεβάσαμε το Pafish [3] που κάνει περεταίρω έλεγχο σε σημεία που μπορούν να αποκαλύψουν την εικονική μηχανή μας στο Ransomware και μας βοηθάει στην διόρθωσή τους. Παρακάτω δείχνουμε εικόνες ώστε να κατανοήσουμε την λειτουργία του Pafish, όπου δείχνουν τα αποτελέσματα του Pafish αφού εκτελέσαμε διάφορες λειτουργίες για την διόρθωση του συστήματός μας.



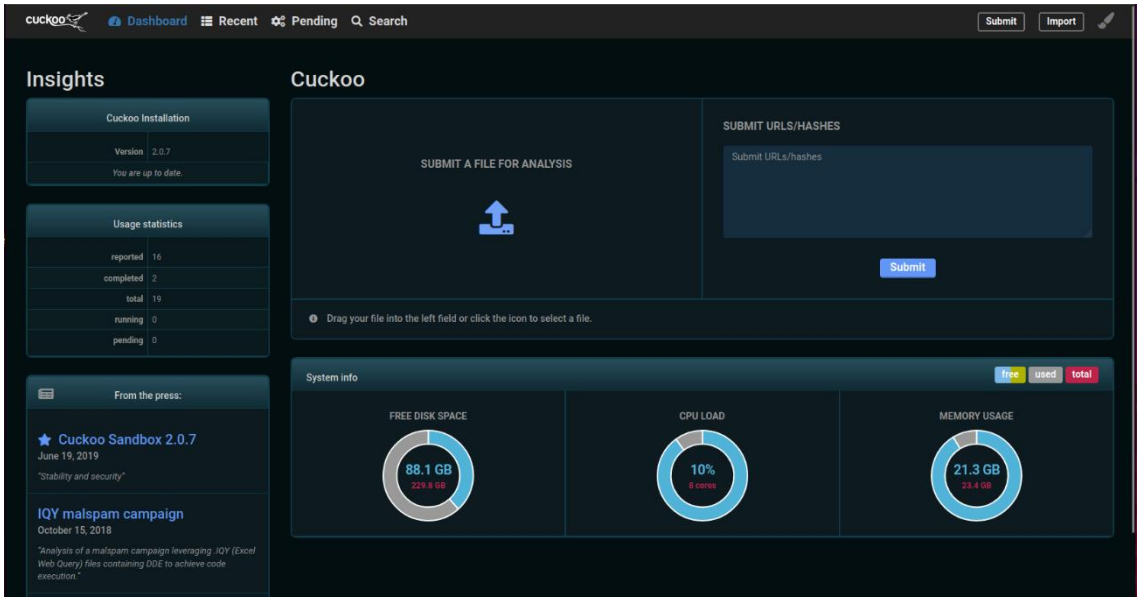
Εικόνα 2: Pafish 1

Παρατηρούμε πως ακόμα υπάρχουν δύο εντοπισμοί από το Pafish σχετικά με το rdtsc που δείχνουν δίπλα με κόκκινο φόντο το αναγραφόμενο traced!. Πριν εκτελέσουμε διάφορες λύσεις πολλές από τις ανιχνεύσεις είχαν το traced διπλά. Παρόλες τις προσπάθειές μας δεν καταφέραμε να βγάλουμε αυτά τα δύο traced και μέσω επικοινωνίας που είχαμε με τον δημιουργό του Pafish ενημερωθήκαμε ότι δεν γνωρίζει να υπάρχει λύση για αυτό το πρόβλημα με την χρήση Virtual Box. Οπότε συνεχίσαμε εις γνώσης μας για την αδυναμία αυτή του συστήματος μας. Με αυτά ως δεδομένα προχωράμε στην αποθήκευση του εικονικού μας μηχανήματος και είμαστε έτοιμοι πλέον για να ξεκινήσουμε την διαδικασία των αναλύσεων



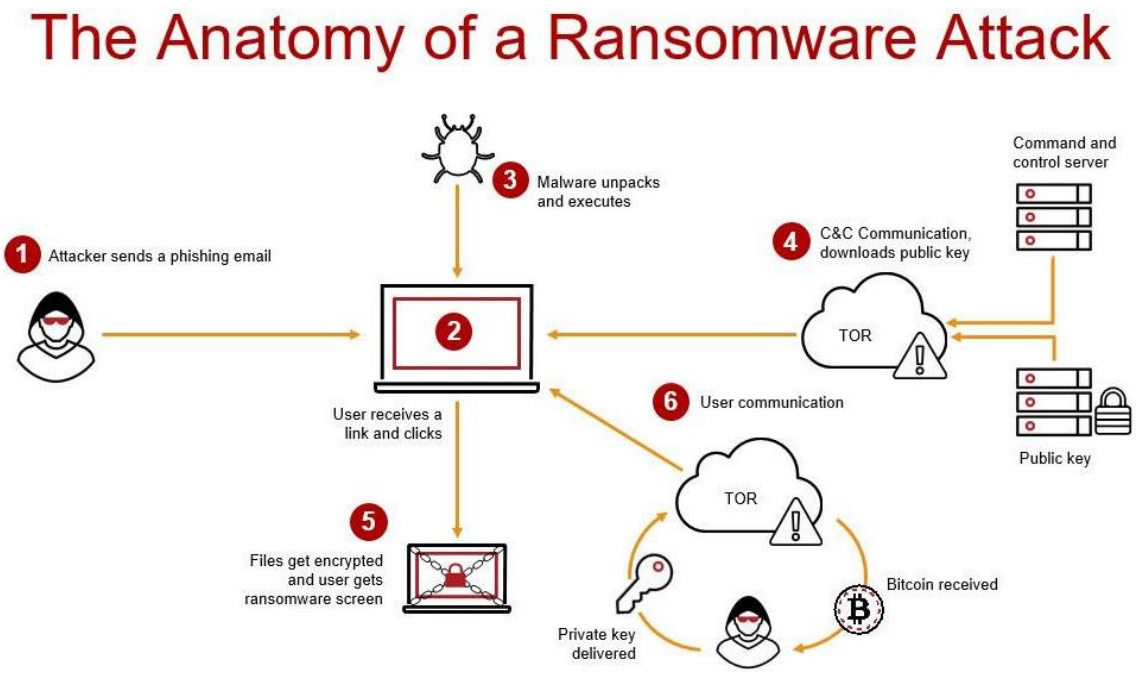
Εικόνα 3: Pafish 2

Αφού λοιπόν ξεκινήσαμε το Cuckoo με τις εντολές που προανέφερα ανοίγουμε τον browser μας πληκτρολογούμε το 127.0.0.1:8080 και βλέπουμε το γραφικό του περιβάλλον. Έχοντας έτοιμα τα Ransomware εκτελέσιμα επιλέγουμε το πρώτο και πατάμε το submit μετά Analyze και αναλόγως τις επιλογές ανάλυσης που έχουμε κάνει περιμένουμε να ολοκληρωθεί η διαδικασία δυναμικής ανάλυσης. Επιπρόσθετα μπορούμε την ώρα που γίνεται η ανάλυση να ανοίξουμε το εικονικό μηχάνημα και να δούμε τις αλλαγές στο σύστημα σε πραγματικό χρόνο.



Εικόνα 4: Cuckoo Environment

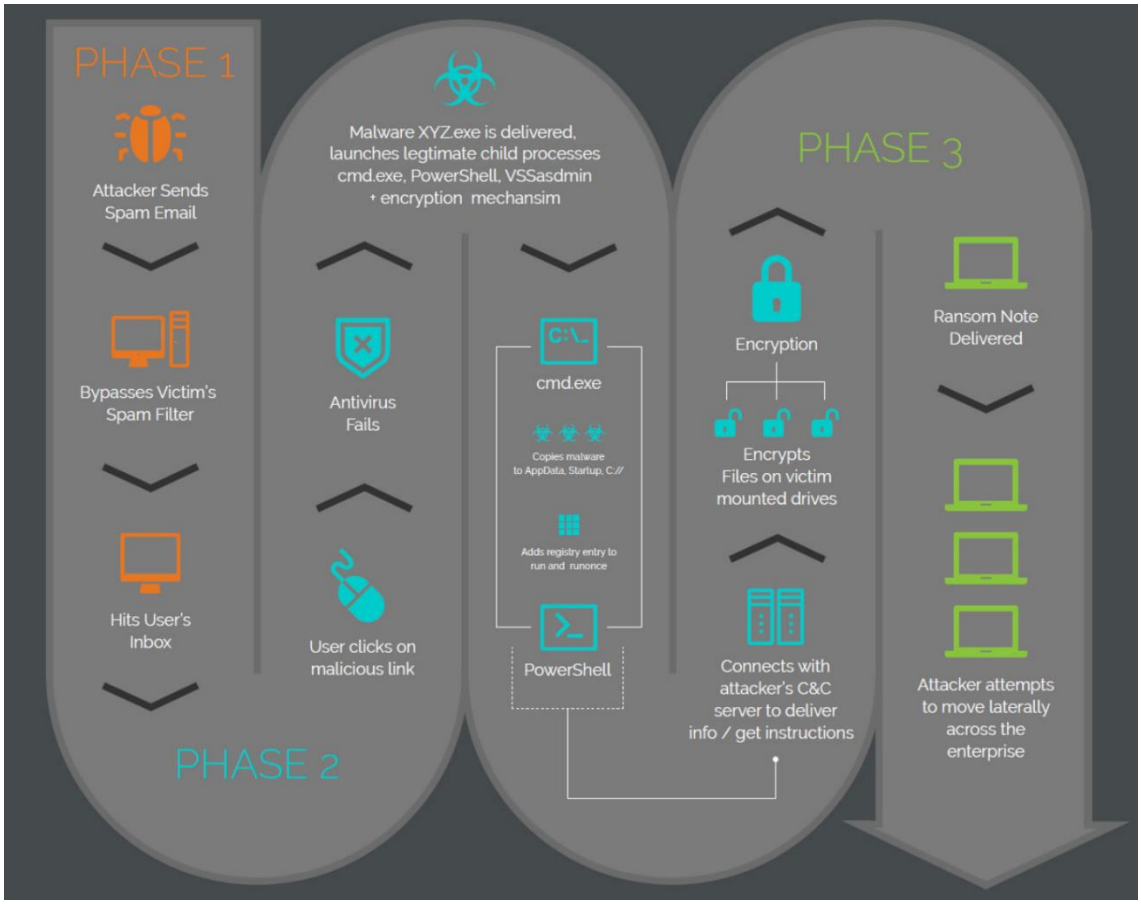
Από την διεξαγωγή των αυτόματων δυναμικών αναλύσεων που εκτελέσαμε μέσω του Cuckoo Sandbox και των δυνατοτήτων που μας προσφέρει πήραμε τα αποτελέσματα της ανάλυσης σε διάφορες μορφές όπως Html, Pdf, Json. Ενδιαφερόμαστε κυρίως για την Json μορφή, καθώς εκεί εμπεριέχεται πολύτιμη πληροφορία που θα αναλύσουμε στην συνέχεια στα διακόσια αποτελέσματα, σε αντίθεση βέβαια με τις άλλες μορφές που μας παρουσιάζουν την ανάλυση σε μια πιο φιλική προς τον χρήστη μορφή αλλά στερούνται πληροφορίας. Εδώ παρουσιάζουμε μια εικόνα [4] σχετικά με την ανατομία μιας επίθεσης Ransomware ώστε να κατανοήσουμε καλύτερα το τι θα δούμε στην συνέχεια.



Εικόνα 5 Ανατομία επίθεσης Ransomware

Στην εικόνα πέντε βλέπουμε σε ένα γενικευμένο και σύντομο πλάνο των κύριων λειτουργιών μιας επίθεσης από Ransomware. Η επίθεση ξεκινάει με τον επιτιθέμενο να βρίσκει ένα τρόπο να εισβάλει στο σύστημα μας. Εδώ συγκεκριμένα βλέπουμε πως χρησιμοποιεί την τεχνική phishing email όπου στέλνει ένα σύνδεσμο με το ιομορφικό λογισμικό. Στο δεύτερο βήμα ο χρήστης κάνει click τον σύνδεσμο και κατεβάζει το κακόβουλο αρχείο εν αγνοία του. Συνεχίζοντας στο τρίτο βήμα ανοίγει το αρχείο και αρχίζει η διαδικασία μόλυνσης του συστήματος όπως απενεργοποίηση του antivirus και σχετικών λειτουργιών, κλιμάκωση προνομίων (privilege escalation), μονιμότητα (persistence). Στο τέταρτο βήμα βλέπουμε σύνδεση με τον επιτιθέμενο μέσω Command and Control Server μέσω του δικτύου Tor και κατέβασμα του δημόσιου κλειδιού του για την χρήση του στην κρυπτογράφηση. Στο πέμπτο βήμα τα αρχεία κρυπτογραφούνται και παρουσιάζεται το μήνυμα πληρωμής λύτρων στον χρήστη. Τέλος γίνεται η πληρωμή των λύτρων από το θύμα σε κάποιο κρυπτονόμισμα όπως το Bitcoin, και ο επιτιθέμενος δίνει το ιδιωτικό κλειδί του στο θύμα για την αποκρυπτογράφηση των αρχείων του.

Επιπρόσθετα βλέπουμε μια εικόνα [5] που δίνει περισσότερη έμφαση στον εσωτερικό μηχανισμό ενός συστήματος όταν δέχεται επίθεση από Ransomware μέσω τριών φάσεων.



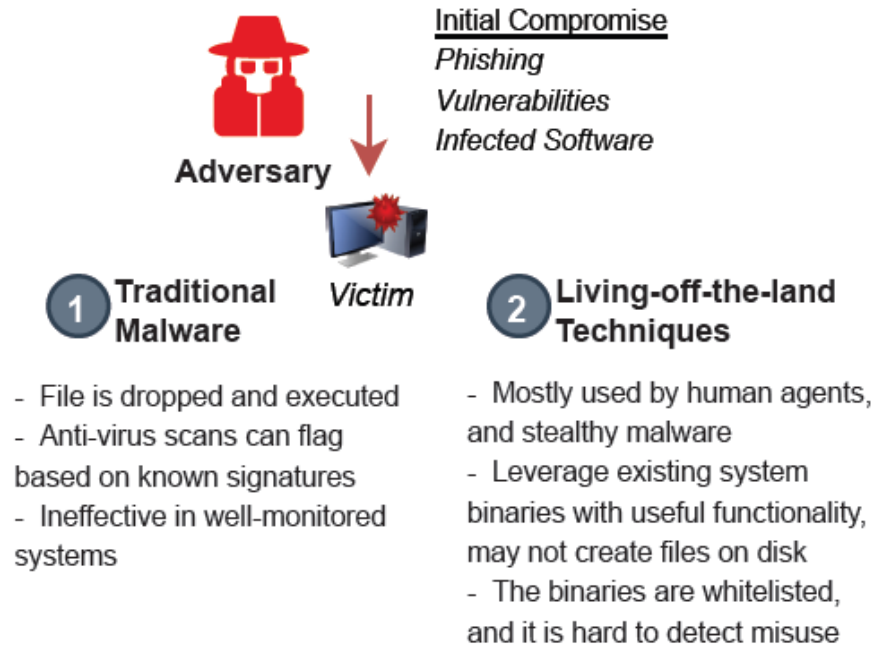
Εικόνα 6: Εσωτερική Ανατομία Επίθεσης Ransomware

Ξεκινώντας έρχεται το email από τον επιτιθέμενο και περνάει το φίλτρο για τα spam email. Στην δεύτερη φάση γίνεται click στο κακόβουλο link. Το αρχείο κατεβαίνει στον υπολογιστή του θύματος καθώς το antivirus αποτυγχάνει να το αναγνωρίσει. Από αυτό το σημείο ξεκινάει η δράση του Ransomware δημιουργώντας διεργασίες ώστε να εκτελέσει τις πολλαπλές λειτουργίες του μέσω του Cmd, Powershell, Vssadmin και πολλών άλλων. Όπως βλέπουμε στην εικόνα μέσω του Cmd αντιγράφει το κακόβουλο λογισμικό στο φάκελο Appdata, Startup ώστε να ξεκινάει με το ξεκίνημα του συστήματος. Μέσω του Powershell γίνεται σύνδεση με τον Command and Control Server ώστε να παίρνει οδηγίες και να κλαπούν πληροφορίες του θύματος. Μέσω του Vssadmin γίνεται η διαγραφή των αντιγράφων ασφαλείας όπως θα δούμε και στα επόμενα κεφάλαια, και στο τέλος της δεύτερης φάσης αρχίζει η διαδικασία της κρυπτογράφησης. Στην τελευταία φάση

παρουσιάζεται το μήνυμα πληρωμής λύτρων και γίνονται απόπειρες για την επέκταση του λογισμικού σε περισσότερα μηχανήματα του οργανισμού για μεγιστοποίηση της ζημιάς.

1.3 Ανάλυση Αποτελεσμάτων για τον εντοπισμό Living Off the Land Binaries, Scripts and Libraries.

Στα συστήματα Windows υπάρχουν πολλά αρχεία κώδικα που είναι προ-εγκατεστημένα ή εγκαθίστανται κατά την διάρκεια χρήσης του συστήματος από την ίδια την Microsoft. Τα αρχεία αυτά είναι μέρος του λειτουργικού συστήματος οπότε σαφώς και δεν θεωρούνται επικίνδυνα. Ωστόσο μερικά από αυτά αν εκτελεστούν με τον κατάλληλο τρόπο μπορούν να παρουσιάσουν κακόβουλη συμπεριφορά και λόγο του ότι είναι υπογεγραμμένα προσπερνούν κάθε μηχανισμό ασφαλείας όπως το User Account Control, Antivirus, EDR, EPP. Τα αρχεία αυτά μπορούν να εκτελέσουν αυθαίρετο κώδικα, να κάνουν λήψη και φόρτωση αρχείων, να συλλέξουν αρχεία διαπιστευτηρίων και διαδικασιών, και όλα αυτά χωρίς να ζητηθεί καμία αλληλεπίδραση με τον χρήστη. Αυτά τα αρχεία είναι γνωστά ως Living Off The Land Binaries and Scripts (LOLBAS). Η έρευνα που διεξάγουμε στα LOLBAS είναι πολύ σημαντική καθώς οι επιθέσεις που κάνουν χρήση τους είναι πολύ δύσκολο να εντοπιστούν και γι' αυτό τον λόγο χρησιμοποιούνται από επιθετικές ομάδες για να εκτελέσουν το κακόβουλο φορτίο και να μολύνουν τον κεντρικό υπολογιστή με ανεμπόδιστο και κρυφό τρόπο [6]. Στην εικόνα που ακολουθεί [7] δείχνουμε τις διαφορές χρήσης παραδοσιακών κακόβουλων λογισμικών σε σχέση με την χρήση LOLBAS αρχείων όσον αφορά την ανίχνευση τους από τα σημερινά συστήματα.



Εικόνα 7: Σύγκριση μεθόδων Malware με και χωρίς LOLBAS τεχνικές

Συνεχίζοντας αναφέρουμε τον τρόπο με τον οποίο εντοπίζουμε τα LOLBAS στα αποτελέσματα των Json αρχείων που πήραμε από την ανάλυση μέσω του Cuckoo Sandbox. Παρακάτω θα αναλύσουμε, γιατί είναι σημαντικά τα LOLBAS [8], [7], πως χρησιμοποιεί τα LOLBAS το κάθε ransomware ξεχωριστά. Για τον ταχύτερο εντοπισμό των LOLBAS αρχείων θα κάνουμε χρήση της παρακάτω εντολής που δημιουργήσαμε και μας αναφέρει τα ονόματα των LOLBAS αρχείων που κάνει χρήση το συγκεκριμένο Ransomware και σε ποιον αριθμό γραμμής χρησιμοποιούνται. Άρα πηγαίνουμε στο φάκελο που έχουμε το κάθε Json αποτέλεσμα και ανοίγοντας το τερματικό σε αυτόν τον φάκελο πληκτρολογούμε την παρακάτω εντολή. Αφού γίνει

ο εντοπισμός τους μπορούμε μετά να δούμε ολόκληρη την λειτουργικότητα τους μέσα στο Json αρχείο του εκάστοτε Ransomware.

Πληροφορίες Εντολής

- -i Απενεργοποιεί το Case Sensitive
- -w Δείχνει μόνο τα αποτελέσματα που αναγράφουν ακριβώς την λέξη και δεν εμπεριέχονται σε άλλες λέξεις.
- -n Δείχνει τον αριθμό της γραμμής που βρέθηκε η λέξη.

Εντολή

```
grep -iwn
'AppInstaller.exe\|Aspnet_Compiler.exe\|At.exe\|Atbroker.exe\|Bash.exe\|Bitsadmin.exe\|CertOC
.exe\|CertReq.exe\|Certutil.exe\|Cmd.exe\|Cmdkey.exe\|cmdl32.exe\|Cmstp.exe\|ConfigSecurity
Policy.exe\|Control.exe\|Csc.exe\|Cscript.exe\|DataSvcUtil.exe\|Desktopimgdownldr.exe\|Dfsvc.
exe\|Diantz.exe\|Diskshadow.exe\|Dllhost.exe\|Dnscmd.exe\|Esentutl.exe\|Eventvwr.exe\|Expan
d.exe\|Explorer.exe\|Extexport.exe\|Extrac32.exe\|Findstr.exe\|Finger.exe\|fltMC\|Forfiles.exe\|Ft
p.exe\|GfxDownloadWrapper.exe\|Gpscript.exe\|Hh.exe\|IMEWDBLD\|Ie4uinit.exe\|Ieexec.exe\|I
lasm.exe\|Infdefaultinstall.exe\|Installutil.exe\|Jsc.exe\|Makecab.exe\|Mavinject.exe\|Microsoft.W
rokflow.Compiler.exe\|Mmc.exe\|MpCmdRun.exe\|Msbuild.exe\|Msconfig.exe\|Msdt.exe\|Mshta.
exe\|Msiexec.exe\|Netsh.exe\|Odbcconf.exe\|OfflineScannerShell.exe\|OneDriveStandaloneUpd
ater.exe\|Pcalua.exe\|Pcwrun.exe\|Pktmon.exe\|Pnputil.exe\|Presentationhost.exe\|Print.exe\|Pri
ntBrm.exe\|Psr.exe\|Rasautou.exe\|Reg.exe\|Regasm.exe\|Regedit.exe\|Regini.exe\|Register-
cimprovider.exe\|Regsvcs.exe\|Regsvr32.exe\|Replace.exe\|Rpcping.exe\|Rundll32.exe\|Runon
ce.exe\|Runscripthelper.exe\|Sc.exe\|Schtasks.exe\|Scriptrunner.exe\|SettingSyncHost.exe\|Sto
rdiag.exe\|SyncAppvPublishingServer.exe\|Ttdinject.exe\|Tttracer.exe\|vbc.exe\|Verclsid.exe\|W
ab.exe\|Wlrmdr.exe\|Wmic.exe\|WorkFolders.exe\|Wscript.exe\|Wsreset.exe\|wuauclt.exe\|Xwiz
ard.exe\|Advpack.dll\|Dfshim.dll\|Ieadvpack.dll\|Ieaframe.dll\|Mshtml.dll\|Pcwutl.dll\|Setupapi.dll\|
Shdocvw.dll\|Shell32.dll\|Syssetup.dll\|Url.dll\|Zipfldr.dll\|Comsvcs.dll\|adplus.exe\|AgentExecutor
.exe\|Appvlp.exe\|Bginfo.exe\|Cdb.exe\|coregen.exe\|csi.exe\|DefaultPack.EXE\|Devtoolslaunch
er.exe\|dnx.exe\|Dotnet.exe\|Dxcap.exe\|Excel.exe\|Fsi.exe\|FsiAnyCpu.exe\|Mftrace.exe\|Msde
ploy.exe\|msxsl.exe\|ntdsutil.exe\|Powerpnt.exe\|Procdump(64).exe\|rcsi.exe\|Remote.exe\|Sqld
umper.exe\|Sqlps.exe\|SQLToolsPS.exe\|Squirrel.exe\|te.exe\|Tracker.exe\|Update.exe\|VSIISE
xeLauncher.exe\|VisualUiaVerifyNative.exe\|vsjitdebugger.exe\|Wfc.exe\|Winword.exe\|Wsl.exe\
|CL_LoadAssembly.ps1\|CL_Mutexverifiers.ps1\|CL_Invocation.ps1\|Manage-
bde.wsf\|Pubprn.vbs\|Syncappvpublishingserver.vbs\|UtilityFunctions.ps1\|winrm.vbs\|Pester.bat
' report.json
```

2 Conti

2.1 Living Off the Land Binaries, Scripts and Libraries

Στα αποτελέσματα των αναλύσεων των δειγμάτων με sha256 Hash :

1) 1b8081bae0e493d098b8756b1e7c4b19715a78946cf227f2c27f9311e6718420
2) 1ce8a939b3e7d84c59c12dc9e1091532f4336dac533847b6533b01d9dcf494e9
3) 2afea8912f1b7afa3a4348ef4e027f7a46f4a2ade824196265ea1ac952e172b3
4) 002dd9b7cbf8ca2a09434f8c4abd85631efe922ab8daa1219d86d83a0228aeda
5) 3a6e8f5a226edc006e29e38c48eaa721c12e48953abd56dcbc9b241975631247
6) 4d8fcd9492a256c0cb9567e63b53b75c2a591f5eebe7293298239736034b742a
7) 5d8a701110d58ab7c1aa8bae6bc9d5358b8cd508115891320e6af6c68f3bbd74
8) 5d4350bcfecb0746c8aebf6f31052ea95b1901d558e27215bb0197e50f3b9e5c
9) 53b1c1b2f41a7fc300e97d036e57539453ff82001dd3f6abf07f4896b1f9ca22
10) 98a09f7896a7c20229e696d6e8344fe9593fd70afada5d986e04c0d6933cc4db
11) 198667b1eda010a431dfb051a101cc73ead1d45ba8d0f6641ec1c14bca4106f3
12) 917788f1d9fd2664f18414faec3244a17c7d7ec29296b14b22cf19be90c95df4
13) ae90567ca1a3f4dfc430ae9d6cfa5139385c418b6b70f01c2dd4931dc76ff97b
14) b8ae41c1122afd180d8dbc011866233945edcb0f19a3f43a1d1033709279cf32

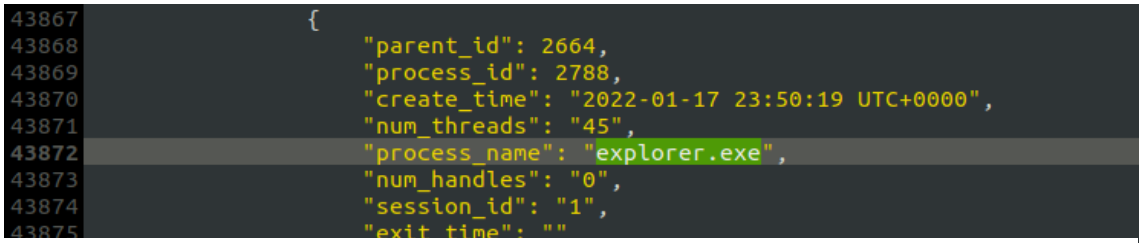
Χρησιμοποιώντας την παραπάνω εντολή βρίσκουμε εύκολα ότι γίνεται χρήση των παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στην συνέχεια θα εξετάσουμε την λειτουργικότητα του καθενός με την σειρά ώστε να κατανοήσουμε αν χρησιμοποιούνται για κακόβουλο σκοπό μέσα στο Json αρχείο ξεκινώντας από το πεδίο processes.

Explorer.exe

Το Explorer.exe είναι ένα στοιχείο γραφικού κελύφους που διαχειρίζεται την εμφάνιση διαχείρισης αρχείων που επιτρέπει στους χρήστες να ανοίγουν, να αντιγράφουν, να αποκόπτουν, να διαγράφουν, να μετακινούν και να εκτελούν άλλες ενέργειες με αρχεία στον υπολογιστή. Το αρχείο explorer.exe βρίσκεται στο φάκελο C:\Windows και έχει συνήθως μόνο μερικά byte σε μέγεθος. Στην παρακάτω εικόνα βλέπουμε μερικές πληροφορίες όπως το process identification number (PID) και parent identification number (PPID).



Εικόνα 8: Explorer.exe

Η διεργασία explorer.exe χρησιμοποιείται επίσης για να σταματά διεργασίες, υπηρεσίες και προγραμματισμένες εργασίες [9]. Στην συγκεκριμένη περίπτωση για την ανίχνευση και την διακοπή διεργασιών γίνεται χρήση του API Process32NextW από την διεργασία Explorer.exe .

```
10582                {
10583                    "category": "process",
10584                    "status": 1,
10585                    "stacktrace": [],
10586                    "api": "Process32NextW",
10587                    "return_value": 1,
10588                    "arguments": {
10589                        "process_name": "explorer.exe",
10590                        "snapshot_handle": "0x0000026c",
10591                        "process_identifier": 2788
10592                    },
10593                    "time": 1644187504.828575,
10594                    "tid": 5704,
10595                    "flags": {}
10596                },
```


Εικόνα 9: Process32NextW

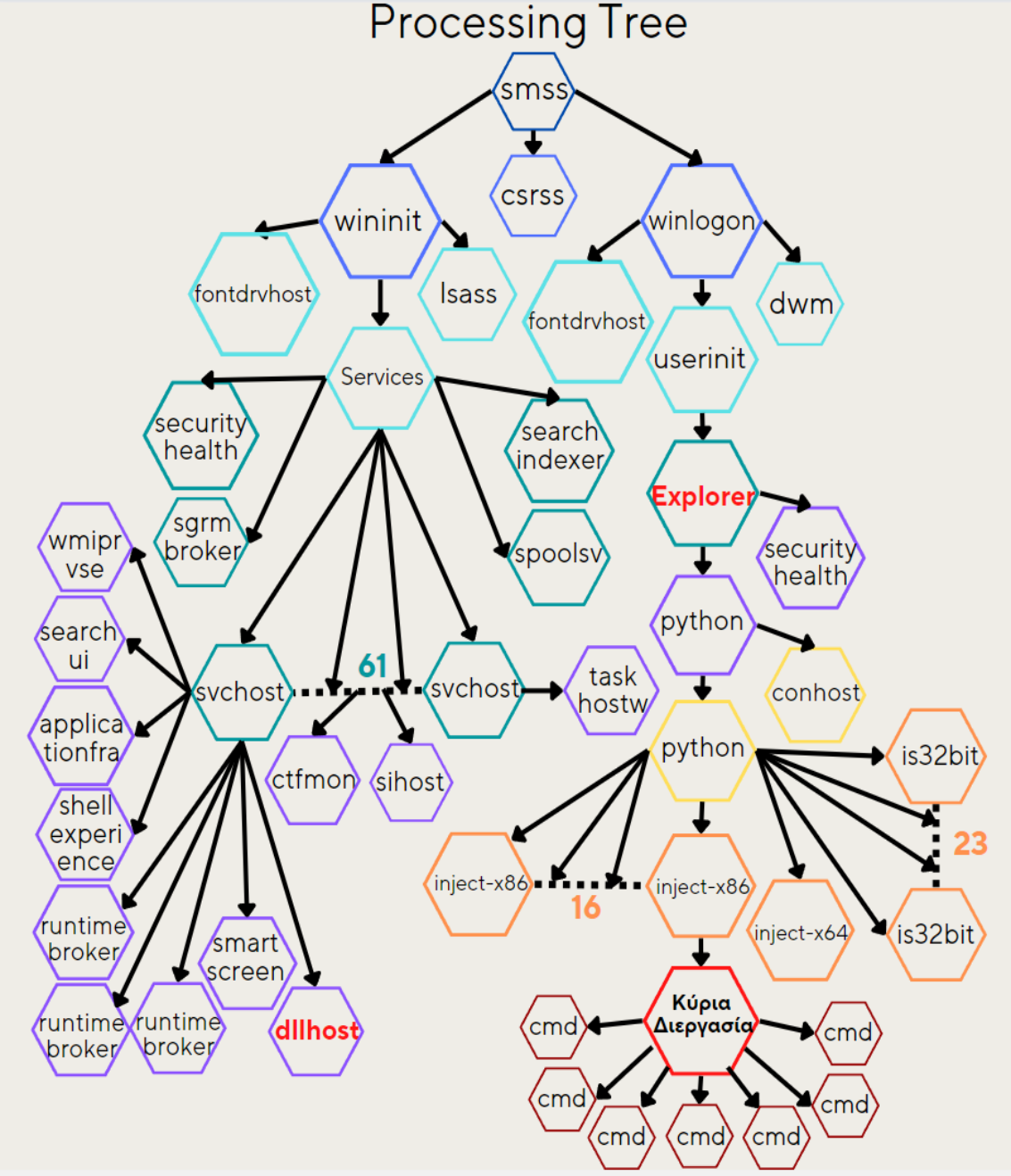
Λειτουργικότητα και Ροή Διεργασιών

Στην συνέχεια παίρνοντας παράδειγμα το δείγμα 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24 βρίσκουμε το όνομα της διεργασίας γονιός userinit.exe (2664) που εκκίνησε την διεργασία παιδί explorer.exe(2788) όπου είναι μια φυσιολογική διαδικασία των Windows λειτουργικών συστημάτων και παρουσιάζεται σε όλα τα δείγματα των Ransomware που αναλύουμε καθώς εκκινεί την διεργασία του αρχείου agent.py που χρειάζεται το Cuckoo για την λειτουργία του. Με την σειρά της η userinit.exe(2664) έχει ως PPID την winlogon(588) και αυτή έχει ως PPID την smss.exe που ενεργοποιείται όταν πατήσουμε το κουμπί εκκίνησης στον υπολογιστή μας και μετά σταματά, οπότε αρκετά εργαλεία ανίχνευσης διεργασιών δεν θα την εντοπίσουν. Από την αντίθετη κατεύθυνση πάμε να δούμε ποιες διεργασίες έχουν ως διεργασία γονιό την explorer.exe. Γράφοντας στην αναζήτηση "parent_id": 2788(pid του explorer.exe) βλέπουμε ότι καλούνται δύο διεργασίες η SecurityHealth(5076) και η python.exe(4240). Ακολουθώντας την 4240 βλέπουμε πάλι δύο διεργασίες την conhost(2576) και μια ακόμα python.exe(2316). Η 2316 βλέπουμε πως καλεί δεκαέξι διεργασίες με ονόματα inject-x86.exe, είκοσι τρείς is32bit.exe και μια inject-x64.exe που είναι πολύ ύποπτες. Συνεχίζοντας βρίσκουμε πως η κύρια διεργασία 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9 c24.exe(4684) έχει ως διεργασία γονιό την inject-x86.exe(3872). Η κύρια διεργασία έχει διεργασίες παιδιά επτά cmd.exe με PID (5172, 4664, 5444, 3604, 4520, 3588, 720). Όσο αφορά την λειτουργικότητα των cmd μπορούμε να την δούμε παρακάτω στο αντίστοιχο πεδίο. Σε μερικά δείγματα φαίνεται πως η ανάλυση σταματάει στις διεργασίες inject-x86.exe καθώς δεν μας δίνεται η πληροφορία τι καλείται στην συνέχεια.

Τα παραπάνω ισχύουν με μερικές διαφοροποιήσεις στα PID των διεργασιών τους στα Json αρχεία των δειγμάτων με hash sha256:

1) 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24
2) 4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9
3) 5d4350bcfecb0746c8aebf6f31052ea95b1901d558e27215bb0197e50f3b9e5c
4) 24ac73821de77cc9644d2ac40e97067ff63f625b5f20e085ad10535e47d7db59
5) 53b1c1b2f41a7fc300e97d036e57539453ff82001dd3f6abf07f4896b1f9ca22
6) 98a09f7896a7c20229e696d6e8344fe9593fd70afada5d986e04c0d6933cc4db
7) 6815e1e06e29863290319eb3e814ae2a394271aa2f95cc7c31a649c4c2f4fd04
8) b2d3143d0778a10d5d03bb9e4d2712a980e2a8ec12d47958a8ab4b3192f4bf6
9) b8ae41c1122afd180d8dbc011866233945edcb0f19a3f43a1d1033709279cf32

Παρακάτω παρατίθεται το Processing Tree που παρουσιάζει ένα διάγραμμα με τις πιο βασικές διεργασίες και τα LOLBAS exe. Ως κύρια διεργασία υπενθυμίζουμε ότι είναι η 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24.exe(4684).



Εικόνα 10: Processing Tree

Πολλές από τις διεργασίες του Process Tree θεωρούνται κακόβουλες καθώς βλέπουμε αρκετές παρεμβάσεις στα δικαιώματα χρήσης τους στον παρακάτω πίνακα.

Process (PID)	Privilege	Description
fontdrvhost.exe (744)	SeIncreaseWorkingSetPrivilege	Allocate more memory for user applications
dwm.exe (1000)	SeIncreaseBasePriorityPrivilege	Increase scheduling priority
	SeIncreaseWorkingSetPrivilege	Allocate more memory for user applications
python.exe (2316)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
is32bit.exe (All 23)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
Inject-x64.exe (4956)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
Inject-x86.exe (All 16)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
Κύρια Διεργασία (4684)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
Cmd.exe (All 7)	SeLoadDriverPrivilege	Load and unload device drivers
	SeDebugPrivilege	Debug programs
Lsass.exe(612)	SeCreateTokenPrivilege	Create a token object
fontdrvhost.exe (752)	SeIncreaseWorkingSetPrivilege	Allocate more memory for user applications
Svchost.exe (1704,1980,2444)	SeAuditPrivilege	Generate security audits

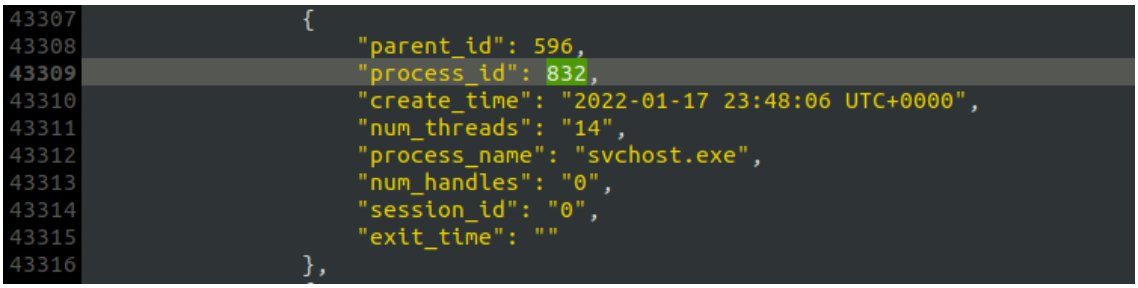
Dllhost.exe

Το Dllhost.exe ενεργοποιείται με την έναρξη των Windows και είναι μια νόμιμη διεργασία, η οποία ανήκει στο λειτουργικό σύστημα Microsoft Windows Operating System. Αυτή η εκτελέσιμη διεργασία είναι ευρέως γνωστή ως COM surrogate, η οποία διαχειρίζεται εφαρμογές τύπου DLL. Επίσης, παίζει σημαντικό ρόλο στον έλεγχο των διεργασιών στις Internet Information Services (IIS), φορτώνει το .NET περιβάλλον εκτέλεσης και παράλληλα συμμετέχει και σε άλλες σημαντικές δραστηριότητες του συστήματος. Συνήθως το μέγεθος αυτού του αρχείου είναι γύρω στα 5,12 bytes και βρίσκεται στον φάκελο C:\Windows\ System32.

```
{
    "parent_id": 832,
    "process_id": 6100,
    "create_time": "2022-01-17 23:51:32 UTC+0000",
    "num_threads": "5",
    "process_name": "dllhost.exe",
    "num_handles": "0",
    "session_id": "1",
    "exit_time": ""
},
```


Εικόνα 11: Dllhost.exe

Συχνά, τα malwares χρησιμοποιούν αρχεία με νόμιμα και έγκυρα ονόματα έτσι ώστε να κρυφτούν εντός του συστήματος. Το Dllhost.exe μπορεί επίσης να χρησιμοποιηθεί με αυτόν τον τρόπο για την ενεργοποίηση συγκεκριμένων παρασίτων ή μερών αυτών.



```
{
    "parent_id": 596,
    "process_id": 832,
    "create_time": "2022-01-17 23:48:06 UTC+0000",
    "num_threads": "14",
    "process_name": "svchost.exe",
    "num_handles": "0",
    "session_id": "0",
    "exit_time": ""
},
```


Εικόνα 12: Svchost.exe

Όπως βλέπουμε η διεργασία dllhost.exe(6100) έχει ως διεργασία γονέα την svchost.exe(832) και αυτή με την σειρά της την services.exe(596). Στην συνέχεια βλέπουμε πως η services.exe(596) καλέστηκε από την wininit.exe(496) και η wininit(496) από την smss(412) που δεν εντοπίζεται το όνομά της στο report παρά μόνο το PID της .Μετά την ανάλυση της εν λόγο διεργασίας συμπεραίνουμε πως δεν χρησιμοποιείται για κάποια κακόβουλη δραστηριότητα.

Shell32.dll, Setupapi.dll, Shdocvw.dll

Εντοπίστηκαν επίσης οι LOLBAS βιβλιοθήκες όπως shell32.dll, setupapi.dll, shdocvw.dll όπου χρησιμοποιήθηκαν αρκετές φορές από διάφορες διεργασίες όπως το svchost.exe και το explorer.exe χωρίς όμως η ανάλυση να μας δίνει πληροφορίες για την χρήση τους με ακρίβεια. Παρακάτω δίνεται ενδεικτικά για ποιους λόγους χρησιμοποιούνται αυτές οι βιβλιοθήκες.

- Shell32.dll: Είναι μια βιβλιοθήκη που περιέχει λειτουργίες του Windows Shell API, οι οποίες χρησιμοποιούνται κατά το άνοιγμα ιστοσελίδων και αρχείων.
- Setupapi.dll: Είναι μια βιβλιοθήκη που χρησιμοποιούν οι εγκαταστάτες και οι εφαρμογές εγκατάστασης. Παρέχει γενικές συναρτήσεις εγκατάστασης και συναρτήσεις εγκατάστασης συσκευών.
- Shdocvw.dll: Είναι ένα αρχείο Βιβλιοθήκης δυναμικής σύνδεσης που βρίσκεται στο "C:\Windows\System32", γνωστό ως βιβλιοθήκη αντικειμένων και ελέγχου του Shell Doc. Περιέχει βασικές λειτουργίες του προγράμματος περιήγησης ιστού Internet Explorer για πλοήγηση στο Διαδίκτυο, επιτόπια σύνδεση, διαχείριση αγαπημένων και ιστορικού

Στο δείγμα με Hash sha256 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd 3ef4add7c2837ddc67f9c24 του Conti που αναλύθηκε εμφανίστηκαν πάλι τα προαναφερθέντα εκτελέσιμα explorer.exe και Dllhost.exe με την ίδια λειτουργικότητα ακριβώς καθώς γίνεται και χρήση των shell32.dll, setupapi.dll και shdocvw.dll. Επιπρόσθετα όμως χρησιμοποιείται και η

διεργασία cmd.exe που είναι στην λίστα των LOLBAS και θα δούμε την δράση της παρακάτω αναλυτικά.

Cmd.exe

Μέσα στα αποτελέσματα της ανάλυσης του δείγματος με Hash sha256 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24 βλέπουμε την γενικότερη λειτουργία των διεργασιών cmd.exe που έχουν την ίδια λειτουργικότητα. Στις παρακάτω εικόνες βλέπουμε την δράση μιας εξ αυτών όπως το τι φάκελους άνοιξε, τι τιμές των registries διαβάστηκαν και την τιμή του PPID για να εντοπίσουμε πια διεργασία την κάλεσε. Όλες οι διεργασίες έχουν το ίδιο PPID που αναφέρεται στο κύριο εκτελέσιμο αρχείο.


```
{
    "process_path": "C:\\Windows\\SysWOW64\\cmd.exe",
    "process_name": "cmd.exe",
    "pid": 3588,
    "summary": {
        "file_failed": [
            "C:\\Windows\\SysWOW64\\en-US\\cmd.exe.mui"
        ],
        "file_exists": [
            "C:\\Users\\Z\\AppData\\Local\\Temp"
        ]
```


Εικόνα 13: Cmd.exe

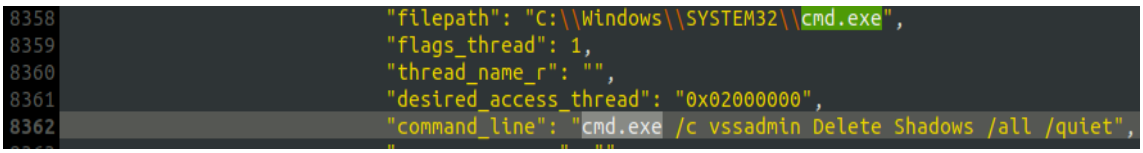

```
    ],
    "regkey_read": [
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\Language\\InstallLanguageFallback",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\DefaultColor",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\AutoRun",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\DelayedExpansion",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\EnableExtensions",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\CustomLocale\\en-US",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\PathCompletionChar",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\ExtendedLocale\\en-US",
        "HKEY_CURRENT_USER\\Control Panel\\Desktop\\PreferredUILanguages",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\MUI\\UILanguages\\en-US\\Type",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\DisableUNCCheck",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Session Manager\\ResourcePolicies",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\MUI\\UILanguages\\en-US\\AlternateCodePage",
        "HKEY_CURRENT_USER\\Control Panel\\Desktop\\MuiCached\\MachinePreferredUILanguages",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\CustomLocale\\EMPTY",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\WOW6432Node\\Microsoft\\Command Processor\\CompletionChar"
    ],
    "directory_enumerated": [
        "C:\\Windows\\System32\\vssadmin.*",
        "C:\\Users\\Z\\AppData\\Local\\Temp",
        "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\vssadmin.*",
        "C:\\Users\\Z\\AppData\\Local\\Temp\\vssadmin.*",
        "C:\\Windows\\System32\\OpenSSH\\vssadmin.*",
        "C:\\Program Files (x86)\\Common Files\\Oracle\\Java\\javapath\\vssadmin.*",
        "C:\\Users\\Z",
        "C:\\Users\\Z\\AppData",
        "C:\\Users",
        "C:\\Users\\Z\\AppData\\Local",
        "C:\\Windows\\vssadmin.*",
        "C:\\Users\\Z\\AppData\\Local\\Microsoft\\WindowsApps\\vssadmin.*",
        "C:\\Windows\\System32\\wbem\\vssadmin.*"
    ],
    "file_opened": [
        "C:\\Windows\\System32\\en-US\\cmd.exe.mui"
    ]
},
"first_seen": 1644184013.975584,
"ppid": 4684
```


Εικόνα 14: Cmd.exe Activity

Συνεχίζοντας μπορούμε να δούμε την δραστηριότητα σε βάθος από τις διεργασίες που καλεί (API Calls) [10]:

- NtOpenThread
- LdrGetDllHandle module_name
- NtOpenKey
- NtQueryValueKey
- NtClose
- NtEnumerateKey
- NtDeviceIoControlFile
- NtOpenKeyEx
- NtAllocateVirtualMemory
- GetFileAttributesW
- FindFirstFileExW
- GetFileType
- NtCreateFile
- NtCreateSection
- NtMapViewOfSection
- WriteConsolew
- NtTerminateProcess

Η πιο αξιοσημείωτη και κακόβουλη δραστηριότητα με την χρήση του cmd.exe είναι ότι δημιουργείται μια διεργασία μέσω της συνάρτησης NtCreateUserProcess όπου δίνεται η εντολή cmd.exe : /c vssadmin Delete Shadows /all /quiet. Αυτή η εντολή χρησιμοποιείται για την διαγραφή των Volume Shadow Copies δηλαδή τα backups αρχεία που διατηρούν τα Windows σε περίπτωση βλάβης του συστήματος.



Εικόνα 15: Delete Shadow Copies with Vssadmin

Με σκοπό πάλι την διαγραφή των αντιγράφων ασφαλείας είναι οι παρακάτω εντολές που εντοπίσαμε στην συνέχεια της ανάλυσής μας :

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded

Συνεχίζοντας στις παρακάτω αναλύσεις βλέπουμε πως γίνεται χρήση καινούριων LOLBAS που παρουσιάζουμε παρακάτω με μπλε χρωματισμό.

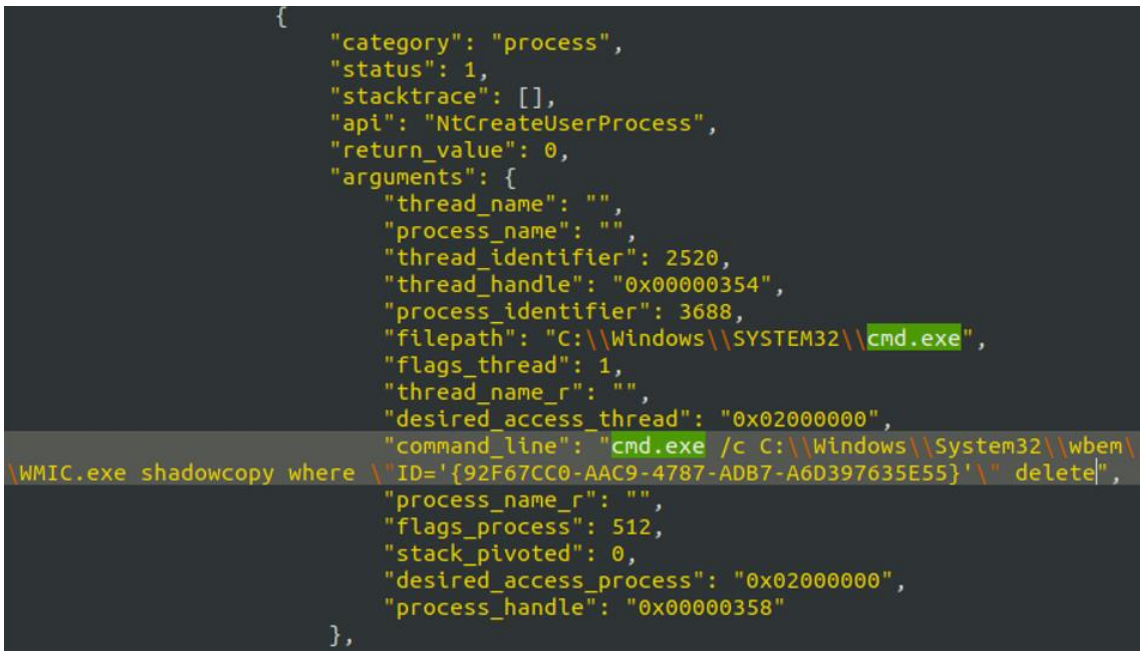
Δείγματα :

1) 4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9
2) 24ac73821de77cc9644d2ac40e97067ff63f625b5f20e085ad10535e47d7db59
3) 6815e1e06e29863290319eb3e814ae2a394271aa2f95cc7c31a649c4c2f4fd04
4) b2d3143d0778a10d5d03bb9e4d2712a980e2a8ec12d47958a8ab4b3192f4bf6a

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Cmd.exe	**ExtExport.exe**	**Wmic.exe**		

Wmic.exe

Η συμπεριφορά των explorer.exe και dllhost.exe είναι ακριβώς η ίδια με τις προηγούμενες αναλύσεις οπότε προχωράμε στην cmd.exe. Εδώ το εκτελέσιμο αρχείο έχει πολλά κοινά σημεία με προηγούμενες αναλύσεις, αλλά υπάρχει διαφορά καθώς η διεργασία που δημιουργείται για την διαγραφή των αντίγραφων shadow copies χρησιμοποιεί το εκτελέσιμο WMIC.exe και όχι το vssadmin.



Εικόνα 16: Delete Shadow Copies with WMIC.exe

ExtExport.exe

Το ExtExport.exe είναι ένα εκτελέσιμο αρχείο που αποτελεί μέρος του προγράμματος Internet Explorer που αναπτύχθηκε από τη Microsoft Corporation. Η χρήση του ExtExport.exe επιτρέπει στους εισβολείς να χρησιμοποιούν αυτό το νόμιμο βοηθητικό πρόγραμμα για να φορτώσουν οποιοδήποτε DLL που περνά στην ελεγχόμενη από τον εισβολέα διαδρομή. Στα συγκεκριμένα δείγματα γίνεται χρήση της συνάρτησης GetFileAttributesW για να ανακτηθούν τα χαρακτηριστικά επεξεργασίας και χρήσης του αρχείου και στις δύο παρακάτω διαδρομές :

- C:\\Program Files\\internet explorer\\ExtExport.exe
- C:\\Program Files (x86)\\Internet Explorer\\ExtExport.exe

Όπως προαναφέραμε το Wmic.exe χρησιμοποιείται για την διαγραφή αντιγράφων ασφαλείας. Τέλος τα LOLBAS DLL καλούνται αρκετές φορές από τις διεργασίες SecurityHealth, svchost.exe και explorer.exe.

Schtasks.exe

Στην ανάλυση με sha256 hash 2579148e5f020145007ac0dc1be478190137d7915e6fbca 2c787b55dbec1d370 παρουσιάζονται τα ίδια LOLBAS με της ανάλυσης 1b8081bae0e493d098b8 756b1e7c4b19715a78946cf227f2c27f9311e6718420 με την μόνη διαφορά ότι χρησιμοποιεί επιπρόσθετα το LOLBAS schtasks.exe μέσω της συνάρτησης LdrloadDll προφανώς για να προγραμματίσει την αυτόματη εκτέλεση κάποιας διεργασίας χωρίς όμως να δίνονται περεταίρω πληροφορίες.

2.2 Μηχανισμοί Λειτουργίας

Πριν προχωρήσουμε στην ανάλυση των μηχανισμών λειτουργίας θα δούμε τις διεργασίες που καλεί η κύρια διεργασία του δείγματος 4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9 για να πάρουμε ένα παράδειγμα για την λειτουργία τους. Το κύριο εκτελέσιμο έχει όνομα 4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9.exe και βλέπουμε πως δημιουργεί, ανιχνεύει απαριθμεί, διαβάζει, ανοίγει και γράφει σε αρκετούς φακέλους, αρχεία, αλλά και registry τιμές. Η κύρια διεργασία μας με PID 2648 βλέπουμε πως καλεί δύο διεργασίες cmd.exe με PID 3688 και 5852. Η 3688 καλεί την conhost.exe(348) και η 5852 καλεί πάλι μια conhost.exe(6048), και αυτό γίνεται καθώς για να τρέξει η cmd.exe χρειάζεται την conhost.exe. Όπως είδαμε και στην παράγραφο για το WMIC.exe το βλέπουμε και εδώ καθώς χρησιμοποιείται από το cmd.exe(3688) και την cmd.exe(5852) για την διαγραφή των αντιγράφων shadow copies.

```
                    "process_name": "cmd.exe",
                    "process_id": 3688,
                    "commandline": "cmd.exe /c C:\\Windows\\System32\\wbem\\WMIC.exe
shadowcopy where \"ID='{92F67CC0-AAC9-4787-ADB7-A6D397635E55}'\" delete",
```

Εικόνα 17: Delete Shadow Copies with WMIC.exe 2

Στην συνέχεια η κύρια διεργασία βλέπουμε πως κάνει process injection στην διεργασία συστήματος csrss.exe(508) και δραστηριοποιείται από εκεί.

```
1258                    "424 -> 2600/2604",
1259                    "508 -> 5076/5080",
1260                    "508 -> 2648/5704",
1261                    "424 -> 2716/2720",
1262                    "424 -> 760/764",
```

Εικόνα 18: Process Injection

Σε αυτό το σημείο θα αναλύσουμε τους μηχανισμούς λειτουργίας του προς ανάλυση Ransomware. Το Cuckoo Sandbox χρησιμοποιεί διάφορα Signatures στα αποτελέσματα της ανάλυσης για την αναγνώριση μοτίβων που έχουν να κάνουν με κάποια ύποπτη δραστηριότητα, και με αυτόν τον τρόπο μας ενημερώνει για την δραστηριότητα του συγκεκριμένου Ransomware μέσα στο δοκιμαστικό λειτουργικό σύστημα. Μπορούμε επίσης να δημιουργήσουμε δικά μας Signatures και να τα χρησιμοποιήσουμε στα αποτελέσματα τις ανάλυσης ώστε να δούμε αν το κακόβουλο λογισμικό εξυπηρετεί την συγκεκριμένη συμπεριφορά. Αναλύοντας ακριβείς συμπεριφορές μπορούμε να καθορίσουμε το είδος του κακόβολου λογισμικού και την οικογένεια που ανήκει όπως Adware, Botnet, Keylogger, Cryptocurrency miner, Trojan ή Ransomware. Επίσης μπορούμε να κατηγοριοποιήσουμε τα δείγματα προς εξέταση, και να αναγνωρίσουμε διάφορες τροποποιήσεις που μπορεί να έχουν γίνει ειδικότερα σε κάποιο δείγμα.

Το πρώτο Signature που μας δίνει η ανάλυση με hash sha256 1b8081bae0e493d098b8756b1e7c4b19715a78946cf227f2c27f9311e6718420 είναι ότι το εκτελέσιμο έχει μια PDB διαδρομή. Ένα αρχείο βάσης δεδομένων προγράμματος (PDB) περιέχει πληροφορίες εντοπισμού σφαλμάτων και κατάστασης έργου που επιτρέπουν τη σταδιακή σύνδεση μιας διαμόρφωσης εντοπισμού σφαλμάτων του προγράμματός. Όπως κάθε προγραμματιστής λογισμικού, οι δημιουργοί κακόβουλου λογισμικού συχνά πρέπει να διορθώσουν τον κώδικά τους και μερικές φορές καταλήγουν να δημιουργούν PDB ως μέρος της διαδικασίας ανάπτυξής τους. Εάν δεν αφιερώσουν χρόνο στον εντοπισμό σφαλμάτων του κακόβουλου λογισμικού τους, διατρέχουν τον κίνδυνο να μην λειτουργήσει σωστά στους οικοδεσπότες-θύματα ή να μην μπορέσουν να επικοινωνήσουν με επιτυχία με το κακόβουλο λογισμικό τους εξ αποστάσεως.

```
34            "description": "This executable has a PDB path",
```

Εικόνα 19: PDB Path

Το δεύτερο signature μας δείχνει ότι μια διεργασία διακόπηκε (crashed). Αυτή η διεργασία είναι το notification. Στην συνέχεια βλέπουμε ότι καλείται το api exception άρα καταλαβαίνουμε πως σκοπός αυτής της δραστηριότητας είναι να βάλει κάποια εξαίρεση στο εκτελέσιμο του notification ώστε να μην μας ενημερώσει για κάποια δραστηριότητα που θα θέλαμε να δούμε.

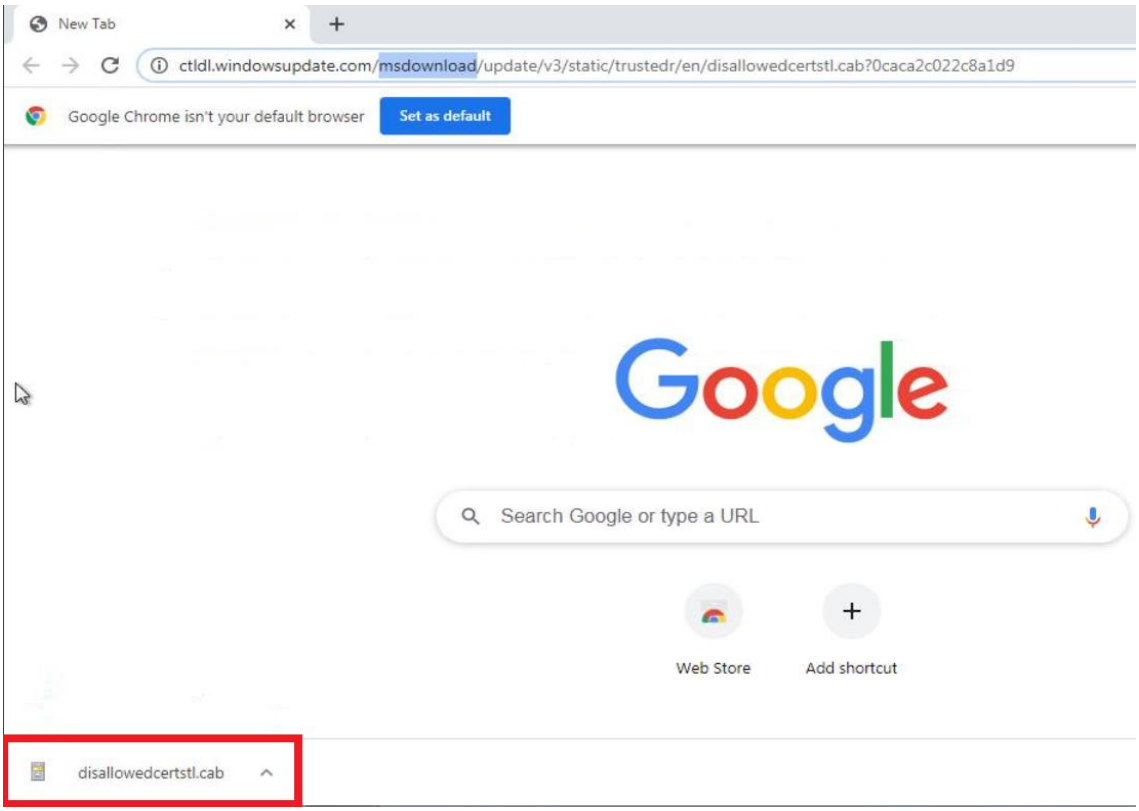
```
51            "description": "One or more processes crashed",
```

Εικόνα 20: Process Crashed

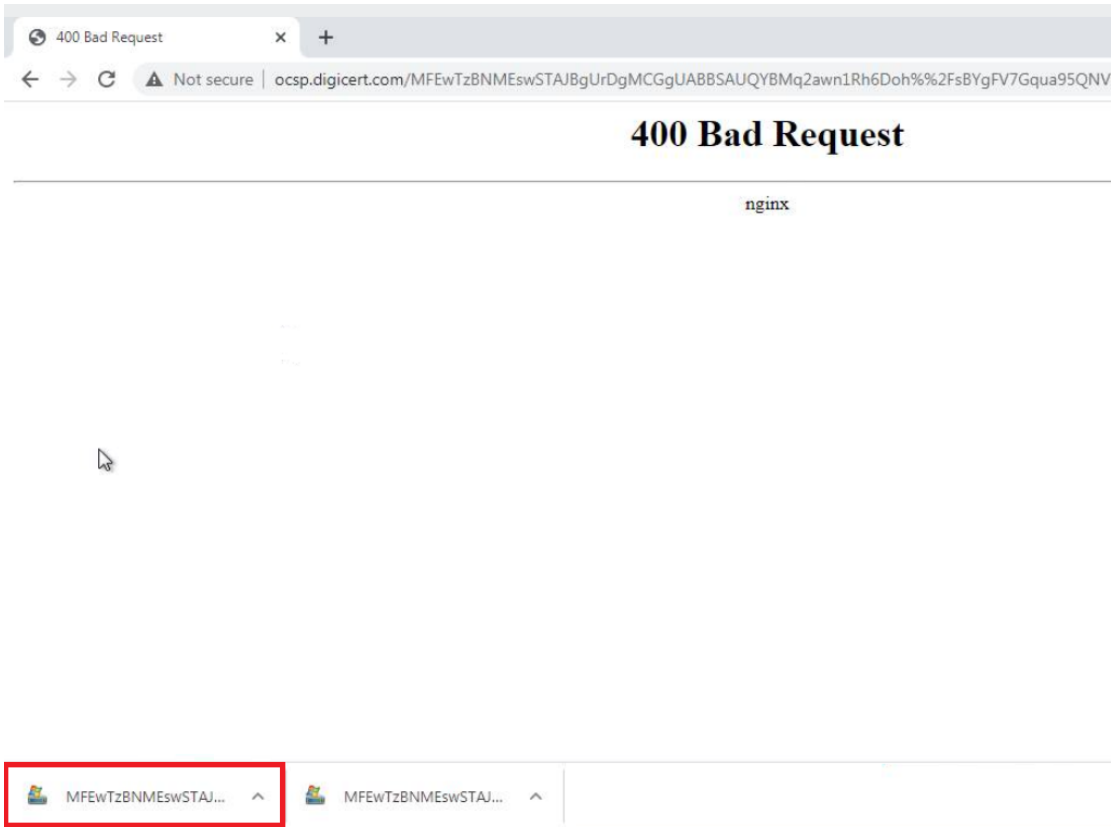
Συνεχίζοντας εκτελεί HTTP requests :

1. GET http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr /en/disallowedcertstl.cab?0caca2c022c8a1d9
2. GET http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGgUABBSAU QYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95QNVbRTLtm8KPiGxvDl7I90VUCEAJ0LqoXyo4hxx e7H%2Fz9DKA%3D"

Οι σύνδεσμοι κατεβάζουν αρχεία ονόματος disallowedcertstl.cab. και **Error! Hyperlink reference not valid.**CGgUABBSAUQYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95\QNVbRTLtm8KPiGxvDl7I90VUCEAJ0LqoXyo4hxxe7H%2Fz9DKA%3D. Τα CAB αρχεία ή αλλιώς Cabinet περιέχουν συμπιεσμένα αρχεία βιβλιοθήκης. Επίσης χρησιμοποιούνται για την οργάνωση της εγκατάστασης αρχείων που θα αντιγραφούν στο σύστημα του χρήστη. Το δεύτερο εμφανίζεται με όνομα file και δεν παρουσιάζονται επιπρόσθετες πληροφορίες. Τα αρχεία προφανώς συμμετέχουν στην λειτουργικότητα του ίο-μορφικού λογισμικού [11].



Εικόνα 21: Cab File



Εικόνα 22: Suspicious File

Ανιχνεύθηκε επίσης μια ξένη γλώσσα που δείχνει ότι είναι τα κινέζικα.


```
"families": [],
"description": "Foreign language identified in PE resource",
"severity": 2,
"ttp": {},
"markcount": 11,
"references": [],
"marks": [
    {
        "name": "RT_ICON",
        "language": "LANG_CHINESE",
        "offset": "0x00045ec0",
        "filetype": "Device independent bitmap graphic, 16 x 32 x 24, image size 0",
        "sublanguage": "SUBLANG_CHINESE_SIMPLIFIED",
        "type": "generic",
        "size": "0x00000368"
    },
```


Εικόνα 23: Chinese Language

Ένας ακόμα μηχανισμός λειτουργίας είναι ότι γίνεται επικοινωνία με εξωτερικό host προφανώς με τον σκοπό του ελέγχου από κάποιο Control Server σύστημα. Σημαντική είναι η πληροφορία ότι δεν γίνεται DNS query δηλαδή δεν αντιστοιχεί όνομα διεύθυνσης με τις IP που εντοπίσαμε.

```
{
    "families": [],
    "description": "Communicates with host for which no DNS query was performed"
    "severity": 3,
    "ttp": {},
    "markcount": 2,
    "references": [],
    "marks": [
        {
            "host": "20.199.120.85",
            "type": "generic"
        },
        {
            "host": "93.184.220.29",
            "type": "generic"
        }
    ],
    "name": "nolookup_communication"
```


Εικόνα 24: Host With No DNS Query

Εντοπίστηκαν ακόμα και κομμάτια κώδικα kernel modules χωρίς ονόματα που αποτελεί σίγουρα επικίνδυνο φαινόμενο.


```
1274            "description": "Kernel module without a name",
1275            "severity": 3,
1276            "ttp": {},
1277            "markcount": 2,
1278            "references": [],
1279            "marks": [
1280                {
1281                    "kernel_module": {
1282                        "kernel_module_offset": "0xf8000a4aee52",
1283                        "kernel_module_file": "",
1284                        "kernel_module_base": "0xccccccccccceceb",
1285                        "kernel_module_name": "",
1286                        "kernel_module_size": 4924454
1287                    },
1288                    "type": "volatility",
1289                    "plugin": "mysterious_kernel_module"
```


Εικόνα 25: Kernel Module Without a Name

Προχωράμε σε αποτελέσματα των επόμενων αναλύσεων του ίδιου ransomware και επικεντρωνόμαστε στις διαφορές και στα καινούρια ευρήματα που φανερώνουν.

Στην ανάλυση με hash 1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24 εντοπίστηκαν αποτελέσματα από την γραμμή εντολών όπου απέτυχε να εκτελεστεί εντολή με το vssadmin καθώς διατυπώνει πως δεν την αναγνωρίζει το σύστημα.


```
                {
                    "call": {
                        "category": "misc",
                        "status": 1,
                        "stacktrace": [],
                        "api": "WriteConsoleW",
                        "return_value": 1,
                        "arguments": {
                            "buffer": "'vssadmin' is not recognized as an internal or external
command,\r\noperable program or batch file.\r\n",
                            "console_handle": "0x000000a0"
                        },
```


Εικόνα 26: Vssadmin Fail

Επίσης μας δόθηκε η πληροφορία ότι εντοπίζει χώρο στην μνήμη πιθανώς για να αποκρυπτογραφηθεί κάποιο αρχείο.


```
{
    "families": [],
    "description": "Allocates read-write-execute memory (usually to unpack itself)",
    "severity": 2,
    "ttp": {},
    "markcount": 1,
    "references": [],
    "marks": [
        {
            "call": {
                "category": "process",
                "status": 1,
                "stacktrace": [],
                "api": "NtProtectVirtualMemory",
                "return_value": 0,
                "arguments": {
                    "process_identifier": 4684,
                    "stack_dep_bypass": 0,
                    "stack_pivoted": 0,
                    "heap_dep_bypass": 0,
                    "length": 4096,
                    "protection": 64,
                    "process_handle": "0xffffffff",
                    "base_address": "0x75c43000"
                },
                "time": 1644183785.460657,
                "tid": 3164,
                "flags": {
                    "protection": "PAGE_EXECUTE_READWRITE"
                }
            },
            "pid": 4684,
            "type": "call",
            "cid": 40
        }
    ],
    "name": "allocates_rwx"
},
```


Εικόνα 27: Allocated Memory

Γίνεται σύνδεση σε διεύθυνση IP η οποία δεν ανταποκρίνεται σε αιτήματα και με αυτόν το τρόπο καταλαβαίνουμε ότι είναι ύποπτη καθώς οι νόμιμες υπηρεσίες ανταποκρίνονται σε αιτήματα και παραμένουν ενεργές.


```
            "families": [],
            "description": "Connects to an IP address that is no longer responding to requests (legitimate
services will remain up-and-running usually)",
            "severity": 5,
            "ttp": {},
            "markcount": 1,
            "references": [],
            "marks": [
                {
                    "category": "dead_host",
```


Εικόνα 28: Dead Host

Επιπρόσθετα στα Json files μας δίνεται και η ενότητα ονόματος Generic που είναι υποενότητα του Behavior και δίνει γενικές πληροφορίες για την δραστηριότητα του δείγματος που αναλύουμε. Οι πληροφορίες είναι :

1. Κύρια εκτελέσιμα αρχεία που χρησιμοποιήθηκαν.
2. Εμφάνηση του process path.

3. Τι Registry Keys ανοίχτηκαν και διαβάστηκαν.
4. Τι βιβλιοθήκες φορτώθηκαν.
5. Για ποιούς φακέλους ζητήθηκαν πληροφορίες
6. Το pid και ppid των διεργασιών

Ενδεικτικά παραθέτονται εικόνες του αντίστοιχου πεδίου από αναλύσεις του Conti.


```
"process_path": "C:\\Users\\Z\\AppData\\Local\\Temp\\1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24.exe
"process_name": "1ef1ff8b1e81815d13bdd293554ddf8b3e57490dd3ef4add7c2837ddc67f9c24.exe",
"pid": 4684,
"summary": {
    "dll_loaded": [
        "Iphlpapi.dll",
        "kernel32.dll",
        "Kernel32.dll",
        "Netapi32.dll",
        "Rstrtmgr.dll",
        "Advapi32.dll",
        "ntdll.dll",
        "Shlwapi.dll",
        "Ws2_32.dll",
        "User32.dll"
    ],
    "file_opened": [
        "C:\\Windows\\System32\\oleaut32.dll",
        "C:\\Windows\\System32\\netapi32.dll",
        "C:\\Windows\\Globalization\\Sorting\\sortdefault.nls",
        "C:\\Windows\\System32\\IPHLPAPI.DLL",
        "C:\\Windows\\System32\\ncrypt.dll",
        "C:\\Windows\\System32\\ws2_32.dll",
        "C:\\Windows\\System32\\netutils.dll",
        "C:\\Windows\\System32\\RstrtMgr.dll",
        "C:\\Windows\\System32\\ole32.dll",
        "C:\\Windows\\System32\\ntasn1.dll"
    ],
    "regkey_opened": [
        "HKEY_LOCAL_MACHINE\\System\\CurrentControlSet\\Control",
        "HKEY_LOCAL_MACHINE\\Software\\Microsoft\\RestartManager"
    ],
    "file_exists": [
        "C:\\Windows\\System32\\cmd.exe"
    ],
    "mutex": [
        "Local\\SM0:4684:168:WilStaging_02",
        "_CONTI_"
    ],
    "regkey_read": [
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\Policies\\Microsoft\\Windows\\safer\\codeidentifiers\\AuthenticodeEnabled",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\CustomLocale\\en-US",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\WaitToKillServiceTimeout",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\Sorting\\Versions\\000602xx",
        "HKEY_LOCAL_MACHINE\\SOFTWARE\\Policies\\Microsoft\\Windows\\safer\\codeidentifiers\\TransparentEnabled",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\ExtendedLocale\\en-US",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\Sorting\\Ids\\en",
        "HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders\\Cache",
        "HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\Sorting\\Ids\\en-US"
    ]
},
"first_seen": 1644183776.648657,
"ppid": 3872
```


Εικόνα 29: General Behavior


```
{
    "parent_id": 3872,
    "process_id": 4684,
    "create_time": "2022-02-06 19:42:48 UTC+0000",
    "num_threads": "1",
    "process_name": "1ef1ff8b1e8181",
    "num_handles": "0",
    "session_id": "1",
    "exit_time": ""
},
```


Εικόνα 30: PPID

Εδώ βλέπουμε ότι έχει το κύριο εκτελέσιμο έχει ώς PPID το 3872 που είναι το εκτελέσιμο inject-x86.exe. Αυτή η διεργασία εμφανίζεται σε αρκετά δείγματα και φαίνεται ότι είναι κακόβουλη.

Για το inject-x86.exe δυστηχώς δεν δίνονται πολλές πληροφορίες παρα μόνο ότι παραποιεί μερικά δικαιώματα του συστήματος όπως το SeLoadDriverPrivilege και το SeDebugPrivilege [12]. Η χρήση τους είναι :

- Το SeLoadDriverPrivilege χρησιμοποιείται για τη φόρτωση ή την εκφόρτωση ενός προγράμματος οδήγησης συσκευής.
- Το δικαίωμα SeDebugPrivilege απαιτείται για εντοπισμό σφαλμάτων και προσαρμογή της μνήμης μιας διαδικασίας που ανήκει σε άλλο λογαριασμό.

Παρακάτω παρατίθενται συνοπτικά οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox σε όλα τα δείγματα του Conti.

1) This executable has a PDB path
2) Command line console output was observed
3) Allocates read-write-execute memory (usually to unpack itself)
4) Communicates with host for which no DNS query was performed
5) Performs some HTTP requests
6) Foreign language identified in PE resource
7) Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)
8) Found URLs related to Tor in process memory dump (e.g. onion services, Tor2Web, and Ransomware)
9) Writes a potential ransom message to disk
10) File has been identified by 55 AntiVirus engines on VirusTotal as malicious

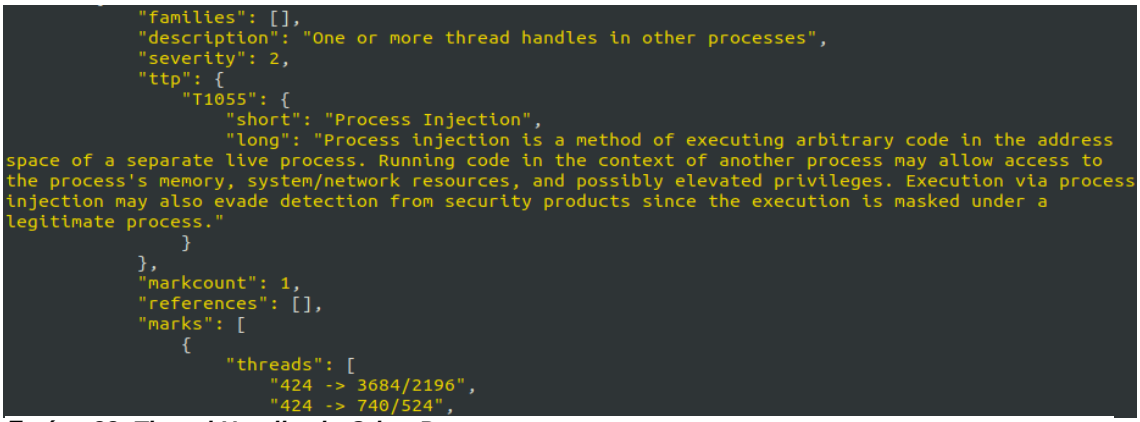
2.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Ένας ακόμα μηχανισμός λειτουργίας του εξεταζόμενου δείγματος είναι ότι με βάση την υψηλή εντροπία που εμφανίζεται σε μέρη των αρχείων του, καταλαβαίνουμε ότι χρησιμοποιείται κάποια συμπίεση-κρυπτογράφηση αρχείων για την αποφυγή ανίχνευσης [13].


```
],
"marks": [
    {
        "entropy": 7.161468098656868,
        "section": {
            "size_of_data": "0x0007c000",
            "virtual_address": "0x00040000",
            "entropy": 7.161468098656868,
            "name": ".rsrc",
            "virtual_size": "0x0007b3e8"
        },
        "type": "generic",
        "description": "A section with a high entropy has been found"
    },
    {
        "entropy": 0.6739130434782609,
        "type": "generic",
        "description": "Overall entropy of this PE file is high"
    }
],
"name": "packer_entropy"
```

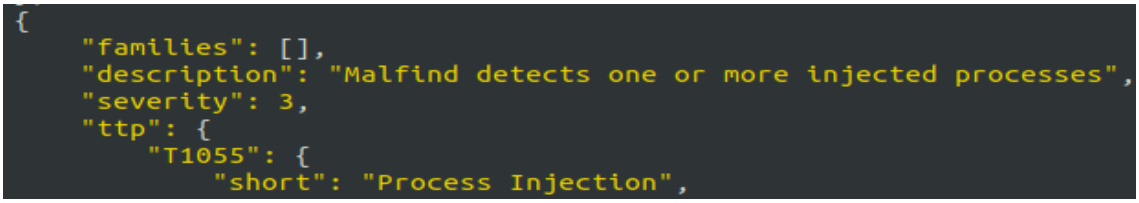

Εικόνα 31: Entropy

Ακολούθως από την λειτουργία του Volatility στην ανάλυση βλέπουμε ότι γίνεται χρήση της τεχνικής process injection [14] καθώς γίνεται εκτέλεση κώδικα ενσωματωμένη μέσα σε νόμιμες διεργασίες ώστε να μην γίνει αντιληπτή η κακόβουλη δραστηριότητα από προϊόντα ασφαλείας. Η εκτέλεση κώδικα στο πλαίσιο μιας άλλης διεργασίας μπορεί να επιτρέψει την πρόσβαση στη μνήμη της διεργασίας, στους πόρους συστήματος/δικτύου και, ενδεχομένως, σε αυξημένα προνόμια.



Εικόνα 32: Thread Handles in Other Processes

Η χρήση της τεχνικής process injection επαληθεύεται και από την πρόσθετη λειτουργία του Volatility Malfind.

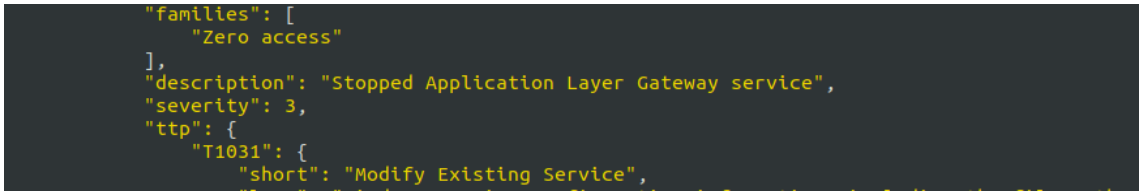


Εικόνα 33: Malfind Detection

Συνεχίζοντας βλέπομε ότι το Process Environment Block (PEB) έχει αλλαχτεί για να κρύψει τα φορτωμένα modules. Αρκετά DLL είναι πολύ πιθανό να μην έχουν φορτωθεί από το LoadLibrary. Πρέπει να σημειωθεί ότι το PEB είναι μια δομή δεδομένων των Windows και χρησιμοποιείται κυρίως από το λειτουργικό σύστημα εσωτερικά. Περιγράφουν πλήρως την δομή των διεργασιών όπως παραμέτρων εκκίνησης, αντικειμένων συγχρονισμού και δομών δεδομένων φόρτωσης εικόνων. Έτσι η αλλαγή στην δομή του PEB χωρίς την χρήση του load library είναι εξαιρετικά ύποπτη δραστηριότητα.

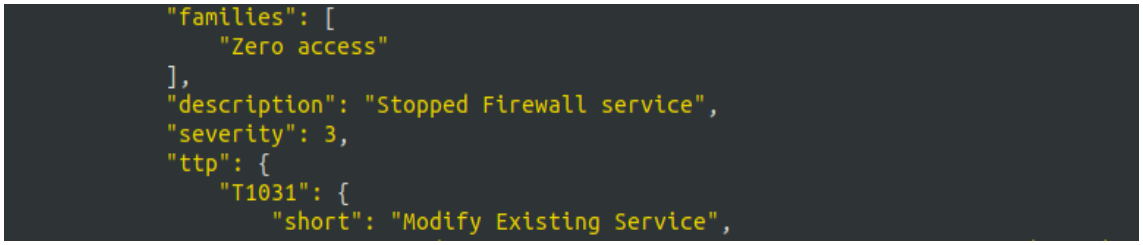
Εικόνα 34: PEB Modified

Έπειτα σταματάει η διεργασία του Application Layer Gateway όπου στα Microsoft Windows παρέχει υποστήριξη για προσθήκες τρίτων που επιτρέπουν στα πρωτόκολλα δικτύου να περνούν μέσα από το Τείχος προστασίας των Windows και να λειτουργούν πίσω από αυτό και την Κοινή χρήση σύνδεσης στο Internet.



Εικόνα 35: Stopped Application Layer Gateway Service

Στην συνέχεια σταματά η υπηρεσία του Firewall για την αποφυγή αναγνώρισης ύποπτης δικτυακής δραστηριότητας καθώς και πολλές άλλες διεργασίες του συστήματος.

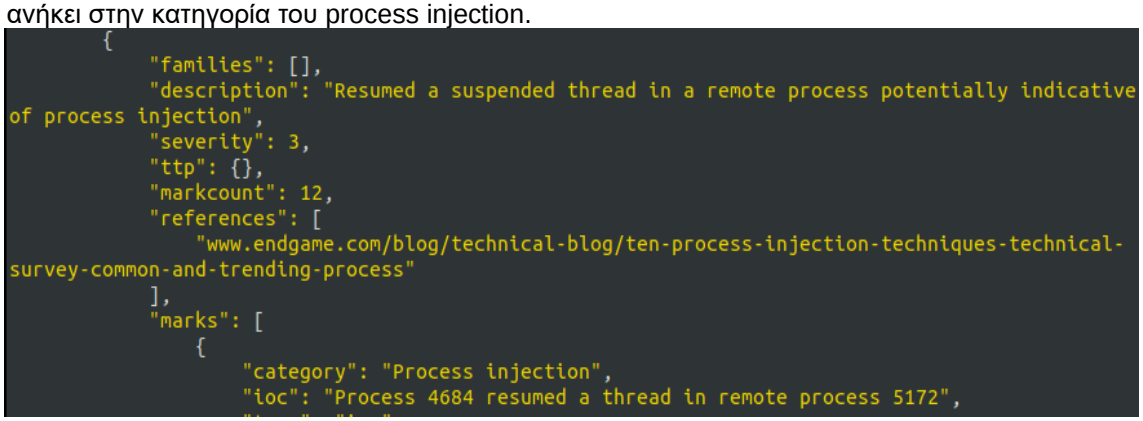


```
            "families": [
                "Zero access"
            ],
            "description": "Stopped Firewall service",
            "severity": 3,
            "ttp": {
                "T1031": {
                    "short": "Modify Existing Service",
```


Εικόνα 36: Stopped Firewall Service

Από αυτό το σημείο αναφέρουμε διαφορές των υπόλοιπων αναλύσεων σε σχέση με την πρώτη ανάλυση.

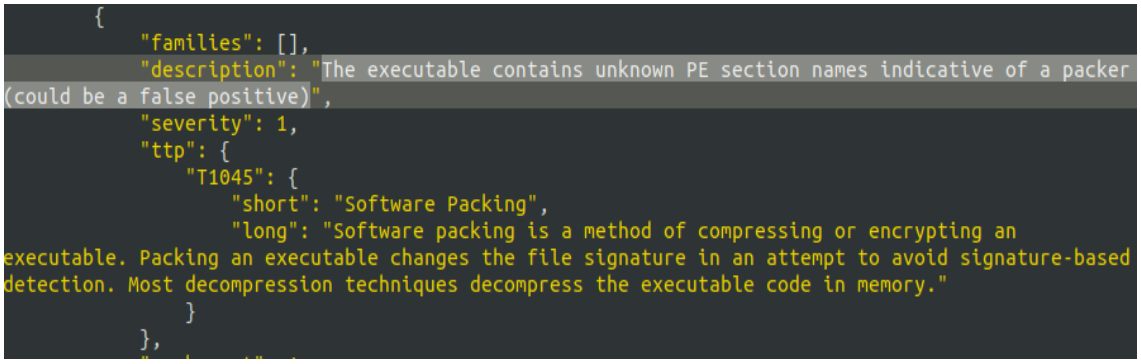
Σε επόμενες αναλύσεις εντοπίστηκε ότι έγινε επανεκκίνηση σε threads ορισμένων διεργασιών. Αυτή η τεχνική χρησιμοποιείται για την αποφυγή εντοπισμού από το σύστημα και ανήκει στην κατηγορία του process injection.



```
        {
            "families": [],
            "description": "Resumed a suspended thread in a remote process potentially indicative
of process injection",
            "severity": 3,
            "ttp": {},
            "markcount": 12,
            "references": [
                "www.endgame.com/blog/technical-blog/ten-process-injection-techniques-technical-
survey-common-and-trending-process"
            ],
            "marks": [
                {
                    "category": "Process injection",
                    "ioc": "Process 4684 resumed a thread in remote process 5172",
```


Εικόνα 37: Resumed Suspended Thread

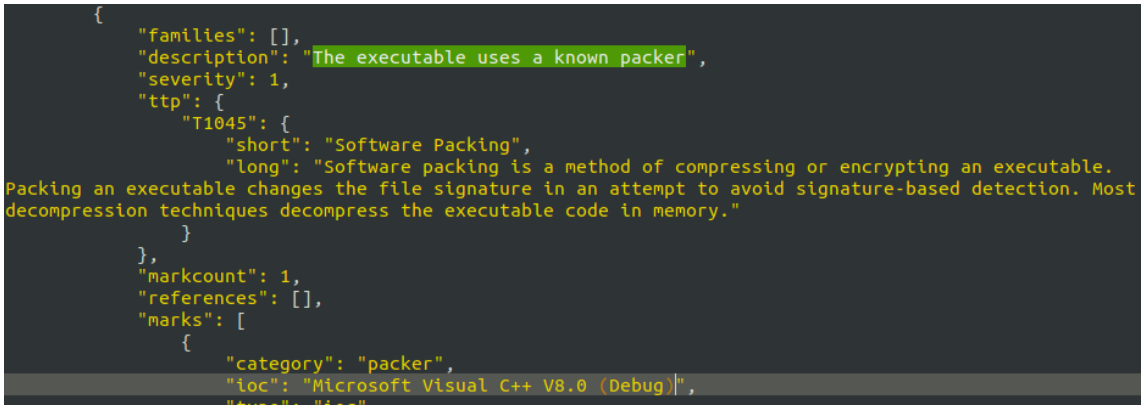
Τα αποτελέσματα μας δίνουν πληροφορίες ότι υπάρχουν άγνωστα ονόματα PE που είναι ενδείξεις ενός packer.



```
        {
            "families": [],
            "description": "The executable contains unknown PE section names indicative of a packer
(could be a false positive)",
            "severity": 1,
            "ttp": {
                "T1045": {
                    "short": "Software Packing",
                    "long": "Software packing is a method of compressing or encrypting an
executable. Packing an executable changes the file signature in an attempt to avoid signature-based
detection. Most decompression techniques decompress the executable code in memory."
                }
            },
```


Εικόνα 38: Unknown PE Section

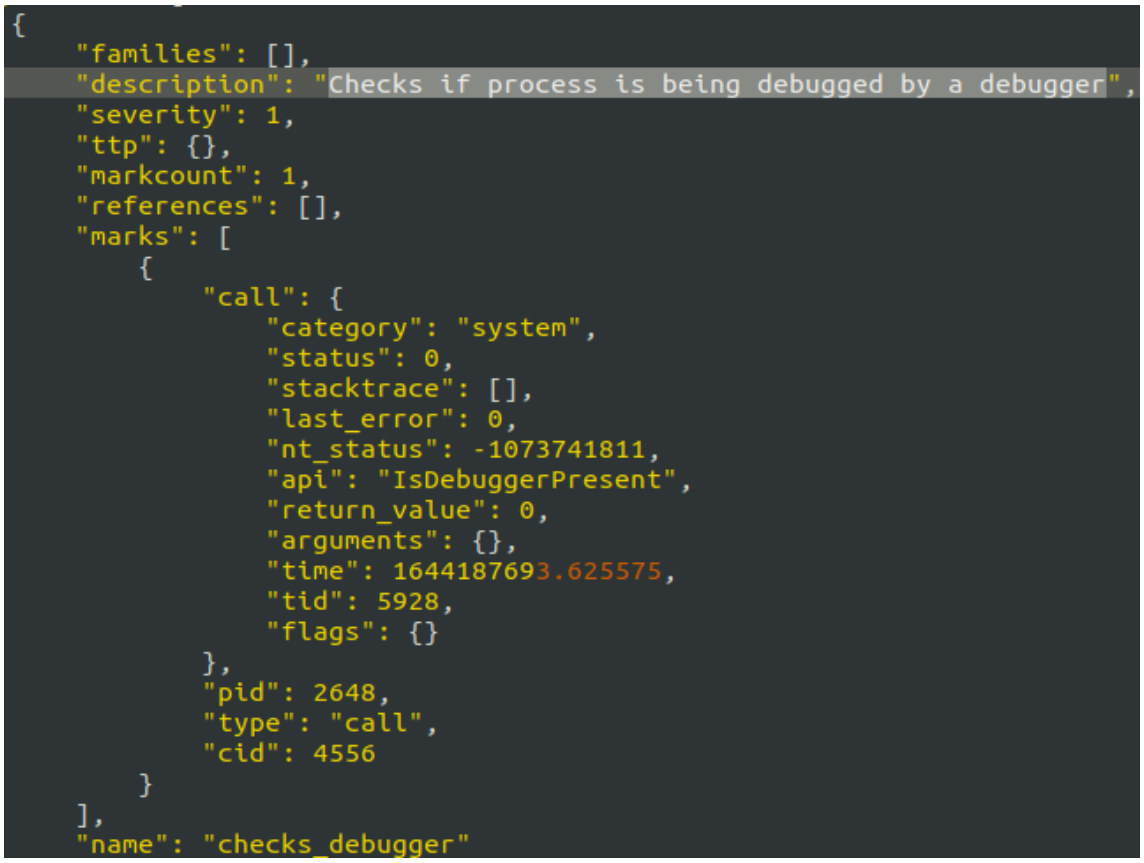
Ανακαλύψαμε επίσης το όνομα του συγκεκριμένου packer που είναι το Microsoft Visual C++ V8.0 (Debug).



```
        {
            "families": [],
            "description": "The executable uses a known packer",
            "severity": 1,
            "ttp": {
                "T1045": {
                    "short": "Software Packing",
                    "long": "Software packing is a method of compressing or encrypting an executable.
Packing an executable changes the file signature in an attempt to avoid signature-based detection. Most
decompression techniques decompress the executable code in memory."
                }
            },
            "markcount": 1,
            "references": [],
            "marks": [
                {
                    "category": "packer",
                    "ioc": "Microsoft Visual C++ V8.0 (Debug)",
```


Εικόνα 39: Known Packer

Σε επόμενη ανάλυση εντοπίσαμε ότι γίνεται έλεγχος αν γίνεται χρήση debugger που μπορεί να αναλύσει το ίο-μορφικό λογισμικό. Αν εντοπιστεί debugger τότε θα σταματήσει την λειτουργία του το συγκεκριμένο ransomware και δεν θα φανερώσει τις ικανότητές του [15]. Όπως βλέπουμε στο return value η τιμή είναι μηδέν, και αυτό σημαίνει ότι το ransomware δεν εντόπισε την χρήση κάποιου debugger. Εδώ μας παρουσιάζεται ένα διαφορετικό είδος αποφυγής της ανίχνευσης καθώς εδώ το ransomware προσπαθεί να αποφύγει την ανάλυσή του και όχι μόνο να μην γίνει αντιληπτό από το σύστημα.



```
{
    "families": [],
    "description": "Checks if process is being debugged by a debugger",
    "severity": 1,
    "ttp": {},
    "markcount": 1,
    "references": [],
    "marks": [
        {
            "call": {
                "category": "system",
                "status": 0,
                "stacktrace": [],
                "last_error": 0,
                "nt_status": -1073741811,
                "api": "IsDebuggerPresent",
                "return_value": 0,
                "arguments": {},
                "time": 1644187693.625575,
                "tid": 5928,
                "flags": {}
            },
            "pid": 2648,
            "type": "call",
            "cid": 4556
        }
    ],
    "name": "checks_debugger"
```


Εικόνα 40: Debugged by a debugger

Στην ίδια ανάλυση ανακαλύψαμε και άλλον μηχανισμό ασφαλείας του Conti ransomware για να αποφύγει την ανάλυση μέσα σε sandbox. Παρακάτω βλέπουμε πως το εκτελέσιμο προς

ανάλυση έμεινε ανενεργό για 706 δευτερόλεπτα ώστε να καθυστερήσει την ανάλυση με απώτερο σκοπό να μην αναλυθεί.


```
        {
            "families": [],
            "description": "A process attempted to delay the analysis task.",
            "severity": 2,
            "ttp": {},
            "markcount": 1,
            "references": [],
            "marks": [
                {
                    "type": "generic",
                    "description":
"4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9.exe tried to sleep 706 seconds,
actually delayed analysis time by 706 seconds"
                }
            ],
            "name": "antisandbox_sleep"
        },
```


Εικόνα 41: Delay Analysis

Συνεχίζοντας μια ακόμα τεχνική αντί-ανάλυσης είναι η αναζήτηση διεργασιών που σχετίζονται με γνωστά sandbox περιβάλλοντα που γίνεται προσπάθεια να ανακαλυφθούν.


```
        {
            "families": [],
            "description": "Searches running processes potentially to identify processes for sandbox evasion,
code injection or memory dumping",
            "severity": 2,
            "ttp": {
                "T1057": {
                    "short": "Process Discovery",
                    "long": "Adversaries may attempt to get information about running processes on a system.
Information obtained could be used to gain an understanding of common software running on systems within the
network."
                }
            },
```


Εικόνα 42: Searches Running Processes

Σε επόμενες αναλύσεις εντοπίσαμε διαφορετική χρήση packer για την συμπίεση η κρυπτογράφηση των αρχείων τον Armadillo v1.71.


```
        {
            "families": [],
            "description": "The executable uses a known packer",
            "severity": 1,
            "ttp": {
                "T1045": {
                    "short": "Software Packing",
                    "long": "Software packing is a method of compressing or encrypting an executable.
Packing an executable changes the file signature in an attempt to avoid signature-based detection. Most
decompression techniques decompress the executable code in memory."
                }
            },
            "markcount": 1,
            "references": [],
            "marks": [
                {
                    "category": "packer",
                    "ioc": "Armadillo v1.71",
                    "type": "ioc",
                    "description": null
                }
            ],
            "name": "peid_packer"
        },
```


Εικόνα 43: Known Packer

Τα Evasion Techniques που εντοπίστηκαν συνοπτικά σε όλα τα δείγματα με βάση τα Signatures [16] του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping
3) The executable uses a known packer
4) The file contains an unknown PE resource name possibly indicative of a packer
5) The executable contains unknown PE section names indicative of a packer (could be a false positive)
6) A process attempted to delay the analysis task
7) One or more thread handles in other processes
8) One or more processes crashed
9) Potentially malicious URLs were found in the process memory dump
10) Stopped Application Layer Gateway service
11) Kernel module without a name
12) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
13) Malfind detects one or more injected processes
14) Stopped Firewall service
15) Checks if process is being debugged by a debugger
16) Resumed a suspended thread in a remote process potentially indicative of process injection

2.4 Κρυπτογράφηση

Όπως βλέπουμε η βιβλιοθήκη cryptsp.dll (Cryptographic Service Provider) των Microsoft Windows χρησιμοποιείται για την επιλογή εργαλείων και ρυθμίσεων που θα χρησιμοποιηθούν στην κρυπτογράφηση. Χρησιμοποιείται από διεργασίες όπως πολλαπλές svchost.exe, spoolsv.exe, explorer.exe για την κάλυψη της χρήσης του. Επίσης χρησιμοποιείται από όλες της αναλύσεις του Conti που εκτελέσαμε.


```
{
    "dll_in_init": true,
    "init_full_dll_name": "C:\\Windows\\System32\\cryptsp.dll",
    "process_id": 1360,
    "dll_in_mem": true,
    "mem_full_dll_name": "C:\\Windows\\System32\\cryptsp.dll",
    "dll_mapped_path": "\\Windows\\System32\\cryptsp.dll",
    "process_name": "svchost.exe",
    "dll_base": "0x7ffaf2210000",
    "dll_in_load": true,
    "load_full_dll_name": "C:\\Windows\\System32\\cryptsp.dll"
},
```


Εικόνα 44: Cryptsp.dll

Επίσης φαίνεται πως γίνεται επεξεργασία του πρωτοκόλλου SSL μέσω πρόσβασης που υπήρξε στο registry key του.


```
{
    "handle_granted_access": "131097",
    "process_id": 612,
    "handle_value": "3168",
    "handle_type": "Key",
    "handle_name": "MACHINE\\SYSTEM\\CONTROLSET001\\CONTROL\\CRYPTOGRAPHY\\CONFIGURATION\\LOCAL\\SSL\\00010002"
},
```


Εικόνα 45: SSL

Χρησιμοποιείται επίσης η διεργασία Cryptsvc των Windows που παρέχει τέσσερις υπηρεσίες διαχείρισης:

1. Database service, η οποία επιβεβαιώνει τις υπογραφές των αρχείων των Windows και επιτρέπει την εγκατάσταση νέων προγραμμάτων.
2. Protected Root Service, η οποία προσθέτει και αφαιρεί πιστοποιητικά Trusted Root Certification Authority από αυτόν τον υπολογιστή.
3. Automatic Root Certificate Update Service, η οποία ανακτά πιστοποιητικά root από το Windows Update και ενεργοποιεί το SSL.
4. Key Service, η οποία βοηθά στην εγγραφή αυτού του υπολογιστή για πιστοποιητικά.


```
{
    "service_display_name": "Cryptographic Services",
    "service_binary_path": "C:\\Windows\\system32\\svchost.exe -k NetworkService -p",
    "process_id": 2600,
    "service_name": "CryptSvc",
    "service_type": "SERVICE_WIN32_OWN_PROCESS, SERVICE_WIN32_SHARE_PROCESS",
    "service_order": 89,
    "service_offset": "0x1dc27c359a0",
    "service_state": "SERVICE_RUNNING"
},
```


Εικόνα 46: Cryptsvc

Στις αναλύσεις 4f17d7fa344b970890ed1bc52a0da95146cab9fe56ecabafafacb0ad212558c9, 24ac73821de77cc9644d2ac40e97067ff63f625b5f20e085ad10535e47d7db5 και 6815e1e06e29863290319e b3e814ae2a394271aa2f95cc7c31a649c4c2f4fd04 βλέπουμε πως καλείται το API CryptAcqireContextA για την RSA και AES αλγορίθμων κρυπτογράφησης.


```
{
    "category": "crypto",
    "status": 1,
    "stacktrace": [],
    "api": "CryptAcquireContextA",
    "return_value": 1,
    "arguments": {
        "crypto_handle": "0x00b8ad58",
        "container": "",
        "flags": 4026531840,
        "provider": "Microsoft Enhanced RSA and AES Cryptographic Provider",
        "provider_type": 24
    },
    "time": 1644189286.770062,
    "tid": 5996,
    "flags": {}
},
```


Εικόνα 47: CryptAcqireContextA

Στις ίδιες αναλύσεις εντοπίσαμε και μια επιπρόσθετη δραστηριότητα που ψάχνει και εντοπίζει αν στο σύστημα υπάρχει εγκατεστημένο το πρόγραμμα Winrar. Μέσω του FindFirstFileExW API πραγματοποιεί αναζήτηση με όνομα και χαρακτηριστικά που ταιριάζουν με αυτά που έχουν καθοριστεί. Μέσω του GetFileAttributesW βλέπει τα δικαιώματά του, μετά μέσω του NtCreateFile δημιουργεί φάκελο και συνεχίζει με το NtWriteFile που γράφει σε αρχείο δεδομένα δηλαδή κρυπτογραφεί.


```
    "api": "FindFirstFileExW",
    "return_value": 12386680,
    "arguments": {
        "filepath_r": "C:\\Program Files\\WinRAR\\*",
        "filepath": "C:\\Program Files\\WinRAR\\*"
```


Εικόνα 48: Winrar

Σε παραπάνω εικόνα είδαμε ότι χρησιμοποιείται ο αλγόριθμος AES. Από διάφορα επιστημονικά άρθρα όμως εντοπίσαμε πως ο αλγόριθμος κρυπτογράφησης που χρησιμοποιείται στα τελευταία δείγματα του Conti είναι ο CHACHA8 [17] ή ο CHACHA20 [18] ανάλογα το δείγμα

με την χρήση ταυτόχρονα 32 CPU threads ώστε να πετυχαίνει να είναι από τα ταχύτερα ransomware στον τρόπο κρυπτογράφησης. Ο CHACHA είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης, όμως είδαμε πως γίνεται και χρήση της ασύμμετρης κρυπτογράφησης RSA. Αυτό σημαίνει ότι τα αρχεία κρυπτογραφούνται με συνδυασμό των δύο μεθόδων συμμετρικής και ασύμμετρης κρυπτογράφησης ώστε να αποκτήσουν τα πλεονεκτήματα και των δύο μεθόδων. Σε επόμενες αναλύσεις θα εξηγήσουμε διεξοδικά πως λειτουργεί αυτού του είδους η υβριδική κρυπτογράφηση.

Παρακάτω παρατίθεται μήνυμα λύτρων προς τον χρήστη όταν ολοκληρώνεται η διαδικασία παραβίασης του συστήματός μας από το Conti ransomware, και τα αρχεία μας έχουν κλειδωθεί στον υπολογιστή μας ή και σε περισσότερους που βρίσκονται στο ίδιο δίκτυο. Σκοπός του μηνύματος είναι η ενημέρωση για το τρόπο που πρέπει το θύμα να πληρώσει καθώς πρέπει να διασφαλιστεί η ανωνυμία τους. Παροτρύνουν λοιπόν να κατεβάσουμε τον περιηγητής ιστού TOR γνωστός για την διατήρηση της ανωνυμίας των χρηστών του, ώστε να μπορέσουμε να μπούμε στον ιστοσελίδα τους και να μας δώσουν οδηγίες για την πληρωμή των λύτρων. Η πληρωμή γίνεται δεκτή μόνο στο κρυπτονόμισμα της επιλογής του θύτη καθώς αυτό είναι ένα μέτρο κατά του εντοπισμού του.


```
                        "filepath": "C:\\readme.txt",
                        "buffer": "All of your files are currently encrypted by
CONTI strain. \r\n\r\nAs you know (if you don't - just \"google it\"), all of the data
that has been encrypted by our software cannot be recovered by any means without
contacting our team directly. \r\nIf you try to use any additional recovery software -
the files might be damaged, so if you are willing to try - try it on the data of the
lowest value.\r\n\r\nTo make sure that we REALLY CAN get your data back - we offer you
to decrypt 2 random files completely free of charge.\r\n\r\nYou can contact our team
directly for further instructions through our website :\r\n\r\nTOR VERSION :\r\n(you
should download and install TOR browser first https://torproject.org)\r\n\r\nhttp://
contirecj4hbzmyzuydyzrvm2c65blmvhoj2cvf25zqj2dwrrqcq5oad.onion/\r\n\r\nHTTPS VERSION :
\r\nhttps://contirecovery.xyz/\r\n\r\nYOU SHOULD BE AWARE!\r\nJust in case, if you try
to ignore us. We've downloaded a pack of your internal data and are ready to publish it
on out news website if you do not respond. So it will be better for both sides if you
contact us as soon as possible.\r\n\r\n\r\n---BEGIN ID---
\r\n4C5vgfrLN5lYVhVYOrHK7ENa9GytbQOmpNgUIjyeAr8lROT5z1cscxMBiZcgyIZD\r\n---END ID---",
                        "offset": 0
```


Εικόνα 49: Conti Ransom Message

Σε επόμενες αναλύσεις θα επικεντρωθούμε σε νέα ευρήματα και διαφορές που έχουν τα επόμενα Ransomware σε σχέση με το Conti καθώς αρκετές λειτουργίες επαναλαμβάνονται.

3 Netwalker(Mailto)

3.1 Living Off the Land Binaries, Scripts and Libraries

Στις περισσότερες αναλύσεις του Netwalker με hash sha256 :

1) 4f7bdda79e389d6660fca8e2a90a175307a7f615fa7673b10ee820d9300b5c60
2) 5d869c0e077596bf0834f08dce062af1477bf09c8f6aa0a45d6a080478e45512
3) 6c39c5f5d143700d4ad43b0aa7fb6a51e77817060467cf3462ef037176e1f50f
4) 44b5d24e5e8fd8e8ee7141f970f76a13c89dd26c44b336dc9d6b61fda3abf335
5) 46dbb7709411b1429233e0d8d33a02cccd54005a2b4015dcfa8a890252177df9
6) 57cf4470348e3b5da0fa3152be84a81a5e2ce5d794976387be290f528fa419fd
7) 62df5824be844b16d99bf4a59e6bf51bcaa7a2c2f4852d983bc33eaf60a9c5ca
8) 27319e75c23693399977e92b9a7ba5680a7a9db448f93b3221840c61301604d5
9) ac0882d87027ac22fc79cfe2d55d9a9d097d0f8eb425cf182de1b872080930ec
10) ce399a2d07c0851164bd8cc9e940b84b88c43ef564846ca654df4abf36c278e6
11) d5c106719e9c8878795899bede78505796659b1b347fe9374d8b2061fcc6a84c
12) d5c106719e9c8878795899bede78505796659b1b347fe9374d8b2061fcc6a84c
13) e7dfaf7ac518e174e42e59d45c179d083ad53eead79ed4420e311441e1af79d4
14) e01691e3b7d9d1c6de7e0ef902bf609543cdf084e600fd0a3833deaa501464ee
15) ee531cd7011cb5c2625d40892b70cf7e3860dbb92648391068e1f340e5d6c47f
16) f2215e1a848bc5a5d172745201ea428b1d16fee7c814c5c5180a94a134592e86
17) fd3489f6067ef7ca3999776205839424cb7349134baaeb693abcecaa2c5bf913

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις με hash sha256 :

1) 1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540
2) e167717c47bc2776217b5cd16fe7ad8a7228d626b3a452b2c8664f9c1c057b66
3) f93209fccd0c452b8b5dc9db46341281344156bbedd23a47d2d551f80f460534

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Rundll32.exe				

Λειτουργικότητα και Ροή Διεργασιών

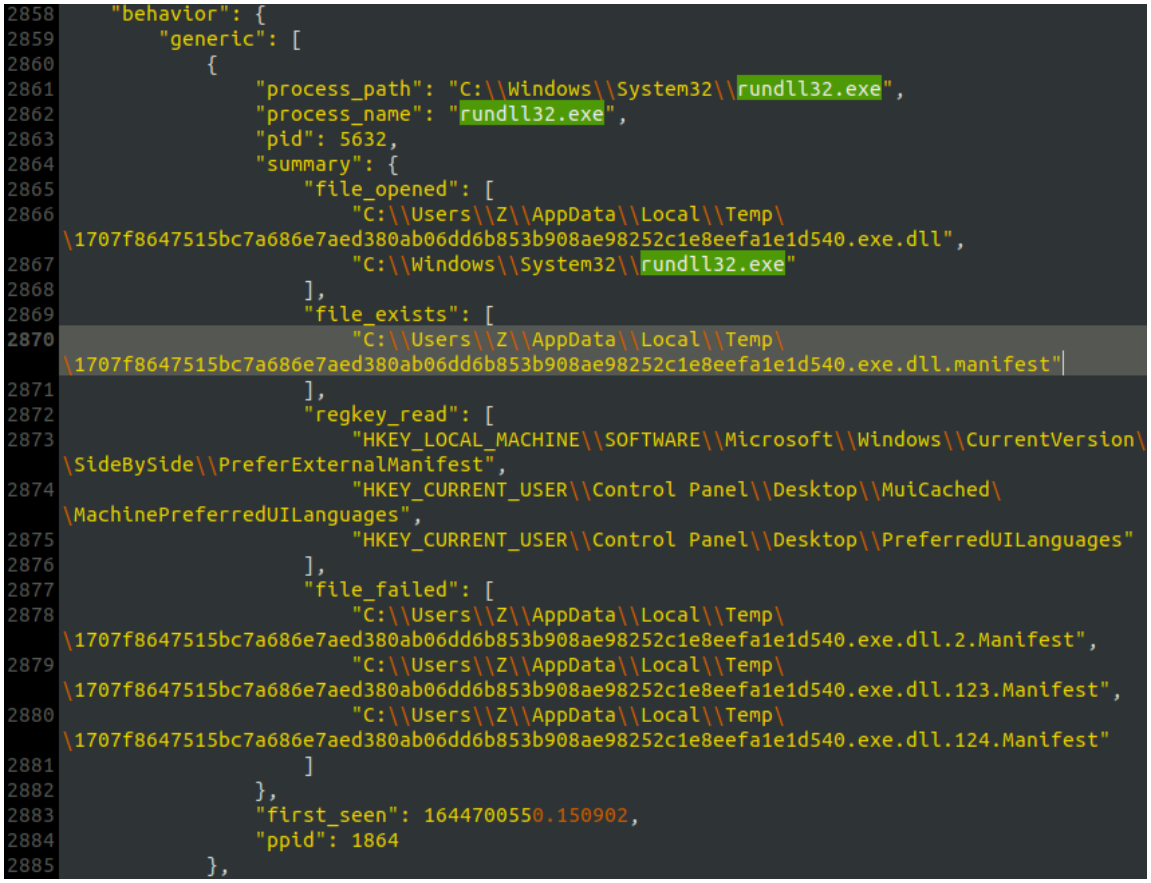
Η διεργασία explorer.exe καλείται με τον ίδιο τρόπο όπως και στο Conti παραπάνω σε όλα τα δείγματά μας. Η μόνη διαφορά όταν φτάνουμε στο κύριο εκτελέσιμο είναι ότι δεν καλούνται διεργασίες cmd και αντί αυτού γίνεται χρήση DLL αρχείου. Οι διεργασίες Dllhost.exe, Shell32.dll, Setupapi.dll, Shdocvw.dll εντοπίζονται στις αναλύσεις χωρίς να δείχνουν κάποια κακόβουλη δραστηριότητα οπότε προχωράμε στο Rundll32.exe.

Στις αναλύσεις που δραστηριοποιείται η διεργασία Rundll32.exe εντοπίσαμε πως η χρήση της είναι πανομοιότυπη σε όλα τα δείγματα. Υπάρχουν δυο διεργασίες με το ίδιο όνομα αλλά σαφώς διαφορετικό PID. Τα process path είναι τα παρακάτω :

- C:\\Windows\\System32\\rundll32.exe
- C:\\Windows\\SysWOW64\\rundll32.exe

Όπως παρατηρήθηκε η rundll32.exe που εμπεριέχεται στο System32 καλείται από την rundll32.exe που εμπεριέχεται στο SysWOW64 και το βλέπουμε από το PPID. Η

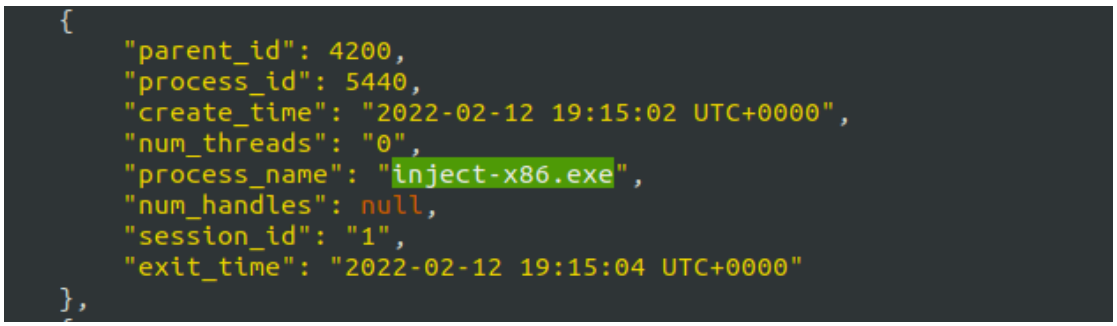
rundll32.exe(1864) που περιέχεται στο SysWOW64 καλείται από την διεργασία inject-x86.exe(5440).



```
    "behavior": {
        "generic": [
            {
                "process_path": "C:\\Windows\\System32\\rundll32.exe",
                "process_name": "rundll32.exe",
                "pid": 5632,
                "summary": {
                    "file_opened": [
                        "C:\\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll",
                        "C:\\Windows\\System32\\rundll32.exe"
                    ],
                    "file_exists": [
                        "C:\\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll.manifest"
                    ],
                    "regkey_read": [
                        "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\
\SideBySide\\PreferExternalManifest",
                        "HKEY_CURRENT_USER\\Control Panel\\Desktop\\MuiCached\
\MachinePreferredUILanguages",
                        "HKEY_CURRENT_USER\\Control Panel\\Desktop\\PreferredUILanguages"
                    ],
                    "file_failed": [
                        "C:\\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll.2.Manifest",
                        "C:\\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll.123.Manifest",
                        "C:\\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll.124.Manifest"
                    ]
                },
                "first_seen": 1644700550.150902,
                "ppid": 1864
            },
```


Εικόνα 50: Rundll32.exe

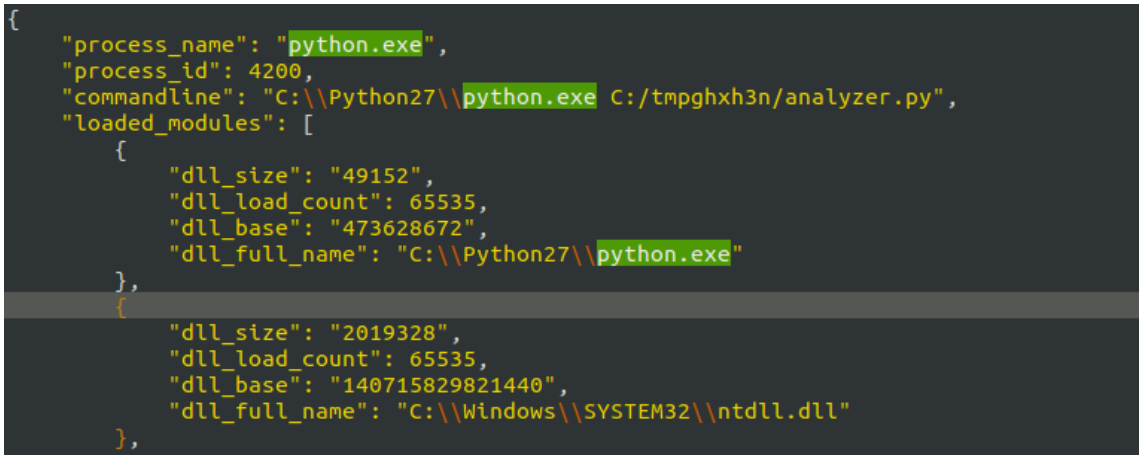
Συνεχίζοντας πιο βαθιά την προέλευση της inject-x86.exe βλέπουμε ότι από το pslist(process list) του Volatility έχει ως PPID το 4200 που είναι η διεργασία python.exe.



```
    {
        "parent_id": 4200,
        "process_id": 5440,
        "create_time": "2022-02-12 19:15:02 UTC+0000",
        "num_threads": "0",
        "process_name": "inject-x86.exe",
        "num_handles": null,
        "session_id": "1",
        "exit_time": "2022-02-12 19:15:04 UTC+0000"
    },
```


Εικόνα 51: PPID of Inject-x86.exe

Έπειτα βλέπουμε και το αρχείο που καλείται ονόματος analyzer.py χρησιμοποιώντας την γλώσσα python.



```
{
    "process_name": "python.exe",
    "process_id": 4200,
    "commandline": "C:\\Python27\\python.exe C:/tmpghxh3n/analyzer.py",
    "loaded_modules": [
        {
            "dll_size": "49152",
            "dll_load_count": 65535,
            "dll_base": "473628672",
            "dll_full_name": "C:\\Python27\\python.exe"
        },
        {
            "dll_size": "2019328",
            "dll_load_count": 65535,
            "dll_base": "140715829821440",
            "dll_full_name": "C:\\Windows\\SYSTEM32\\ntdll.dll"
        },
```

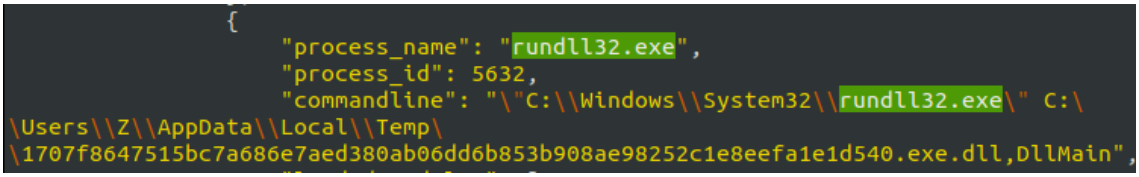

Εικόνα 52: Analyzer.py

Θα πρέπει να προσέξουμε τα process id καθώς στο JSON αρχείο μας θα βρούμε και την python.exe που εκτέλεσε ένα αρχείο ονόματος agent.py όπου το συγκεκριμένο είναι μέρος του Cuckoo Sandbox και χρησιμοποιείται για την ανταλλαγή πληροφοριών του Cuckoo Guest VM με τον Host μας (Ubuntu).

Καταλήγοντας οι διεργασίες καλούνται με την παρακάτω σειρά :

Smss.exe(488) → winlogon.exe(588) → userinit.exe(2664) → explorer.exe(2788) → python.exe(4240) → python.exe(4200) → inject-x86.exe(5440) → rundll32.exe(1864) → rundll32.exe(5632)

Στην συνέχεια εξετάζοντας την χρήση του rundll32.exe βλέπουμε ότι χρησιμοποιείται για να καλέσει ένα Dll αρχείο που από το όνομα εύκολα διακρίνουμε ότι είναι παραγωγής του κακόβουλου δείγματος.



```
        {
            "process_name": "rundll32.exe",
            "process_id": 5632,
            "commandline": "\"C:\\Windows\\System32\\rundll32.exe\" C:\
\Users\\Z\\AppData\\Local\\Temp\
\1707f8647515bc7a686e7aed380ab06dd6b853b908ae98252c1e8eefa1e1d540.exe.dll,DllMain",
```


Εικόνα 53: Rundll32.exe Call

Άρα συνοψίζοντας στην εικόνα σαράντα εννέα Rundll32.exe Call βλέπουμε την δραστηριότητα ενός κακόβουλου DLL που φόρτωσε η rundll32.exe , αρχεία που εντόπισε και άνοιξε καθώς και τιμές του registry που διάβασε. Παρακάτω στην ίδια εικόνα βλέπουμε το πεδίο file_failed που μας ενημερώνει ότι τα συγκεκριμένα αρχεία δεν εκτελέστηκαν. Αυτό έχει σαν αποτέλεσμα να μην ολοκληρώνεται η διαδικασία του ransomware και να μην κρυπτογραφούνται τα αρχεία, καθώς σε όποιο δείγμα δεν εμφανίζεται αυτή η αποτυχία βλέπουμε το μήνυμα πληρωμής λύτρων και αρκετές διεργασίες που επιβεβαιώνουν την ορθή λειτουργία του. Η κύρια διαφορά με το Condi συμβαίνει μόνο στα δείγματα του Netwalker που χρησιμοποιείται η rundll32.exe καθώς εδώ δεν χρησιμοποιείται απευθείας το εκτελέσιμο αλλά ένα αρχείο dll που δημιουργήθηκε από αυτό και εκτελείται από το LOLBAS rundll32.exe.

3.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Netwalker(Mailto). Η παρουσίαση γίνεται συνοπτικότερα σε σχέση με την ανάλυση του Conti καθώς πολλές από τις λειτουργίες επαναλαμβάνονται.

1) Checks for the Locally Unique Identifier on the system for a suspicious privilege
2) Communicates with host for which no DNS query was performed
3) Writes a potential ransom message to disk
4) Performs some HTTP requests
5) Queries for the computername
6) Allocates read-write-execute memory (usually to unpack itself)
7) Creates (office) documents on the filesystem
8) Creates executable files on the filesystem
9) Drops an executable to the user AppData folder
10) Installs itself for autorun at Windows startup
11) Appends a known multi-family ransomware file extension to files that have been encrypted
12) File has been identified by 55 AntiVirus engines on VirusTotal as malicious

Επιπρόσθετα συμπληρώνουμε ότι για την διαγραφή των αντιγράφων ασφαλείας χρησιμοποιείται η ίδια τακτική με το Condi καθώς γίνεται χρήση του vssadmin.exe.

Εικόνα 54: Vssadmin

3.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) One or more thread handles in other processes
3) Attempts to detect Cuckoo Sandbox through the presence of a file
4) Executed a process and injected code into it, probably while unpacking
5) Manipulates memory of a non-child process indicative of process injection
6) Resumed a suspended thread in a remote process potentially indicative of process injection
7) Stopped Application Layer Gateway service
8) Kernel module without a name
9) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
10) One or more processes crashed
11) One or more of the buffers contains an embedded PE file
12) The file contains an unknown PE resource name possibly indicative of a packer
13) Malfind detects one or more injected processes
14) Stopped Firewall service
15) Checks if process is being debugged by a debugger
16) The executable contains unknown PE section names indicative of a packer (could be a false positive)
17) One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc
18) Creates a thread using NtQueueApcThread in a remote process potentially indicative of process injection
19) Used NtSetContextThread to modify a thread in a remote process indicative of process injection
20) Changes read-write memory protection to read-execute (probably to avoid detection when setting all RWX flags at the same time)

3.4 Κρυπτογράφηση

Στην εικόνα παρακάτω βλέπουμε πως γίνονται αλλαγές στο SSL πρωτόκολλο πρόσβασης που υπήρξε στο registry key του.

Επίσης βλέπουμε έντονη χρήση της cryptsp.dll και της cryptsvc.dll από πολλαπλές διεργασίες όπως svchost.exe, spoolsv.exe, explorer.exe και άλλες.


```
"handle_granted_access": "131097",
"process_id": 612,
"handle_value": "3164",
"handle_type": "Key",
"handle_name": "MACHINE\\SOFTWARE\\POLICIES\\MICROSOFT\\CRYPTOGRAPHY\\CONFIGURATION\\SSL\\00010002"


"handle_granted_access": "131097",
"process_id": 612,
"handle_value": "3168",
"handle_type": "Key",
"handle_name": "MACHINE\\SYSTEM\\CONTROLSET001\\CONTROL\\CRYPTOGRAPHY\\CONFIGURATION\\LOCAL\\SSL\\00010002"
```


Εικόνα 55: SSL Protocol


```
{
    "dll_in_init": true,
    "init_full_dll_name": "C:\\Windows\\System32\\CRYPTSP.dll",
    "process_id": 920,
    "dll_in_mem": true,
    "mem_full_dll_name": "C:\\Windows\\System32\\CRYPTSP.dll",
    "dll_mapped_path": "\\Windows\\System32\\cryptsp.dll",
    "process_name": "svchost.exe",
    "dll_base": "0x7ffaf2210000",
    "dll_in_load": true,
    "load_full_dll_name": "C:\\Windows\\System32\\CRYPTSP.dll"
},
```


Εικόνα 56: Cryptsp.dll

Ωστόσο από τα reports της δυναμικής ανάλυσης που εκτελέσαμε μέσω του Cuckoo δεν εντοπίζουμε κάτι πιο συγκεκριμένο όσων αφορά τις μεθόδους κρυπτογράφησης οπότε προχωράμε σε μια θεωρητική έρευνα για μια πιο ολοκληρωμένη παρουσίαση.

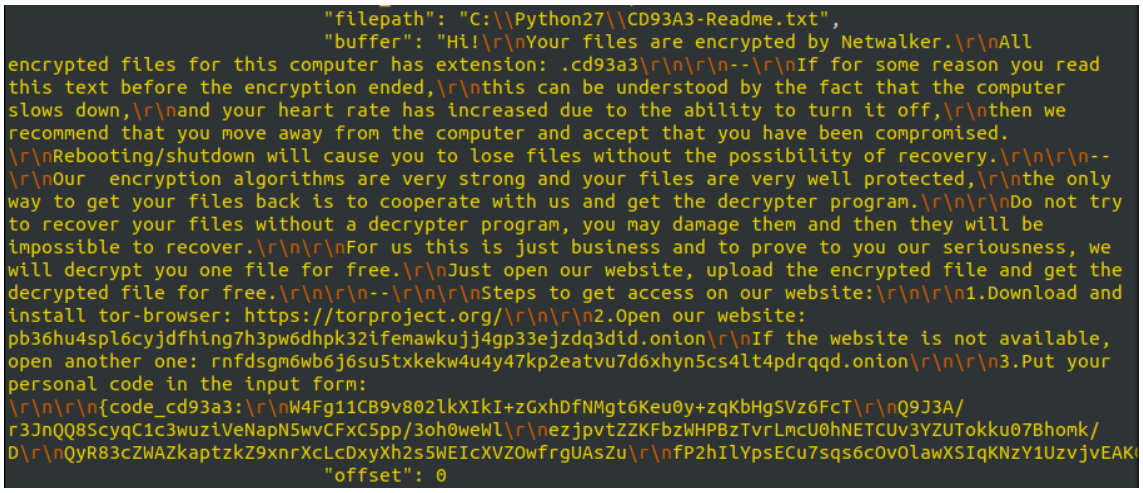
Αρχικά θα πρέπει να διατυπώσουμε πως το Netwalker(Mailto) κάνει χρήση υβριδικής κρυπτογραφίας δηλαδή συμμετρικής και ασύμμετρης μαζί. Από θεωρητικές πηγές ανακαλύψαμε πως το Netwalker(Mailto) σε πρώιμες μορφές κάνει χρήση της συμμετρικής κρυπτογράφησης Salsa20 [19]. Όσο αφορά την ασύμμετρη κρυπτογράφηση χρησιμοποιεί την Elliptic-curve Diffie-Hellman (EC-DH). Η χρήση της υβριδικής κρυπτογράφησης γίνεται αρκετά συχνά στα τελευταία δείγματα των Ransomware διότι δεν απαιτείται σύνδεση το διαδίκτυο για την διαδικασία της κρυπτογράφησης παρά μόνο στην αποκρυπτογράφηση. Η διαδικασία αυτού του είδους τις κρυπτογράφησης αναφέρεται αναλυτικά στην συνέχεια.

Στην υβριδική κρυπτογράφηση αρχικά δημιουργούνται τα τέσσερα κλειδιά για την ασύμμετρη κρυπτογράφηση, δύο για το θύμα και δύο για τον επιτιθέμενο. Ένα δημόσιο και ένα ιδιωτικό για τον χρήστη Cpriv.key και Cpub.key αντίστοιχα και πάλι ένα δημόσιο και ένα ιδιωτικό για τον επιτιθέμενο Spriv.key και Spub.key. Σε κάθε μόλυνση ξεχωριστά δημιουργούνται τα δύο κλειδιά του θύματος και το δημόσιο κλειδί του επιτιθέμενου Spub.key βρίσκεται ήδη μέσα στο εκτελέσιμο αρχείο του ransomware. Στην συνέχεια κρυπτογραφείται το Cpriv.key με το Spub.key και αποστέλλετε στον επιτιθέμενο. Έπειτα ξεκινάει η ρουτίνα κρυπτογράφησης. Όλα τα αρχεία κρυπτογραφούνται με την συμμετρική κρυπτογράφηση που έχει ορίσει το αντίστοιχο ransomware όπως για παράδειγμα η AES. Έπειτα όλα τα AES κλειδιά κρυπτογραφούνται με το Cpub.key. Οπότε για να γίνει αποκρυπτογράφηση των αρχείων χρειαζόμαστε τα AES κλειδιά που είναι κρυπτογραφημένα με το Cpub.key, οπότε χρειαζόμαστε το Cpriv.key. Όμως το Cpriv.key είναι κρυπτογραφημένο με το Spub.key, οπότε χρειαζόμαστε το Spriv.key και για να το βρούμε ο μόνος που κατέχει αυτό το κλειδί είναι ο επιτιθέμενος. Εν συντομία εκτελούνται τα παρακάτω βήματα :

1. Το συμμετρικό κλειδί κρυπτογραφεί αρχεία.

2. Το λογισμικό δημιουργεί ένα ζεύγος κλειδιών του θύματος. Το δημόσιο κλειδί του θύματος κρυπτογραφεί το αρχείο συμμετρικού κλειδιού.
3. Το λογισμικό δημιουργεί ένα ζεύγος κλειδιών του εισβολέα. Το δημόσιο κλειδί του εισβολέα κρυπτογραφεί το ιδιωτικό κλειδί του θύματος και στη συνέχεια πηγαίνει στον εισβολέα.
4. Όταν πληρωθούν τα λύτρα, το ιδιωτικό κλειδί του εισβολέα αποκρυπτογραφεί το ιδιωτικό κλειδί του θύματος και αυτό το κλειδί πηγαίνει στο θύμα όπου η αλυσίδα κρυπτογράφησης αντιστρέφεται.

Παρακάτω παρατίθεται μήνυμα προς τον χρήστη όταν ολοκληρώνεται η διαδικασία κρυπτογράφησης παραβίασης του συστήματός μας από το Netwalker(Mailto) ransomware.



Εικόνα 57: Netwalker(Mailto) Ransom Message

4 Locky

4.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του Locky με hash sha256 :

1) 40a340087cc07780bfd61eab92e40f1223a6de88ec191bdedea0b91b16eca2aa
2) 49a48d4ff1b7973e55d5838f20107620ed808851231256bb94c85f6c80b8ebfc
3) ace15d620a4d8a32324351bd7405307873f7101f113a7e022ed9ec06ee1689b9
4) df255af635a2dde04c031db95862f11e1bf44fe5cfc10d3b20bd4678ed818567

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις με hash sha256:

1) 36350904b065500f429e6b2af0c4a1ec835352fed15cad40f07760aede4fcd47

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Url.dll				

Στις αναλύσεις με hash sha256 :

1) 0a2bc257eb1e266e2fd7c608bbb7e1f2ed34660c8ff21f32999fe49c6997329b
2) d4fd2fe61b13c70740ebc900e8d88123683790a43dd500e0f660f92e9fa257dc
3) bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
4) bcafa75153d21270bb3c7cc1fa62783217ab7f5673e101fca99f9174724668b4
5) 0537fa38b88755f39df1cd774b907ec759dacab2388dc0109f4db9f0e9d191a0
6) 5b712f3ced695dd1510320494ecac67b277c0b386ee465303504c89431f87c78
7) 5e945c1d27c9ad77a2b63ae10af46aee7d29a6a43605a9bfbf35cebbcff184d8
8) 47920080055e1707943b1f993ad547e3b0ea0d1a15ff825c500ad5f934c082e6
9) 9609ad385d364afdcd4bcd9ad9b6c6cf2383e3351a254b6f4d76de6b98e940b5
10) 034af3eff0433d65fe171949f1c0f32d5ba246d468f3cf7826c42831a1ef4031
11) 03f6ab1b482eac4acfb793c3e8d0656d7c33cddb5fc38416019d526f43577761

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll				

Στις αναλύσεις με hash sha256 :

1) 6d5d672d9e8402a4e6a2309c71443e93efccccee8f9959afc24ae9a89fe2935c
2) 8514a2eca4090f400a43c4af915eb3ef6e9c15dabe69716189e7c68c72cfa285

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll	Comsvcs.dll			

Στις αναλύσεις με hash sha256 :

1) 21a0201874af80436dc0a36e5cbaf7da9b75217b3e39b712f3850729cf47deb6

2) f6045c3d60fb2e0ddbb264cd61adc37736508471aa5b3881f2510ec36ea6c00f

Χρησιμοποιούνται τα παρακάτω LOLBAS αρχεία όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll	Winword.exe	Excel.exe	Powerpnt.exe	

Λειτουργικότητα και Ροή Διεργασιών

Η διεργασία explorer.exe καλείται με τον ίδιο τρόπο όπως στο Conti και στο Netwalker παραπάνω σε όλα τα δείγματά μας. Η μόνη διαφορά όταν φτάνουμε στο κύριο εκτελέσιμο είναι ότι οι αναλύσεις σταματάνε να μας δίνουν πληροφορίες μετά το κάλεσμα του. Η ροή των διεργασιών έχει ως εξής : Smss(418) → Winlogon(588) → Userinit(2664) → explorer.exe(2788) python(4240) → python(2924 ή 3428) → inject-x86.exe(2632) → Κύριο Εκτελέσιμο(2832)

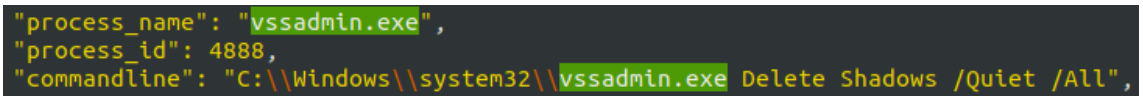
Οι διεργασίες Dllhost.exe, Winword.exe, Excel.exe, Powerpnt.exe και τα DLL αρχεία Shell32.dll, Setupapi.dll, Shdocvw.dll, Zipfldr.dll Comsvcs.dll εντοπίζονται στις αναλύσεις χωρίς να δείχνουν κάποια κακόβουλη δραστηριότητα.

4.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Locky.

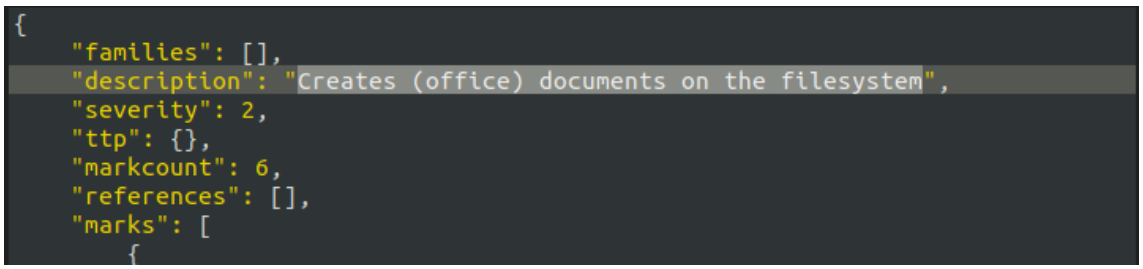
1) This executable has a PDB path
2) Queries for the computername
3) Checks for the Locally Unique Identifier on the system for a suspicious privilege
4) Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
5) Allocates read-write-execute memory (usually to unpack itself)
6) Communicates with host for which no DNS query was performed
7) Checks adapter addresses which can be used to detect virtual network interfaces
8) Writes a potential ransom message to disk
9) Performs some HTTP requests
10) Creates (office) documents on the filesystem
11) Creates executable files on the filesystem
12) Installs itself for autorun at Windows startup
13) File has been identified by 55 AntiVirus engines on VirusTotal as malicious
14) Sends data using the HTTP POST Method
15) HTTP traffic contains suspicious features which may be indicative of malware related traffic
16) Resolves a suspicious Top Level Domain (TLD)
17) Reads the systems User Agent and subsequently performs requests
18) Connects to IP addresses that are no longer responding to requests (legitimate services will remain up-and-running usually)
19) Found URLs in memory pointing to an IP address rather than a domain (potentially indicative of Command & Control traffic)

Εκτός από την πληροφορία που μας δίνουν τα signatures του Cuckoo συμπληρώνουμε ότι για την διαγραφή των αντιγράφων ασφαλείας χρησιμοποιείται η ίδια τακτική με το Condi και το Netwalker(Mailto) καθώς γίνεται χρήση του vssadmin.exe.



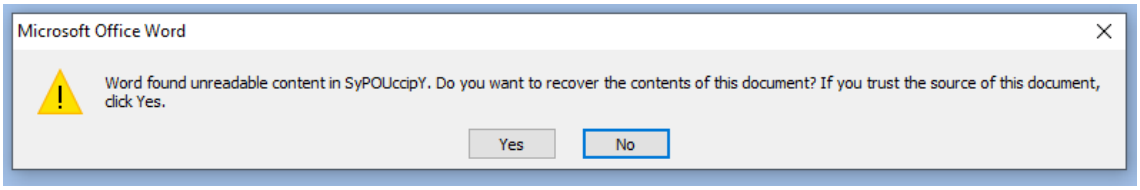
Εικόνα 58: Vssadmin Delete Shadow Copies

Σε αρκετά δείγματα του Locky αναφέρεται ότι δημιουργούνται αρχεία του Microsoft Office όπου αν ανοιχτούν με την επιλογή της επεξεργασίας του και όχι μόνο τις προβολής θα κάνουν σίγουρα μεγάλη ζημιά στον χρήστη [6]. Συνεχίζοντας επιστρέψαμε στα προηγούμενα Ransomware και ανακαλύψαμε ότι και αυτά δημιουργούν χωρίς όμως να εντοπίζει την λειτουργία το Cuckoo.



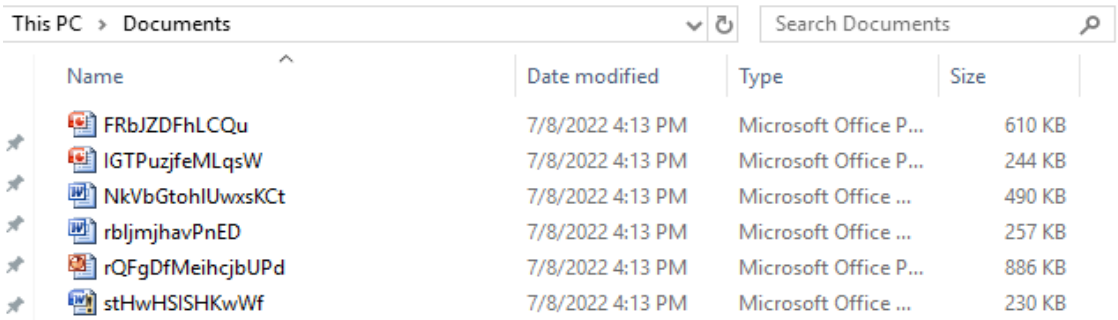
Εικόνα 59: Creates Office Documents

Στην παρακάτω εικόνα φαίνεται η προσπάθειά μας για να επεξεργαστούμε ένα Word αρχείο που δημιουργήθηκε από το Locky Ransomware και βλέπουμε ένα μήνυμα από το Word σχετικά με την περίεργη συμπεριφορά του εγγράφου.

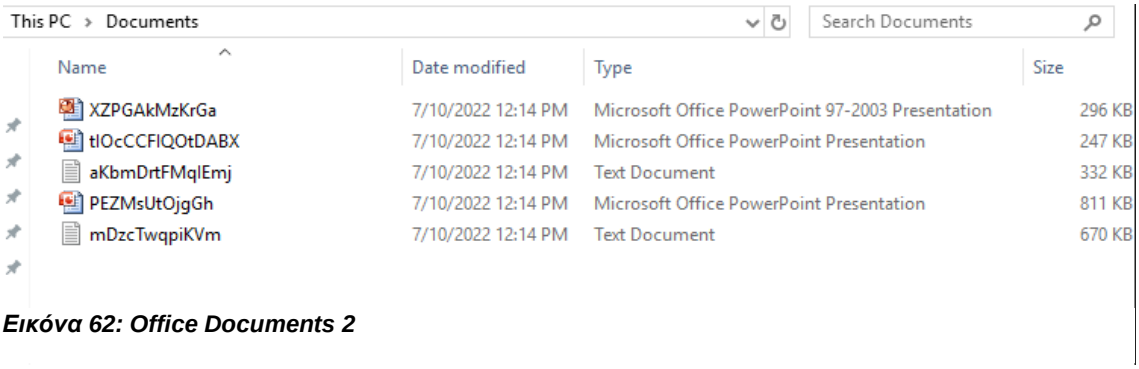


Εικόνα 60: Word Macros

Παρακάτω φαίνονται τα αρχεία που δημιουργήθηκαν από το δείγμα με hash sha256 21a02 01874af80436dc0a36e5cbaf7da9b75217b3e39b712f3850729cf47deb6 του Locky.

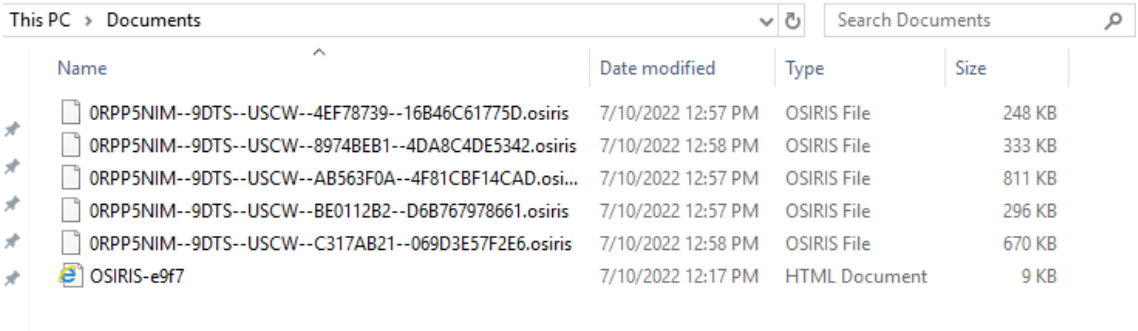


Εικόνα 61: Office Documents



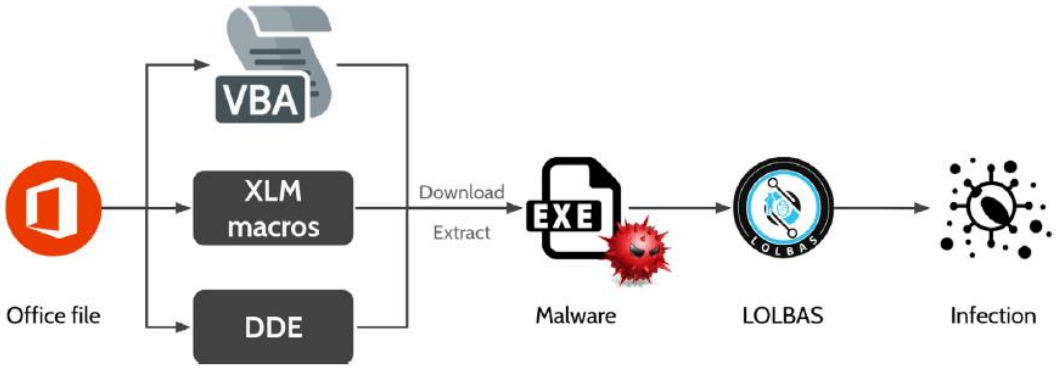
Εικόνα 62: Office Documents 2

Αναλύσαμε το δείγμα μας ξανά που και παρατηρούμε πως δεν δημιουργούνται ίδια αρχεία, καθώς ο αριθμός τους ο τύπος και τα ονόματα τους είναι διαφορετικά. Επίσης οι εκδόσεις τους είναι διαφορετικές καθώς δημιουργούνται αρχεία για Word 2007,2010 και τα λοιπά ώστε ο χρήστης να μην αντιμετωπίσει κάποιο πρόβλημα συμβατότητας στο άνοιγμα των αρχείων [20].



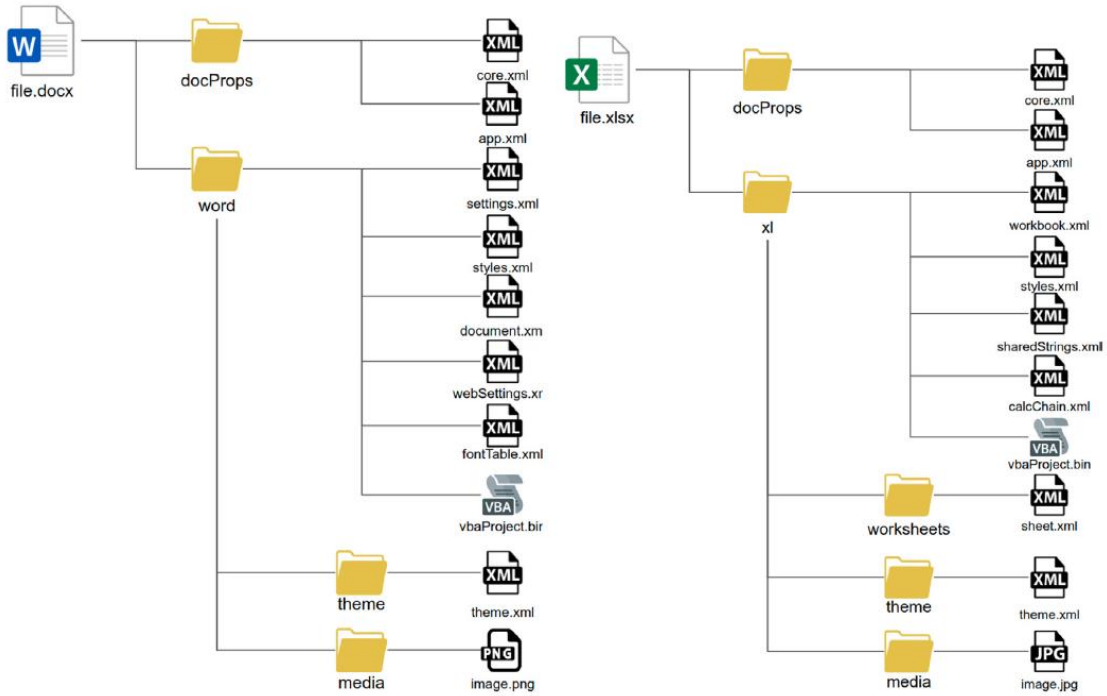
Εικόνα 63: Encrypted Office Documents

Περιμένοντας την ανάλυση να ολοκληρωθεί βλέπουμε πως τα αρχεία office που δημιουργήθηκαν κρυπτογραφούνται με κατάληξη Osiris όπου αυτό είναι το όνομα τις έκδοσης του Locky. Αυτό σημαίνει ότι τα αρχεία έχουν ως σκοπό την μόλυνση του συστήματος του χρήστη σε περίπτωση που αποτύχει η αρχική διαδικασία από κάποιο μηχανισμό ασφαλείας. Επίσης αν το συγκεκριμένο ransomware χτυπήσει έναν υπολογιστή που έχει πρόσβαση σε αρχεία κάποιου File Server ενός οργανισμού, τότε τα office documents θα δημιουργηθούν στον Server και θα υπάρξει μετάδοση σε πολλαπλά υπολογίστηκα συστήματα. Παρακάτω δίνεται μια εικόνα για τον τρόπο που μολύνουν τα αρχεία office.



Εικόνα 64: Attack through malicious MS Office documents

Παρακάτω δίνεται εικόνα της δομής των φακέλων και των αρχείων ενός Microsoft office Word και Excel.



Εικόνα 65: Structure of typical Word and Excel documents

Στις αναλύσεις παρακάτω αναλύσεις δεν τελειώνει η κρυπτογράφηση ενώ βέβαια εντοπίζονται πολλές κακόβουλες δραστηριότητες των εκτελέσιμων αρχείων. Είναι πιθανό να υπάρχει κάποιος προηγμένος anti-analysis μηχανισμός ασφαλείας που να σταματά η εκτέλεση του Locky ransomware η να μπαίνει σε αδράνεια για κάποιο χρονικό διάστημα. Οι αναλύσεις είναι :

1) 40a340087cc07780bfd61eab92e40f1223a6de88ec191bdedea0b91b16eca2aa
2) 49a48d4ff1b7973e55d5838f20107620ed808851231256bb94c85f6c80b8ebfc
3) 36350904b065500f429e6b2af0c4a1ec835352fed15cad40f07760aede4fcd47
4) ace15d620a4d8a32324351bd7405307873f7101f113a7e022ed9ec06ee1689b9
5) bc98c8b22461a2c2631b2feec399208fdc4ecd1cd2229066c2f385caa958daa3
6) bcafa75153d21270bb3c7cc1fa62783217ab7f5673e101fca99f9174724668b4
7) 0537fa38b88755f39df1cd774b907ec759dacab2388dc0109f4db9f0e9d191a0
8) 5e945c1d27c9ad77a2b63ae10af46aee7d29a6a43605a9bfbf35cebbcff184d8
9) 034af3eff0433d65fe171949f1c0f32d5ba246d468f3cf7826c42831a1ef4031
10) 6d5d672d9e8402a4e6a2309c71443e93efccccee8f9959afc24ae9a89fe2935c
11) 03f6ab1b482eac4acfb793c3e8d0656d7c33cddb5fc38416019d526f43577761
12) 9609ad385d364afdcd4bcd9ad9b6c6cf2383e3351a254b6f4d76de6b98e940b5
13) 47920080055e1707943b1f993ad547e3b0ea0d1a15ff825c500ad5f934c082e6

4.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox για το Locky Ransomware.

1) The binary likely contains encrypted or compressed data indicative of a packer

2) The file contains an unknown PE resource name possibly indicative of a packer
3) The executable uses a known packer
4) One or more thread handles in other processes
5) One or more processes crashed
6) Potentially malicious URLs were found in the process memory dump
7) Attempts to detect Cuckoo Sandbox through the presence of a file
8) Stopped Application Layer Gateway service
9) Kernel module without a name
10) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
11) Malfind detects one or more injected processes
12) Stopped Firewall service
13) Checks if process is being debugged by a debugger
14) One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc
15) Drops 64 unknown file mime types indicative of ransomware writing encrypted files back to disk
16) One or more of the buffers contains an embedded PE file
17) The executable contains unknown PE section names indicative of a packer (could be a false positive)
18) Appends a known Locky ransomware file extension to files that have been encrypted
19) Creates a known Locky ransomware decryption instruction / key file
20) Installs a hook procedure to monitor for mouse events

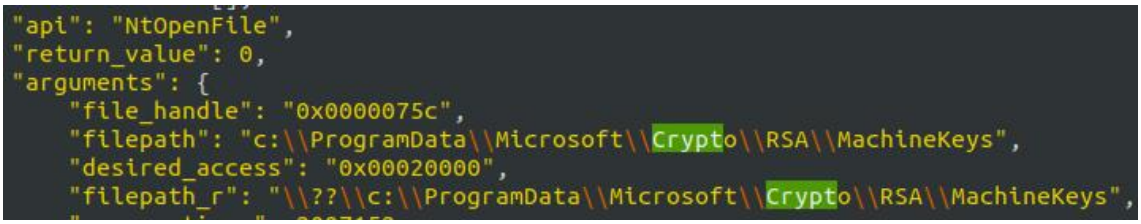
4.4 Κρυπτογράφηση

Στις αναλύσεις που ακολουθούν μπορούμε να δούμε από τα Json αρχεία το είδος της ασύμμετρης κρυπτογράφησης που χρησιμοποιήθηκε.

Στα δείγματα :

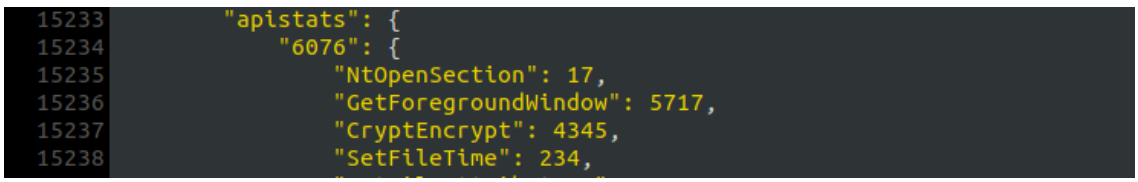
1) 0a2bc257eb1e266e2fd7c608bbb7e1f2ed34660c8ff21f32999fe49c6997329b
2) 21a0201874af80436dc0a36e5cbaf7da9b75217b3e39b712f3850729cf47deb6
3) d4fd2fe61b13c70740ebc900e8d88123683790a43dd500e0f660f92e9fa257dc
4) 5b712f3ced695dd1510320494ecac67b277c0b386ee465303504c89431f87c78
5) f6045c3d60fb2e0ddbb264cd61adc37736508471aa5b3881f2510ec36ea6c00f
6) 8514a2eca4090f400a43c4af915eb3ef6e9c15dabe69716189e7c68c72cfa285

Βλέπουμε ότι ανοίχτηκε το αρχείο με τα RSA Machinekeys άρα έχουμε να κάνουμε με RSA κρυπτογράφηση.



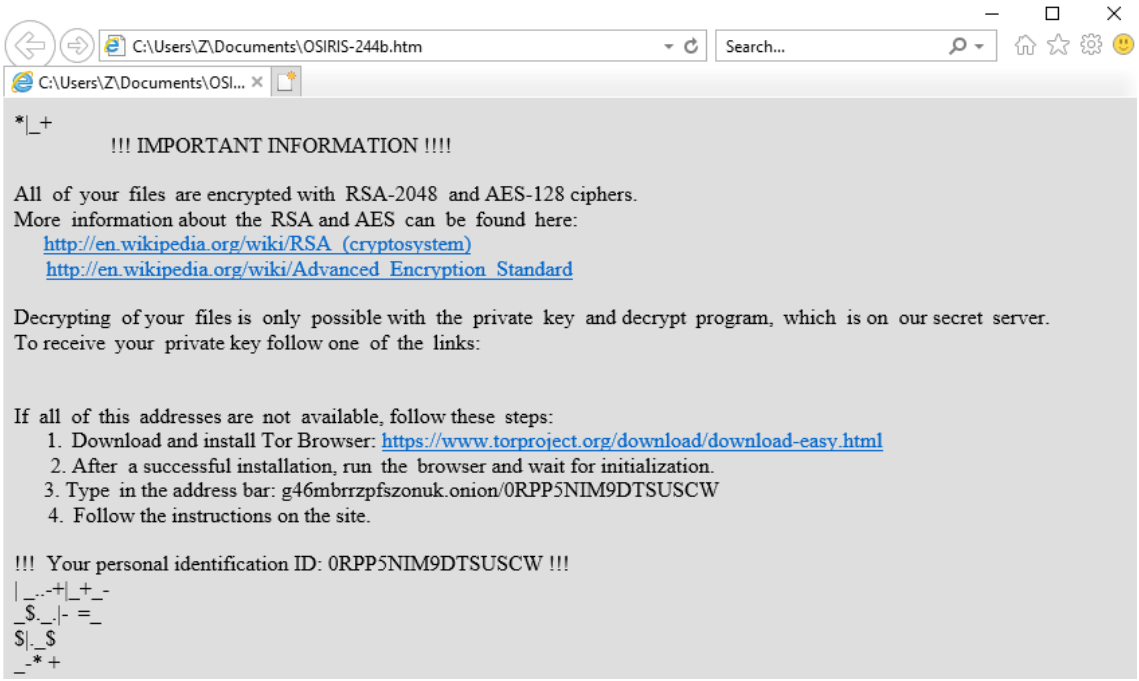
Εικόνα 66: RSA Machine Keys

Συνεχίζοντας βλέπουμε από τα Apistats ότι καλέστηκε το CryptEncrypt API 4345 φορές για την χρήση κρυπτογράφησης.



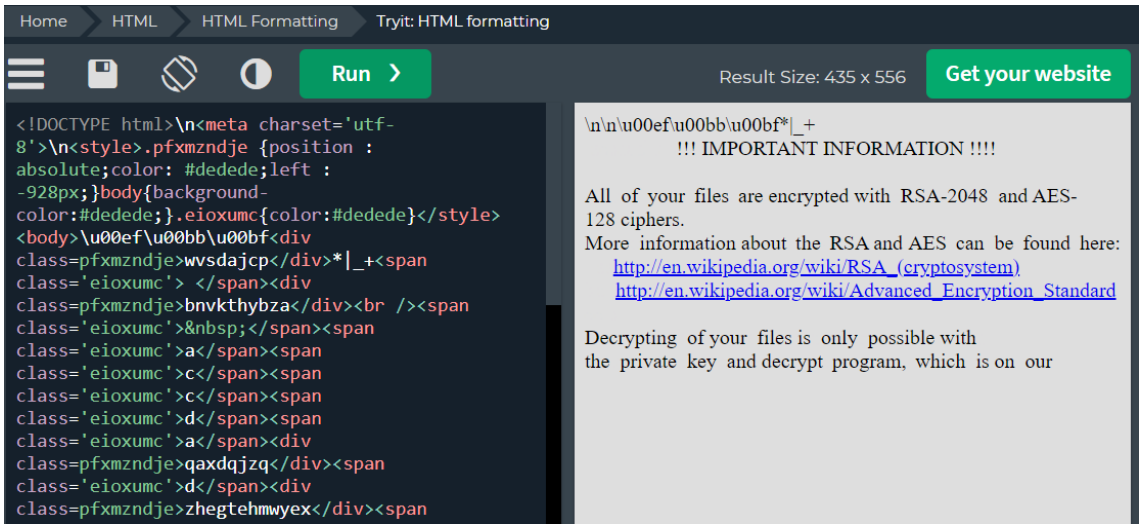
Εικόνα 67: CryptEncrypt

Τέλος βλέπουμε και από το μήνυμα λύτρων ότι ο τύπος κρυπτογράφησης που χρησιμοποιείται από το Locky είναι για την ασύμμετρη RSA-2048 και για την συμμετρική AES-128 με ECB mode [21]. Αυτή την πληροφορία την διασταυρώσαμε και θεωρητικά από άλλες σχετικές έρευνες.



Εικόνα 68: Locky Ransom Message

Μια εμφανής διαφορά με τα προηγούμενα Ransomware είναι ότι το μήνυμα πληρωμής λύτρων εμφανίζεται στον browser μας από αρχείο html. Στο Json file το μήνυμα είναι γραμμένο σε html και δεν μπορούμε να το δούμε από εκεί αμέσως οπότε πάμε σε ένα site που μετατρέπει html σε txt μπορούμε να διαβάσουμε ότι είχε φορτωθεί στον buffer όπως στην παρακάτω εικόνα.



Εικόνα 69: Locky Ransom message from Json file

5 Revil/Sodinokibi

5.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του Revil/Sodinokibi με hash sha256 :

1) 1fb2178279b76d9ed5e3c24b24885a31ec521e58a4688597600d26f88df5b8e1
2) 02a526d298e49da19220022a71e31434183e520c2845c6f3eb23169602f0ed63
3) 2d73ce9f8e11bbbce1bec1147bf30ef60a6d362504fbf650b3c8a0ea6f7c4fbb
4) 2df2fab33c1db5b049284a6bd5aa1f58bec4cb370b0663870b6a57ef33b5028c
5) 7e70460a2e395485fe6328a58e09400fc7963afeb29fdadb924c340e758b267c
6) 8c716101e118ac65d7bdb900e0100d012256abb1d7cdf64830e5943a795ccce2
7) 9b11711efed24b3c6723521a7d7eb4a52e4914db7420e278aa36e727459d59dd
8) 12d8bfa1aeb557c146b98f069f3456cc8392863a2f4ad938722cd7ca1a773b39
9) 15e3b8471e4afea6e07423d88c141e9ae454c27fa901fe320f0885298426abec
10) 93c4b144a4ef5e9ebcb5de425f6151fb6fd892d1042b21e639ab6c358cad3940
11) 42438a67636a6981b4e3209449040f6b393f10fe0636dfca2260fc0f4271e135
12) 329983dc2a23bd951b24780947cb9a6ae3fb80d5ef546e8538dfd9459b176483
13) 52612bceee07152f2e2e6699b3c085149e11979f34fe248bda14e03a0d950e85
14) 66060484cccedb839fb646d4e6020e079319374b2847c52dcec55c5ad60b1beb
15) 7394136299802ff82b9b08a43b196a803949be752fb9efd378d4936ff91bcb90
16) ab0aa003d7238940cbdf7393677f968c4a252516de7f0699cd4654abd2e7ae83
17) c2cf2118550a0fd7f81fe9913fe36be24c03a0ae5430b94557e0ee71c550a58c
18) d55f983c994caa160ec63a59f6b4250fe67fb3e8c43a388aec60a4a6978e9f1e
19) d74f04f0b948d9586629e06e2a2a21bdf20d678e47058afb637414eb3701c1f6
20) d91f951bdcf35012ac6b47c28cf32ec143e4269243d8c229f6cb326fd343df95

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Λειτουργικότητα και Ροή Διεργασιών

Η διεργασία explorer.exe καλείται με τον ίδιο τρόπο όπως και στις αναλύσεις των προηγούμενων Ransomware σε όλα τα δείγματά μας. Οι αναλύσεις σταματάνε να μας δίνουν πληροφορίες μετά το κάλεσμα της διεργασίας του κύριου εκτελέσιμου. Η συνηθισμένη ροή των διεργασιών έχει ως εξής : Smss → Winlogon → Userinit → Explorer.exe → Python.exe → Python.exe → inject-x86.exe → Κύριο Εκτελέσιμο. Σε μερικά όμως δείγματα εντοπίσαμε μερικές διαφορές με τα προηγούμενα.

Στο δείγμα 329983dc2a23bd951b24780947cb9a6ae3fb80d5ef546e8538dfd9459b176483 εντοπίσαμε πως το κύριο εκτελέσιμο καλεί την διεργασία powershell.exe version 1.0. Την συγκεκριμένη έκδοση την καλεί καθώς προσφέρει ελάχιστες έως καθόλου δυνατότητες καταγραφής. Συνεχίζοντας η powershell.exe(1492) καλεί την conhost.exe(4380) όπου η τελευταία χρησιμοποιείται για να καλέσει το API Process32NextW [22] για την ανίχνευση και την διακοπή διεργασιών που μπορεί να παρουσιάσουν πρόβλημα στην λειτουργία του Revil/Sodinokibi.


```
"process_path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
"process_name": "powershell.exe",
"pid": 1492,
"summary": {},
"first_seen": 1644883908.102865,
"ppid": 5392
```

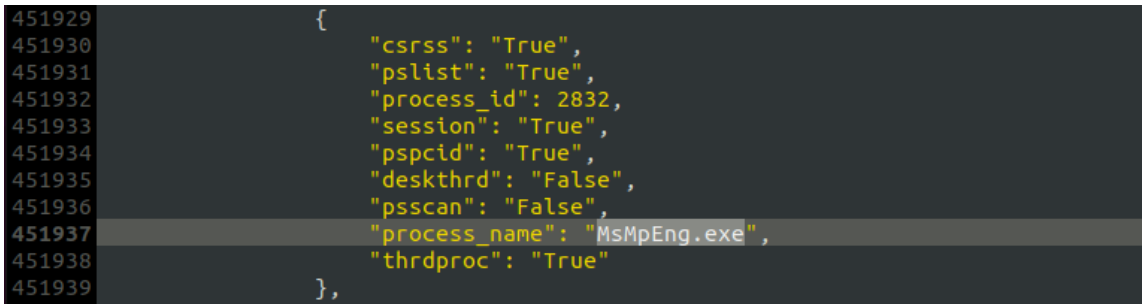

Εικόνα 70: Powershell.exe version 1

Στο δείγμα c2cf2118550a0fd7f81fe9913fe36be24c03a0ae5430b94557e0ee71c550a58c εντοπίσαμε πως το κύριο εκτελέσιμο καλεί επιπρόσθετα την διεργασία conhost.exe(2576) χωρίς να δίνονται περισσότερα στοιχεία.

Στο δείγμα c2cf2118550a0fd7f81fe9913fe36be24c03a0ae5430b94557e0ee71c550a58c εντοπίσαμε πως το κύριο εκτελέσιμο καλεί επιπρόσθετα την διεργασία MsMpEng.exe(2832). Το αρχείο αυτής της διεργασίας δημιουργήθηκε από το κύριο εκτελέσιμο με βασικό λόγο το sandbox evasion καθώς όπως βλέπουμε προσπαθεί να καθυστερήσει την διαδικασία ανάλυσης ώστε να μην καταγραφούν δραστηριότητες. Συνεχίζοντας η λειτουργία του δείγματος γίνεται μερικώς από το κύριο εκτελέσιμο και κυρίως από την διεργασία με όνομα MsMpEng.exe.

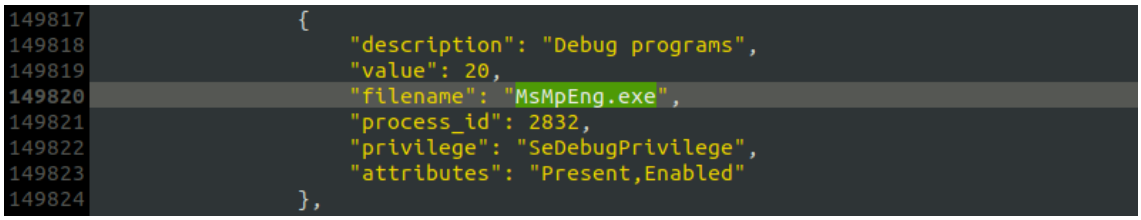
Εικόνα 71: Sleep to Delay

Πολλές διεργασίες που αναφέραμε όπως η MsMpEng.exe, το κύριο εκτελέσιμο, η inject-x86.exe και η μια python.exe φαίνεται πως είναι κακόβουλες ή έχουν παραποιηθεί όπως η explorer.exe από την λειτουργία psxview του Volatility, καθώς εντοπίζονται από το pslist αλλά όχι και από το psscan, πράγμα που καθιστά την διεργασία ύποπτη.



Εικόνα 72: MsMpEng.exe and Volatility

Επίσης εντοπίζονται από την αλλαγή δικαιωμάτων στο πεδίο privs κοιτώντας τα attributes κάθε διεργασίας. Αν στα attributes δεν εμφανίζεται το Default όπως παρακάτω τότε η διεργασία είναι ύποπτη.



Εικόνα 73: Present and Enabled

Το Revil/Sodinokibi για την διαγραφή των shadow copies του συστήματος χρησιμοποιεί όπως και τα προηγούμενα το vssadmin..exe delete shadows /all/quiet. Επίσης γίνεται χρήση του bcdedit.exe για την απενεργοποίηση των αυτόματων Windows recovery features αλλάζοντας τα configuration data.

Η διεργασία Dllhost.exe, και τα DLL αρχεία Shell32.dll, Setupapi.dll, Shdocvw.dll, εντοπίζονται στις αναλύσεις χωρίς να δείχνουν κάποια κακόβουλη δραστηριότητα.

Επιπρόσθετες μελέτες υποστηρίζουν ότι το Revil/Sodinokibi είναι εξοπλισμένο με προηγμένο σύστημα anti-analysis και λειτουργία για privilege escalation. Επίσης κάνει συλλογή δεδομένων από το τοπικό σύστημα, συλλογή συνθηματικών, επεξεργάζεται το Registry, κάνει εγκατάσταση Root Certificate, εξερευνά τις πληροφορίες συστήματος, εντοπίζει περιφερειακές συσκευές. Όλα αυτά έχουν στόχο την απολαβή χρηματικού με μεγάλο όπλο εκτός της απειλής διαγραφής αρχείων και το Defacement δηλαδή τον ηλεκτρονικό βανδαλισμό του θύματος ώστε να πετύχει πλήγμα στην αξιοπιστία και στο γόητρο του οργανισμού θύματος. Σημαντικό είναι επίσης ότι αποφεύγει να εκτελεστεί σε συγκεκριμένα συστήματα που εντοπίζεται η ρωσική γλώσσα. Συνεχίζοντας βλέπουμε ειδικότερα του μηχανισμούς λειτουργίας.

5.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Revil/Sodinokibi.

1) This executable has a PDB path
2) Queries for the computername
3) Tries to locate where the browsers are installed
4) Command line console output was observed
5) Steals private information from local Internet browsers
6) Creates a shortcut to an executable file
7) Attempts to stop active services
8) Allocates read-write-execute memory (usually to unpack itself)
9) Executes one or more WMI queries
10) Communicates with host for which no DNS query was performed
11) Appends a known multi-family ransomware file extension to files that have been encrypted
12) Performs some HTTP requests
13) Creates executable files on the filesystem
14) Installs itself for autorun at Windows startup
15) Expresses interest in specific running processes
16) File has been identified by 55 AntiVirus engines on VirusTotal as malicious

5.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping
3) Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation
4) Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available
5) The executable contains unknown PE section names indicative of a packer (could be a false positive)
6) One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.
7) A process attempted to delay the analysis task
8) One or more thread handles in other processes
9) One or more processes crashed
10) Potentially malicious URLs were found in the process memory dump
11) Attempts to detect Cuckoo Sandbox through the presence of a file
12) Stopped Application Layer Gateway service
13) Kernel module without a name
14) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
15) Malfind detects one or more injected processes
16) Stopped Firewall service
17) Checks if process is being debugged by a debugger
18) Resumed a suspended thread in a remote process potentially indicative of process injection

5.4 Κρυπτογράφηση

Σε μερικά αρχεία αποτελεσμάτων Json μπορούμε να δούμε το είδος της ασύμμετρης κρυπτογράφησης που χρησιμοποιήθηκε. Από το δείγμα 1fb2178279b76d9ed5e3c24b24885a31ec521e58a4688597600d26f88df5b8e1 εντοπίσαμε ότι γίνεται χρήση του RSA αλγόριθμου καθώς ανοίχτηκε το περιεχόμενο με στην τοποθεσία C:\\Users\\Z\\AppData\\Roaming\\Microsoft\\Crypto\\RSA.

Εικόνα 74: Crypto\\RSA

Τα αποτελέσματα των αναλύσεων δεν παρουσιάζουν ακριβή στοιχεία κρυπτογράφησης ωστόσο από θεωρητικές πηγές εντοπίσαμε πως το συγκεκριμένο ransomware κρυπτογραφεί τους φακέλους μέσω του salsa20 stream cipher ή του curve25519. Τα κλειδιά κρυπτογραφούνται με curve25519 ή AES-256-CTR [23]. Η κρυπτογράφηση χρησιμοποιεί ένα ξεχωριστό κλειδί για κάθε φάκελο βασισμένο στο session public key. Επίσης υπάρχει C2 (Command and Control) obfuscation μέσω μια μεγάλης λίστας Domains.

Όσων αφορά το μήνυμα πληρωμής λύτρων αξίζει να σημειωθεί ότι δεν εντοπίστηκε σε όλες τις αναλύσεις αλλά σε μερικές πράγμα που σημαίνει ότι τα δείγματα που δεν εμφάνισαν το μήνυμα προφανώς δεν εκτελέστηκαν πλήρως και έχουν εξελιγμένες μεθόδους προστασίας από αυτόματη δυναμική ανάλυση. Ακόμα και αν εντοπίσαμε όμως το μήνυμα πληρωμής λύτρων αυτό ήταν κρυπτογραφημένο μέσα στο αρχείο των Json αναφορών. Στα παρακάτω δείγματα εντοπίσαμε το μήνυμα κρυπτογραφημένο :

1) 1fb2178279b76d9ed5e3c24b24885a31ec521e58a4688597600d26f88df5b8e1
2) 8c716101e118ac65d7bdb900e0100d012256abb1d7cdf64830e5943a795ccce2
3) d91f951bdcf35012ac6b47c28cf32ec143e4269243d8c229f6cb326fd343df95

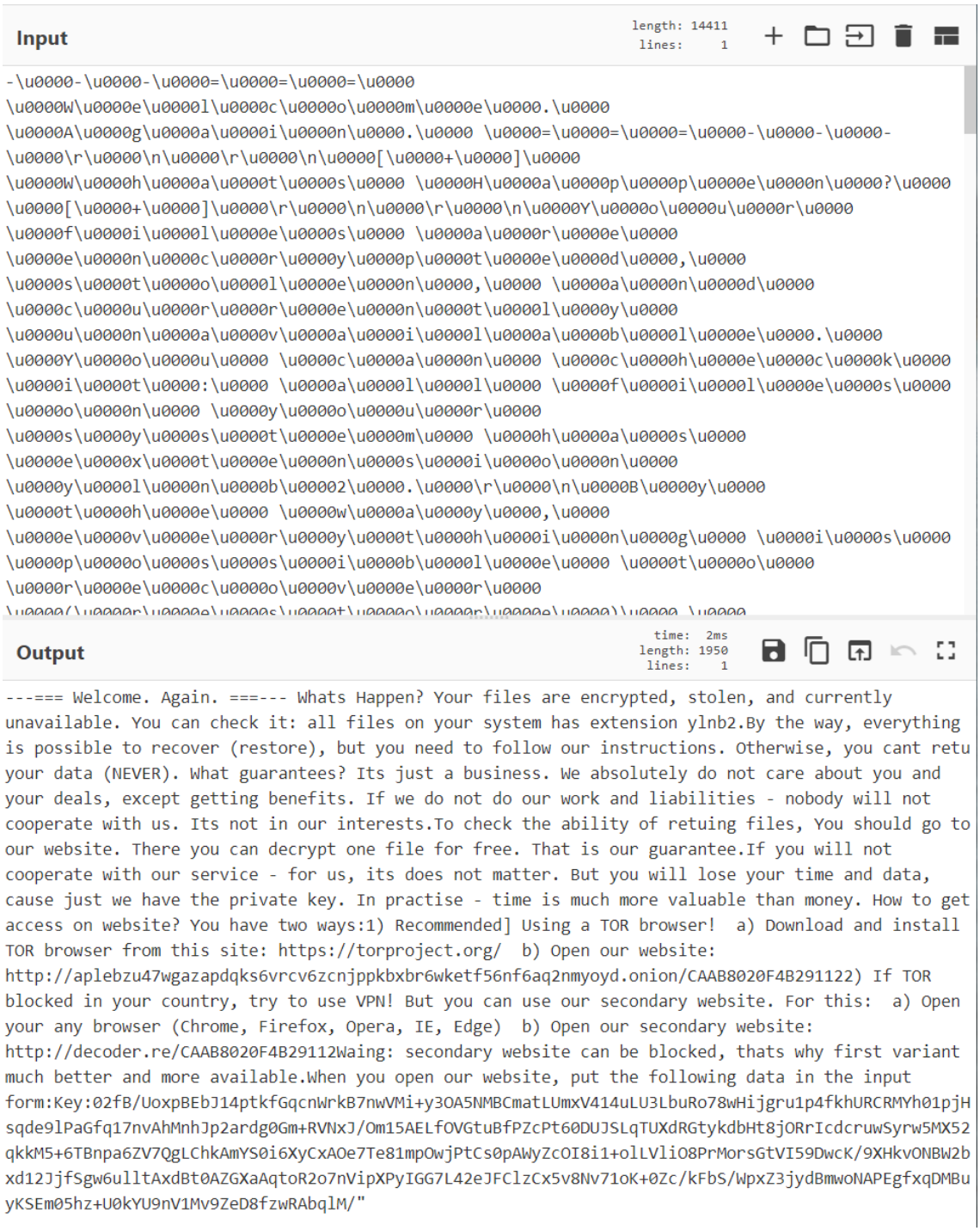
Παρακάτω φαίνεται ένα παράδειγμα του πως φαίνεται το περιεχόμενο του αρχείου readme.txt φορτωμένο στον buffer και η μορφή που έχει.

Εικόνα 75: Readme.txt in Buffer

Για να μπορέσουμε να διαβάσουμε το μήνυμα κάναμε χρήση του online CyberChef όπου είναι ένα εργαλείο αναπτυγμένο από την GCHQ(Government Communications Headquarters) για την αποκρυπτογράφηση πληροφοριών. Κάνοντας αντιγραφή το κρυπτογραφημένο μήνυμα στο CyberChef μέσω μερικών δοκιμών βρήκαμε την συνταγή για να μπορέσουμε να αναγνώσουμε το μήνυμα. Παρακάτω δίνεται η συνταγή σε Chef Format για την αποκρυπτογράφηση του μηνύματος από το Json report του Cuckoo Sandbox και τέλος η εικόνα με τα αποτελέσματα όπου πετύχαμε με μεγάλη ακρίβεια.

```
Find_/_Replace({'option':'Regex','string':'u0000'},'',true,false,true,false)
Find_/_Replace({'option':'Regex','string':'\\\\'},'',true,false,true,false)
Find_/_Replace({'option':'Regex','string':'rn'},'',true,false,true,false)
```

Find_/_Replace({'option':'Regex','string':'\\+]'},'',true,false,true,false)
Find_/_Replace({'option':'Regex','string':'\\['},'',true,false,true,false)



Εικόνα 76: Revil/Sodinokibi Ransom Message

6 DarkSide

6.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του DarkSide με hash sha256 :

1) 1d4c0b32aea68056755daf70689699200ffa09688495ccd65a0907cade18bd2a
2) 2d82be244e23001148ed5a6d83856b6f7cd20c3f7786481303d5d584c51ff5f0
3) 2de09a815efcc64810046de69b8e0aa1c9e9beee77b66560a0b15d737485e3c5
4) 4d9432e8a0ceb64c34b13d550251b8d9478ca784e50105dc0d729490fb861d1a
5) 029c5d48e425206e2ae84a63d62bdbc80362702913b38618a423c541c8a0ed40
6) 56e7b9c4b8962b6ff0d1e0162ca8515a07b576cd47ba90221354838733f8689a
7) 61ca175c2f04cb5279f8507e69385577cf04e4e896a01d0b5357746a241c7846
8) 973dfafc3051d8c2849f62c556ab8057da706f15d1ffd8871de894ae3a24d86b
9) 4098b54c9d27b00ce34d04ffac24213ed28993a2854827851b157d63407c2e4e
10) 6931b124d38d52bd7cdef48121fda457d407b63b59bb4e6ead4ce548f4bbb971
11) 78782fd324bc98a57274bd3fff8f756217c011484ebf6b614060115a699ee134
12) 533672da9d276012ebab3ce9f4cd09a7f537f65c6e4b63d43f0c1697e2f5e48d
13) 516664139b0ddd044397a56482d7308d87c213c320a3151ccb9738e8f932654b
14) a11cc5051e3a88428db495f6d8e4b6381a1cb3fa5946a525ef5c00bfcb44e210
15) ac092962654b46a670b030026d07f5b8161cecd2abd6eece52b7892965aa521b
16) bc32a2ccf158ebe2b76646be865a4c6dd91da6b8e5bb0dd9520102a9bfea5439
17) bfb31c96f9e6285f5bb60433f2e45898b8a7183a2591157dc1d766be16c29893
18) cc54647e8c3fe7b701d78a6fa072c52641ac11d395a6d2ffaf05f38f53112556
19) e0493b082077648eb33ca1294f2b26bc4c96d3820913c46330923e8bb3d73230
20) f3f25af554bedfa4ee2824bb858280282bd87828d446048619dc49fe061741b4

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Λειτουργικότητα και Ροή Διεργασιών

Η διεργασία explorer.exe καλείται με τον ίδιο τρόπο όπως παραπάνω σε όλα τα Ransomware που αναλύσαμε. Όταν φτάνουμε στο κύριο εκτελέσιμο οι αναλύσεις σταματάνε να μας δίνουν πληροφορίες μετά το κάλεσμα του. Η ροή των διεργασιών στο δείγμα με sha256 1d4c0b32aea68056755daf70689699200ffa09688495ccd65a0907cade18bd2a έχει ως εξής : Smss(488) → Winlogon(588) → Userinit(2664) → explorer.exe(2788) → python(4240) - python(2004) → inject-x86.exe(1092) → Διεργασία Κύριου Εκτελέσιμου("1d4c0b32aea680") (5492). Τα δείγματα στα οποία η ανάλυση φτάνει μέχρι το κύριο εκτελέσιμο είναι :

1) 1d4c0b32aea68056755daf70689699200ffa09688495ccd65a0907cade18bd2a
2) 4d9432e8a0ceb64c34b13d550251b8d9478ca784e50105dc0d729490fb861d1a
3) 56e7b9c4b8962b6ff0d1e0162ca8515a07b576cd47ba90221354838733f8689a
4) 61ca175c2f04cb5279f8507e69385577cf04e4e896a01d0b5357746a241c7846
5) 6931b124d38d52bd7cdef48121fda457d407b63b59bb4e6ead4ce548f4bbb971
6) 533672da9d276012ebab3ce9f4cd09a7f537f65c6e4b63d43f0c1697e2f5e48d
7) ac092962654b46a670b030026d07f5b8161cecd2abd6eece52b7892965aa521b
8) f3f25af554bedfa4ee2824bb858280282bd87828d446048619dc49fe061741b4

Αρκετά αξιόλογο να σημειωθεί είναι ότι το κύριο εκτελέσιμο δεν φαίνεται πια διεργασία το καλεί, δηλαδή δεν εμφανίζεται το PPID και η σύνδεση σταματάει στην δεύτερη διεργασία Python.exe. Τα εν λόγο δείγματα με hash sha256 είναι :

1) 2d82be244e23001148ed5a6d83856b6f7cd20c3f7786481303d5d584c51ff5f0
2) 2de09a815efcc64810046de69b8e0aa1c9e9beee77b66560a0b15d737485e3c5

3) 973dfafc3051d8c2849f62c556ab8057da706f15d1ffd8871de894ae3a24d86b
4) 78782fd324bc98a57274bd3fff8f756217c011484ebf6b614060115a699ee134

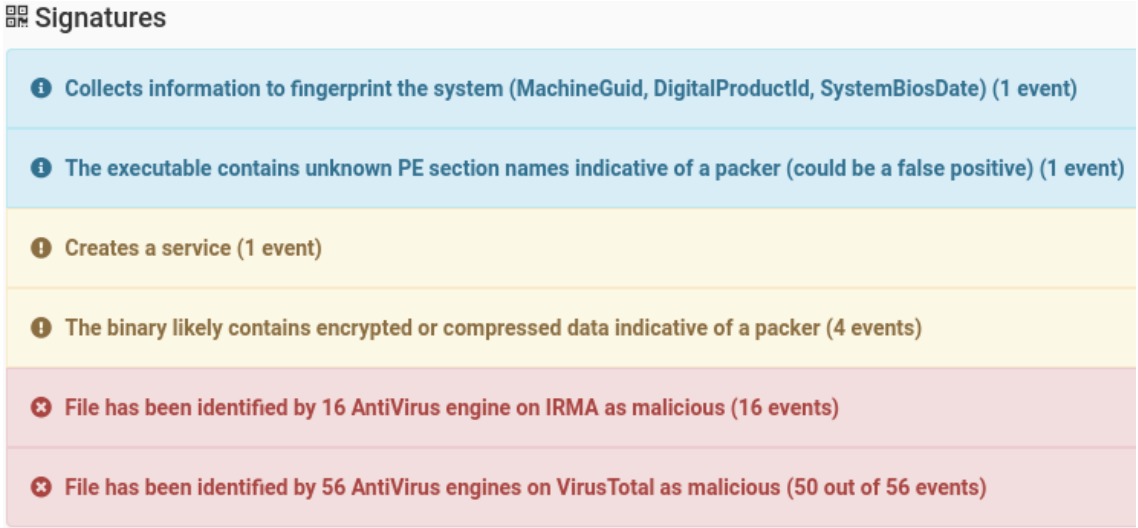
Επίσης τα παρακάτω δείγματα σταματάνε στην πρώτη διεργασία python.exe :

1) 4098b54c9d27b00ce34d04ffac24213ed28993a2854827851b157d63407c2e4e
2) 516664139b0ddd044397a56482d7308d87c213c320a3151ccb9738e8f932654b
3) a11cc5051e3a88428db495f6d8e4b6381a1cb3fa5946a525ef5c00bfcb44e210
4) bc32a2ccf158ebe2b76646be865a4c6dd91da6b8e5bb0dd9520102a9bfea5439
5) bfb31c96f9e6285f5bb60433f2e45898b8a7183a2591157dc1d766be16c29893
6) cc54647e8c3fe7b701d78a6fa072c52641ac11d395a6d2ffaf05f38f53112556
7) e0493b082077648eb33ca1294f2b26bc4c96d3820913c46330923e8bb3d73230

Στο δείγμα 029c5d48e425206e2ae84a63d62bdbc80362702913b38618a423c541c8a0ed40 μπορούμε να δούμε ότι η ροή των διεργασιών σταματάει στην διεργασία jusched.exe δηλαδή Smss(488) → Winlogon(588) → Userinit(2664) → explorer.exe(2788) → python(4240) - python(2004) → jusched.exe(1296).

Οι διεργασίες Dllhost.exe και τα DLL αρχεία Shell32.dll, Setupapi.dll, Shdocvw.dll, εντοπίζονται στις αναλύσεις χωρίς να υφίσταται κάποια κακόβουλη δραστηριότητα.

Το συγκεκριμένο ransomware πρέπει να σημειωθεί ότι δεν εκτελείται πλήρως στο δικό μας Cuckoo Sandbox αλλά το ίδιο και στο Online που βρίσκουμε στην ιστοσελίδα https://cuckoo.ee/. Αυτό μπορεί να γίνει εύκολα αντιληπτό καθώς υπάρχει φτωχή πληροφόρηση ως προς την λειτουργικότητά του και από τα δύο, αλλά και από το ότι δεν βλέπουμε να κρυπτογραφεί τα αρχεία μας και να εμφανίζεται το μήνυμα λύτρων. Για την επιβεβαίωση της υποψίας μας αναλύσαμε τα δείγματα του DarkSide σε διαφορετικά online Sandboxes όπως στο Triage και στο Anyrun, και είδαμε την ολοκληρωτική λειτουργία του με την κατάληξη της κρυπτογράφησης αλλά και του μηνύματος. Ωστόσο μια μεγάλη αδυναμία του Cuckoo είναι ότι δεν μπορεί να εντοπίσει την μερική εκτέλεση και να την αναφέρει έτσι που μπορεί εύκολα να εξαπατηθεί ένας αρχάριος ερευνητής. Στην εικόνα φαίνονται τα λιγοστά signatures που εντόπισε το online Cuckoo για το δείγμα 1d4c0b32aea680 56755daf70689699200ffa09688495ccd65a090 7cade18bd2a.



Εικόνα 77: Too Few Signatures

Ωστόσο θα δώσουμε μερικές πληροφορίες που εντοπίσαμε σε άλλες έρευνες που έχουν γίνει για το DarkSide Ransomware. Αναφέρεται πως μετά την πρόσβαση στο υπολογιστικό σύστημα από τους επιτιθέμενους γίνεται χρήση του powershell για την εγκατάσταση και την λειτουργία του κακόβουλου λειτουργικού και συγκεκριμένα χρησιμοποιούν το Certutil και το Bitsadmin για το κατέβασμα του Ransomware. Το DarkSide ransomware έχει πολλές ομοιότητες με το REvil σε αυτό το βήμα της διαδικασίας, το οποίο περιλαμβάνει τη δομή των σημειώσεων

λύτρων. Στο ξεκίνημα της εκτέλεσης ενός δείγματος DarkSide το ransomware κάνει πολλαπλές λειτουργίες όπως σάρωση δικτύων, εκτέλεση εντολών, απόρριψη διεργασιών και κλοπή διαπιστευτηρίων. Συνεχίζοντας διαγράφονται τα shadow copies κάνοντας χρήση μιας obfuscated Powershell εντολής που παρουσιάζουμε στις παρακάτω εικόνες.

```
powershell -ep bypass -c "(0..61)|%{$s+=[char][byte]
('0x'+'4765742D576D694F626A6563742057696E33325F536861646F77636F7079207C20466F7245616368 2D4
F626A6563742074207B245F2E44656C657465528293B7D20'.Substring(2*$_,2))};iex $s"
```

Εικόνα 78: Obfuscated Powershell Command

The de-obfuscated command:

```
PS C:\Windows\system32> (0..61)|%{$s+=[char][byte]('0x'+'4765742D576D694F626A6563742057696E33325F536861646F77636F7079207
C20466F72456163682D4F626A6563742074207B245F2E44656C657465528293B7D20'.Substring(2*$_,2))};
PS C:\Windows\system32> $s
Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
```

Εικόνα 79: De-Obfuscated PowerShell Command

Τέλος το DarkSide κρυπτογραφεί τα αρχεία μας αποφεύγοντας συγκεκριμένους τύπους όπως τα 386, adv, ani, bat, bin, cab, cmd, com, cpl, cur, deskthemepack, diagcab, diagcfg, diagpkg, dll, drv, exe, hlp, icl, icns, ico, ics, idx, ldf, lnk, mod, mpa, msc, msp, msstyles, msu, nls, nomedia, ocx, prf, ps1, rom, rtp, scr, shs, spl, sys, theme, themepack, wpx, lock, key, hta, msi, pdb.

6.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του DarkSide.

1) Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
2) Creates a service
3) The executable is compressed using UPX
4) Communicates with host for which no DNS query was performed
5) Found URLs related to Tor in process memory dump (e.g. onion services, Tor2Web, and Ransomware)
6) Performs some HTTP requests
7) Allocates read-write-execute memory (usually to unpack itself)
8) Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)
9) File has been identified by 55 AntiVirus engines on VirusTotal as malicious

6.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) The executable contains unknown PE section names indicative of a packer (could be a false positive)
3) The executable is likely packed with VMProtect
4) One or more thread handles in other processes
5) Stopped Application Layer Gateway service
6) Kernel module without a name
7) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary

8) Malfind detects one or more injected processes
9) Stopped Firewall service

6.4 Κρυπτογράφηση

Τα αποτελέσματα των αναλύσεων δεν παρουσιάζουν ακριβή στοιχεία κρυπτογράφησης ωστόσο από θεωρητικές πηγές εντοπίσαμε πως το συγκεκριμένο ransomware κρυπτογραφεί τους φακέλους χρησιμοποιώντας δύο μεθόδους κρυπτογράφησης, ανάλογα με το εάν το λειτουργικό σύστημα στόχος είναι Windows ή Linux. Ως προς την συμμετρική κρυπτογράφηση χρησιμοποιείται ChaCha20 σε συνδυασμό με ασύμμετρη RSA-4096 στα δείγματα για τα Linux συστήματα, ενώ το Salsa20 με RSA-1024 χρησιμοποιείται για τα δείγματα στα Windows [24].

Το Darkside δεν αναπτύσσει τις τεχνικές κρυπτογράφησης ωσότου χαρτογραφήσει το περιβάλλον, εκμεταλλευτεί ενδιαφέροντα δεδομένα, αποκτήσει τον έλεγχο προνομιακών λογαριασμών και προσδιορίσει όλα τα συστήματα ασφαλείας, τους διακομιστές και τις εφαρμογές. Παρατηρούνται συνδέσεις με πρωτεύοντα αποθετήρια αντιγράφων ασφαλείας χρησιμοποιώντας υπηρεσίες παραβιασμένων λογαριασμών λίγο πριν την κρυπτογράφηση. Με αυτόν τον τρόπο φέρνουν τον εαυτό τους σε θέση να μεγιστοποιήσουν τη ζημιά και το κέρδος. Παρακάτω δείχνουμε το μήνυμα με τις οδηγίες πληρωμής λύτρων.

---------- [Welcome to Dark Side] ---------->

What happend?

Your computers and servers are encrypted, backups are deleted. We use strong encryption algorithms, so you cannot decrypt your data.
But you can restore everything by purchasing a special program from us - universal decryptor. This program will restore all your network.
Follow our instructions below and you will recover all your data.

Data leak

First of all we have uploaded more then 100 GB data.

Example of data:
- Accounting data
- Executive data
- Sales data
- Customer Support data
- Marketing data
- Quality data
- And more other...

Your personal leak page: http://darksidedxcftmqa.onion/blog/article/id/6/dQDclB_6Kg-c-6fJesONyHoaKh9Btl8j9Wkw2inG8O72jWaOcKbrxMWbPfKrUbHC
The data is preloaded and will be automatically published if you do not pay.
After publication, your data will be available for at least 6 months on our tor cdn servers.

We are ready:
- To provide you the evidence of stolen data
- To give you universal decrypting tool for all encrypted files.
- To delete all the stolen data.

What guarantees?

We value our reputation. If we do not do our work and liabilities, nobody will pay us. This is not in our interests.
All our decryption software is perfectly tested and will decrypt your data. We will also provide support in case of problems.
We guarantee to decrypt one file for free. Go to the site and contact us.

How to get access on website?

Using a TOR browser:
1) Download and install TOR browser from this site: https://torproject.org/
2) Open our website: http://darksidfqzcuhtk2.onion/K71D6P88YTX04R3ISCJZHMD5IYV55V9247QHJY0HJYUXX68H2P05XPRIR5SP2U68

When you open our website, put the following data in the input form:
Key:
pr9gzRnMz6qEwr6ovMT0cbjd9yT56NctfQZGIiVVLgo0ME2EQpAUyZucG9BLrOJjno5XLPvCN11TFfnlFHa42u5mJxoeR5k5RUgQAC1MC6LBUj4YOOAUyiBrR
HQSUM3pzGoEPRVOzXSZ8YqkJyFL0TDFBbWaBKQDOSo9GzKKoVRQ0Eb02F5geTPkTAqZZSfSQ6PBBlTGPSgGe2kCyuwwp7lDmRSJlNnHssMMZHVhXzyZ6fxiBY
gNiuusFK8JNI5nrtRPp3bMAc6OEddxfJWj6o2GT1Xg9j87Jp4Oyv43E1J61jLJAWBkmoBB3Gqv07mtyDW5PnmxBlNzABbLFEvJMQL23sR8nnw4svzcZHxrqD1
xRcxqyeKtsaQ5yqLvyQgMdnrI2QoCqkHYYUfBIzjO8BXyBZdmjHanXE57jdDAhjaDUUqfL917cCyJr1uwVR0Xj5lJXe8BIKHd3dFrz70CsIXFAhicOsBlFzIn
daNcAXXyL8Fg1avIXOcuEkGRDXt8Cs8b3TAB6n4DrbLJdiFjECo8yCA9pxvzqjXatumUloblWFZaUoLVYzP

!!! DANGER !!!
DO NOT MODIFY or try to RECOVER any files yourself. We WILL NOT be able to RESTORE them.
!!! DANGER !!!

Εικόνα 80: DarkSide Ransom Message

7 Ryuk

7.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του Ryuk με hash sha256 :

1) 2ec5256a7edb90b1c05c92f79e8a48c205b29e1ac910a535aa83c30b8dbbeff8
2) 3ee706f07d13cb9e617eac2b4442479634ab48f11005568c739c6dcab75052a4
3) 5b1f242aee0eabd4dffea0fe5f08aba60abf7c8d1e4f7fc7357af7f20ccd0204
4) 5e2c9d80fa4528fe9777738a9cba9ede08cdae353fd4cb2d9caf0c9801fd5711
5) 05e06709523fd798da963c2c24254de0fcca6c57e1052996798ecc74ff43b41f
6) 7faeb64c50cd15d036ca259a047d6c62ed491fff3729433fefba0b02c059d5ed
7) 9eb7abf2228ad28d8b7f571e0495d4a35da40607f04355307077975e271553b8
8) 88b1b4966650de59cef20c340b28739c52dc9ead91d9959a338a8e531ad38335
9) 92f124ea5217f3fe5cbab1c37a961df0437d5a9cbde1af268c60c4b3194b80ed
10) 180f82bbedb03dc29328e32e054069870a1e65078b78b2120a84c96aaed7d843
11) 781bc4dcbd459893397a8b987bf697f5b95435dfaf7fe3f4d2224728e7a2202a
12) 8862b060db997bc9077e3bece06529c1c116af379985f6138a07ab5fde61b54c
13) d0d7a8f588693b7cc967fb4069419125625eb7454ba553c0416f35fc95307cbe
14) d5d744e0f7984ec01593da35f26bf24e95e4b1cc8bd1c0ff4f31de5dbf94e38f
15) d7333223dcc1002aae04e25e31d8c297efa791a2c1e609d67ac6d9af338efbe8
16) ec3da4ac9ec917e66ab943ab149119807922f64f2e4960ebadc36fe7520b300f

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις του Ryuk με hash sha256 :

1) 40b865d1c3ab1b8544bcf57c88edd30679870d40b27d62feb237a19f0c5f9cd1
2) bbbf38de4f40754f235441a8e6a4c8bdb9365dab7f5cfcdac77dbb4d6236360b
3) cfdc2cb47ef3d2396307c487fc3c9fe55b3802b2e570bee9aea4ab1e4ed2ec28
4) cfe1678a7f2b949966d9a020faafb46662584f8a6ac4b72583a21fa858f2a2e8

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Wmic.exe	Cmd.exe			

Λειτουργικότητα και Ροή Διεργασιών

Η ροή των διεργασιών στο πιο εκδηλωτικό δείγμα με sha256 40b865d1c3ab1b8544bcf57c88edd30679870d40b27d62feb237a19f0c5f9cd1 έχει ως εξής : Smss(488) → Winlogon(588) → Userinit(2664) → explorer.exe(2788) python(4240) → python(3952) → inject-x86.exe(4968) → Διεργασία Κύριου Εκτελέσιμου("40b865d1c3ab1b") (1920) → DXwlLgw.exe (2332) → icacls.exe(5436) → conhost.exe (3584). Πολύ ενδιαφέρον παρουσιάζουν οι καινούριες διεργασίες που βλέπουμε εδώ από ότι στα προηγούμενα Ransomware και θα τις εξηγήσουμε διεξοδικά.

Μέσα στα αποτελέσματα του Json αρχείου για το συγκεκριμένο δείγμα μπορούμε να δούμε ότι το εκτελέσιμο αρχείο DXwlLgw.exe καλείται μέσω του command line με παραμέτρους 8 LAN. Από την ανάγνωση σχετικών ερευνών για τις συγκεκριμένη εντολή στο Ryuk ανακαλύψαμε ότι όταν γίνεται χρήση αυτής της εντολής το Ryuk σαρώνει τον πίνακα ARP της συσκευής ο οποίος είναι μια λίστα με γνωστές διευθύνσεις IP στο δίκτυο και τις σχετικές διευθύνσεις mac και θα ελέγξει εάν οι καταχωρήσεις αποτελούν μέρος των ιδιωτικών υποδικτύων

διευθύνσεων IP που ξεκινούν με "10.", "172.16." και "192.168." Αν βρεθούν καταχωρήσεις το Ryuk θα στείλει ένα πακέτο Wake-on-Lan (WoL) στη διεύθυνση MAC των συσκευών για να τις ενεργοποιήσει. Αυτό το αίτημα WoL έρχεται με τη μορφή ενός πακέτου που περιέχει «FF FF FF FF FF FF FF FF FF FF FF FF». Εάν το αίτημα WoL ήταν επιτυχές, ο Ryuk θα προσπαθήσει στη συνέχεια να κάνει Mount το κοινόχρηστο στοιχείο διαχείρισης C της απομακρυσμένης συσκευής. Αν το Mount γίνει επιτυχώς τότε το Ryuk θα κρυπτογραφήσει και το κοινόχρηστό στοιχείο επίσης κοιτώντας πρώτα αν εκτελείται από εικονικό μηχάνημα με πολλαπλούς τρόπους όπως εξετάζοντας διεργασίες του συστήματος. Από αυτό το στοιχείο φαίνεται πως το Ryuk είναι ένα Network Ransomware που μπορεί να επηρεάσει πολλαπλά συστήματα προσεγγίζοντας τα μέσω WOL & ARP και δείχνει πως μπορεί να διαχειριστεί μεγάλα εταιρικά περιβάλλοντα.

```
    ],
    "command_line": [
        "C:\\Users\\Z\\AppData\\Local\\Temp\\DXwILgw.exe 8 LAN"
    ],
```

Εικόνα 81: Wake on Lan

Ακολούθως καλείται η διεργασία icacls.exe όπου τροποποιεί και εφαρμόζει λίστες ελέγχου πρόσβασης σε καθορισμένα αρχεία. Παρακάτω βλέπουμε την εντολή η οποία αυτό που κάνει με τα γράμματα είναι να παραχωρεί σε όλους του χρήστες την άδεια να έχουν πλήρη έλεγχο σε όλους τους φακέλους, να συνεχίσει η εντολή ακόμα αν υπάρξει μήνυμα λάθους και τέλος να μην υπάρξει μήνυμα ενημέρωσης στον χρήστη.

```
{
    "process_name": "icacls.exe",
    "process_id": 5436,
    "commandline": "icacls \"C:\\*\" /grant Everyone:F /T /C /Q",
```

Εικόνα 82: File Access with icalcs.exe

Η λειτουργία της διεργασίας conhost.exe εκτελείται για να κάνουμε χρήση του Command Prompt, οπότε γίνεται αντιληπτό ότι το Ryuk τρέχει εντολές για τον εντοπισμό και την πρόσβαση σε συγκεκριμένα αρχεία.

Μέσω της παρακάτω εντολής το Ryuk πετυχαίνει persistence στο σύστημα ώστε να συνεχίσει την εξερεύνηση ευαίσθητων δεδομένων και την μόλυνση περισσότερων συστημάτων.

```
"/C REG ADD \"HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"svchos\" /t REG_SZ /d \"",
```

Εικόνα 83: Persistence Command

Το Ryuk πετυχαίνει επίσης privilege escalation μέσω αλλαγής στο όρισμα SeDebugPrivilege της συνάρτησης AdjustTokenPrivileges.

```
12504                {
12505                    "description": "Create a token object",
12506                    "value": 2,
12507                    "filename": "lsass.exe",
12508                    "process_id": 612,
12509                    "privilege": "SeCreateTokenPrivilege",
12510                    "attributes": "Present,Enabled"
12511                },
```

Εικόνα 84: SeDebugPriviledge, AdjustTokenPriviledges

Συνεχίζοντας γίνεται process enumeration, code injection, τερματισμός ορισμένων διεργασιών και διαγραφή των shadow copies αρχείων. Καταλήγοντας γίνεται η κρυπτογράφηση των αρχείων και εμφανίζεται το μήνυμα λύτρων.

Στο δείγμα 3ee706f07d13cb9e617eac2b4442479634ab48f11005568c739c6dcab75052a4 το δέντρο των διεργασιών σταματάει όταν το κύριο εκτελέσιμο καλεί την διεργασία GcpTeQYFQlan.exe. Η λειτουργικότητα σε σχέση με το προηγούμενο δείγμα είναι παρόμοια εκτός του ότι δεν εμφανίζεται κάπου η διεργασία icacls.exe

Η λειτουργία του Ryuk με βάση τα Json reports σταματάει στο κύριο εκτελέσιμο του καθενός στα δείγματα :

1) 2ec5256a7edb90b1c05c92f79e8a48c205b29e1ac910a535aa83c30b8dbbeff8
2) 5e2c9d80fa4528fe9777738a9cba9ede08cdae353fd4cb2d9caf0c9801fd5711
3) 05e06709523fd798da963c2c24254de0fcca6c57e1052996798ecc74ff43b41f
4) 7faeb64c50cd15d036ca259a047d6c62ed491fff3729433fefba0b02c059d5ed
5) 88b1b4966650de59cef20c340b28739c52dc9ead91d9959a338a8e531ad38335
6) 92f124ea5217f3fe5cbab1c37a961df0437d5a9cbde1af268c60c4b3194b80ed
7) 781bc4dcbd459893397a8b987bf697f5b95435dfaf7fe3f4d2224728e7a2202a
8) 8862b060db997bc9077e3bece06529c1c116af379985f6138a07ab5fde61b54c
9) d0d7a8f588693b7cc967fb4069419125625eb7454ba553c0416f35fc95307cbe
10) d5d744e0f7984ec01593da35f26bf24e95e4b1cc8bd1c0ff4f31de5dbf94e38f
11) d7333223dcc1002aae04e25e31d8c297efa791a2c1e609d67ac6d9af338efbe8
12) ec3da4ac9ec917e66ab943ab149119807922f64f2e4960ebadc36fe7520b300f

Στο δείγμα 5b1f242aee0eabd4dffea0fe5f08aba60abf7c8d1e4f7fc7357af7f20ccd0204 η ροή των διεργασιών προς το κύριο εκτελέσιμο δεν εμφανίζεται. Ωστόσο ακολουθώντας την ροή από το lolbas explorer.exe βλέπουμε πως οι διεργασίες σταματάνε στο εκτελέσιμο jusched.exe. Η διεργασίες δηλαδή έχουν ως εξής : Smss(488) - Winlogon(588) - Userinit(2664) - explorer.exe(2788) python(4240) - python(2004) – jusched.exe. Το juched είναι μια διεργασία που χρησιμοποιείται για το update τις γλώσσας προγραμματισμού Java.

Στα δείγματα 9eb7abf2228ad28d8b7f571e0495d4a35da40607f04355307077975e2715 53b8 και 180f82bbedb03dc29328e32e054069870a1e65078b78b2120a84c96aaed7d843 τα κύρια εκτελέσιμα τους αντίστοιχα καλούν 4 νέες διεργασίες αντί για μια όπως είδαμε στο δείγμα με sha256 40b865d1c3ab1b8544bcf57c88edd30679870d40b27d62feb237a19f0c5f9cd1. Για παράδειγμα στο δείγμα 9eb7abf2228ad28d8b7f571e0495d4a35da40607f04355307077975e27 1553b8 το κύριο εκτελέσιμο καλεί την bcpePyHGarep.e(1316), την jWkzsfZhIlan.e(1492), την OdifYGjwclan.e(2216) και την icacls.exe(2784). Αρχικά τα αρχεία των εν λόγο διεργασιών κρύβονται από το σύστημα μέσω του API SetFileAttributesW και επίσης κρύβουν το παράθυρο λειτουργίας τους μέσω του API ShellExecuteExW. Παρατηρώντας τις εντολές που δόθηκαν μέσω του command line βλέπουμε ότι χρησιμοποιούνται παράμετροι όπως 8 LAN όπως είδαμε και παραπάνω και 9 REP. Η διεργασία του Ryuk με την παράμετρο στη γραμμή εντολών "9 REP" είναι υπεύθυνη για την αντιγραφή του σε νέους υπολογιστές.

Σε τρία από τα αποτελέσματά μας βλέπουμε ότι το κύριο εκτελέσιμο δεν εμφανίζεται καθόλου. Αντί αυτού βλέπουμε αρκετά ενεργή την διεργασία lsass.exe μέσω των DLL αρχείων που καλεί, αλλά καμία φανερή δραστηριότητα. Τα τρία δείγματα με sha256 είναι :

1) bbbf38de4f40754f235441a8e6a4c8bdb9365dab7f5cfcdac77dbb4d6236360b
2) cfdc2cb47ef3d2396307c487fc3c9fe55b3802b2e570bee9aea4ab1e4ed2ec28
3) cfe1678a7f2b949966d9a020faafb46662584f8a6ac4b72583a21fa858f2a2e8

Η διαγραφή των αρχείων γίνεται κάνοντας χρήση των εκτελέσιμων wmic και vssadmin δηλαδή μέσω των εντολών "cmd /c \"WMIC.exe shadowcopy delet\" και "vssadmin.exe Delete Shadows /all /quiet". Τα υπόλοιπα LOLBAS αρχεία δεν παρουσίασαν κάποια ύποπτη δραστηριότητα.

7.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Ryuk.

1) Allocates read-write-execute memory (usually to unpack itself)
2) Creates known Napolar files, registry keys and/or mutexes
3) Writes a potential ransom message to disk
4) Communicates with host for which no DNS query was performed

5) Performs some HTTP requests
6) Checks for the Locally Unique Identifier on the system for a suspicious privilege
7) Expresses interest in specific running processes
8) File has been identified by 55 AntiVirus engines on VirusTotal as malicious
9) Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)
10) Generates some ICMP traffic

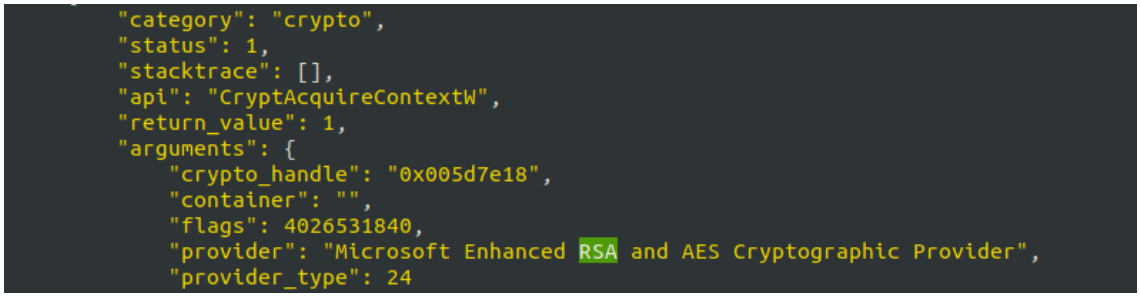
7.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping
3) The executable contains unknown PE section names indicative of a packer (could be a false positive)
4) One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.
5) Creates hidden or system file
6) Checks for the presence of known devices from debuggers and forensic tools
7) Checks for the presence of known windows from debuggers and forensic tools
8) Detects VirtualBox through the presence of a device
9) A process created a hidden window
10) One or more of the buffers contains an embedded PE file
11) Changes read-write memory protection to read-execute (probably to avoid detection when setting all RWX flags at the same time)
12) One or more thread handles in other processes
13) One or more processes crashed
14) Enumerates services, possibly for anti-virtualization
15) Allocates execute permission to another process indicative of possible code injection
16) Creates a thread using CreateRemoteThread in a non-child process indicative of process injection
17) Resumed a suspended thread in a remote process potentially indicative of process injection
18) Stopped Application Layer Gateway service
19) Kernel module without a name
20) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
21) Malfind detects one or more injected processes
22) Stopped Firewall service
23) Checks if process is being debugged by a debugger
24) Manipulates memory of a non-child process indicative of process injection

7.4 Κρυπτογράφηση

Σε αρκετά αποτελέσματα εντοπίσαμε πως γίνεται η χρήση RSA και AES κρυπτογράφησης όπως φαίνεται παρακάτω στην εικόνα από το δείγμα 9eb7abf2228ad28d8 b7f571e0495d4a35da40607f04355307077975e271553b8.

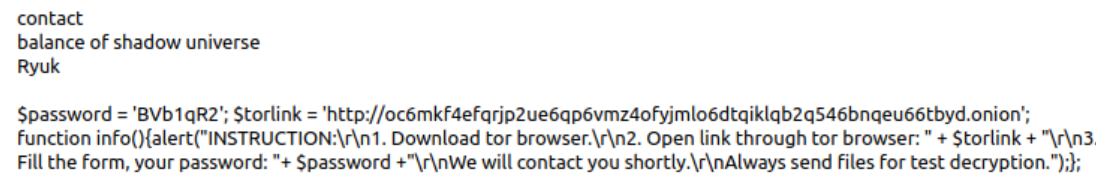


Εικόνα 86: Ryuk Encryption

Τα αποτελέσματα των αναλύσεων δεν παρουσιάζουν ακριβή στοιχεία κρυπτογράφησης ωστόσο από θεωρητικές πηγές εντοπίσαμε πως το συγκεκριμένο ransomware χρησιμοποιεί έναν συνδυασμό αλγορίθμων κρυπτογράφησης, συμπεριλαμβανομένου ενός συμμετρικού αλγορίθμου (AES-256) και ενός ασύμμετρου (RSA 2048 ή 4096) [25] όπου τους χρησιμεύει τόσο για την κρυπτογράφηση των αρχείων όσο και για την προστασία του κλειδιού κρυπτογράφησης, καθιστώντας αδύνατη την αποκρυπτογράφηση των δεδομένων από τρίτους. Παρακάτω φαίνεται το μήνυμα πληρωμής λύτρων όπως φαίνεται στο buffer των Json files.

Εικόνα 85: Ryuk Ransom Message in Json

Εδώ φαίνεται το μήνυμα όπως εμφανίζεται μέσα σε ένα εν λειτουργία Sandbox μηχάνημα.



Εικόνα 87: Ryuk Ransom Message

8 RagnarLocker

8.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του RagnarLocker με hash sha256 :

1) 3b43751ed88e4d1f82cf52ca2d4477e3e35c35f08c1b4e3ab21c80720601e804
2) 04c9cc0d1577d5ee54a4e2d4dd12f17011d13703cdd0e6efd46718d14fd9aa87
3) 10f9ad4e9f6e0dc1793be80203b258f8c5114d01cb17307c1b2fdcca37d4edf9
4) 1602d04000a8c7221ed0d97d79f3157303e209d4640d31b8566dd52c2b09d033
5) 9706a97ffa43a0258571def8912dc2b8bf1ee207676052ad1b9c16ca9953fc2c
6) 5469182495d92a5718e0e1dcdf371e92b79724e427050154f318de693d341c89
7) dd79b2abc21e766fe3076038482ded43e5069a1af9e0ad29e06dce387bfae900

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις του RagnarLocker με hash sha256 :

1) 3bc8ce79ee7043c9ad70698e3fc2013806244dc5112c8c8d465e96757b57b1e1
2) 7af61ce420051640c50b0e73e718dd8c55dddfcb58917a3bead9d3ece2f3e929
3) 9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376
4) 30dcc7a8ae98e52ee5547379048ca1fc90925e09a2a81c055021ba225c1d064c
5) 041fd213326dd5c10a16caf88ff076bb98c68c052284430fba5f601023d39a14
6) 68eb2d2d7866775d6bf106a914281491d23769a9eda88fc078328150b8432bb3
7) 63096f288f49b25d50f4aea52dc1fc00871b3927fa2a81fa0b0d752b261a3059
8) b670441066ff868d06c682e5167b9dbc85b5323f3acfbbc044cabc0e5a594186
9) c2bd70495630ed8279de0713a010e5e55f3da29323b59ef71401b12942ba52f6
10) cf5ec678a2f836f859eb983eb633d529c25771b3b7505e74aa695b7ca00f9fa8
11) dd5d4cf9422b6e4514d49a3ec542cffb682be8a24079010cda689afbb44ac0f4
12) e1957024039b0e48a15c27448f19d4df4f0e4666f9ac34e7f4d42dd3c32e15ed
13) ec35c76ad2c8192f09c02eca1f263b406163470ca8438d054db7adcf5bfc0597

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Wmic.exe				

Λειτουργικότητα και Ροή Διεργασιών

Η ροή των διεργασιών στο ενδιαφέρον δείγμα με sha256 3bc8ce79ee7043c9ad70698e3fc2013806244dc5112c8c8d465e96757b57b1e1 έχει ως εξής : Smss(488) → Winlogon(588) → Userinit(2664) → explorer.exe(2788) python(4240) → python(2004) → inject-x86.exe(2632) → Διεργασία Κύριου Εκτελέσιμου("3bc8ce79ee7043")(2832) → Wmic(1332) και Vssadmin(4448). Οι Wmic και οι Vssadmin τελειώνοντας καλούν ξεχωριστά από μια διεργασία conhost.exe για την εκτέλεση εντολών στο Command Line. Στα 7 πρώτα δείγματα που αναφέρονται χωρίς να φαίνεται κάποια χρήση του Wmic η ροή σταματάει στο κύριο εκτελέσιμο. Στα δείγματα που έχουν στον πίνακα των LOLBAS το Wmic.exe η ροή των διεργασιών είναι όπως του δείγματος που μόλις αναλύσαμε. Συνεχίζοντας βλέπουμε την λειτουργικότητα του Wmic και του vssadmin που έχουμε

ξαναδεί και η δράση τους είναι γνωστή δηλαδή η διαγραφή των αντιγράφων ασφαλείας μας μέσω των εντολών :

"command_line": "wmic.exe shadowcopy delete"

"commandline": "vssadmin delete shadows /all /quiet",

Ενδιαφέρουσες λειτουργίες ακόμα αποτελούν η χρήση geolocation για τον γεωγραφικό εντοπισμό της θέσης του συστήματος ώστε αν ανήκει σε συγκεκριμένες φιλικές χώρες να μην εκτελεστεί η λειτουργία της κρυπτογράφησης. Η λειτουργία αυτή ελέγχεται από το GetLocaleInfoW.

8.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του RagnarLocker.

1) Queries for the computername
2) Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
3) Allocates read-write-execute memory (usually to unpack itself)
4) Communicates with host for which no DNS query was performed
5) Performs some HTTP requests
6) Attempts to stop active services
7) Repeatedly searches for a not-found process, you may want to run a web browser during analysis
8) Terminates another process
9) The executable is compressed using UPX
10) Expresses interest in specific running processes
11) File has been identified by 55 AntiVirus engines on VirusTotal as malicious

8.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping
3) The executable contains unknown PE section names indicative of a packer (could be a false positive)
4) The executable uses a known packer
5) The executable is likely packed with VMProtect
6) Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation
7) One or more thread handles in other processes
8) Potentially malicious URLs were found in the process memory dump
9) Enumerates services, possibly for anti-virtualization
10) Resumed a suspended thread in a remote process potentially indicative of process injection
11) Kernel module without a name

12) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
13) Malfind detects one or more injected processes
14) Stopped Firewall service
15) Stopped Application Layer Gateway service
16) Checks if process is being debugged by a debugger

8.4 Κρυπτογράφηση

Πριν ξεκινήσει η διαδικασία της κρυπτογράφησης το RagnarLocker όπως και τα περισσότερα Ransomware τερματίζουν μια λίστα από διεργασίες που περιλαμβάνουν συμβολοσειρές που σχετίζονται με λύσεις δημιουργίας αντιγράφων ασφαλείας και προστασίας από ιούς (όπως «sophos» και «veeam»), καθώς και εργαλεία λογισμικού απομακρυσμένης διαχείρισης. Μετά αφού εντοπιστούν τα σημαντικά αρχεία γίνεται η χρήση RSA-2048 για την κρυπτογράφηση των κλειδιών των φακέλων και Salsa20 κρυπτογράφηση για τους φακέλους [26]. Το Ragnar Locker δημιουργεί τυχαία τις επεκτάσεις αρχείων ανά χρήστη ανακτώντας την τιμή του ονόματος υπολογιστή και καταλήγοντας παρουσιάζει το μήνυμα πληρωμής λύτρων.

```
buffer:
**************************************************************************************************************
HELLO Omniga.de ! If you reading this message, then your network was PENETRATED and all of your files and data
has been ENCRYPTED by RAGNAR_LOCKER !
**************************************************************************************************************
!!!!! WARNING !!!!! DO NOT Modify, rename, copy or move any files or you can DAMAGE them and decryption will be
impossible. DO NOT use any third party or public decryption software, it also may damage files. DO NOT Shutdown
or reset your system ----------------------------------- There is ONLY ONE possible way to get back your files
- contact us and pay for our special decryption key ! For your GUARANTEE we will decrypt 2 of your files FOR
FREE, as a proof of our capabilities Don't waste your TIME, the link for contacting us will be deleted if there
is no contact made in closest future and you will never restore your DATA. HOWEVER if you will contact us within
2 day since get penetrated - you can get a very SPECIAL PRICE. WARNING ! We had downloaded your private
information including billing info, clients private data, contracts, agreements and a lot of other sensitive
information. Also we take your SQL server DB,access to leadership's mails and correspondence, admin credentials,
VPN-servers, Backup shares, Cloud host, and a lot of other info from your Network. You can check some proofs
here: https://prnt.sc/s5g6gr https://prnt.sc/s5g79t https://prnt.sc/s5gkxh Whole data gathered from your SECRET
files and directories could be published for everyone's view and your partners, clients and investors would be
notified about leak. However if we make a deal everything would be kept in secret and all your data will be
restored. You can take a look on some examples of what we have, right now it's a private hidden page. Use Tor
Browser to open the link: http://p6o7m73ujalhgkiv.onion/temporary-de-page-424/ to view the page's content use
password: OmnigUdk$912f If you wouldn't PAY, we will publish this post with much more information available for
Downloading. In mass media breaking news about the leak will make a lot of noise in every IT-journals, blogs,
sites etc. Besides, your private data will be sold out on Darknet forums! To avoid such troubles and lawsuits
from your clients and partners it's better to make a deal with us.
========================================================================================================== !
HERE IS THE SIMPLE MANUAL HOW TO GET CONTACT WITH US VIA LIVE CHAT !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! a) Download and install TOR browser from
this site : https://torproject.org b) For contact us via LIVE CHAT open our website :
http://stppd5as5x4hxs45.onion/client/?0dFE0B7BAA7C7801ddd746B1DC5ad44bAD82Fc0f77DAC01bD3cf3D2D9deB94bC c) For
visit our NEWS PORTAL with your data, open this website : http://p6o7m73ujalhgkiv.onion/temporary-de-page-424/ (
password: OmnigUdk$912f ) d) If Tor is restricted in your area, use VPN When you open LIVE CHAT website follow
rules : Follow the instructions on the website. At the top you will find CHAT tab. Send your message there and
wait for response (we are not online 24/7, So you have to wait for your turn).
offset: 0
file_handle: 0x00000124
filepath: C:\Users\Public\Documents\RGNR_13B82609.txt
```

Εικόνα 88: RagnarLocker Ransom Message

9 MountLocker

9.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του MountLocker με hash sha256 :

1) 0aa8099c5a65062ba4baec8274e1a0650ff36e757a91312e1755fded50a79d47
2) 00ed4c347cd62526226363a0aceb851b2ef7e3a4da78433a28f2cd6cbd5f1b99
3) 4a5ac3c6f8383cc33c795804ba5f7f5553c029bbb4a6d28f1e4d8fb5107902c1
4) 5eae13527d4e39059025c3e56dad966cf67476fe7830090e40c14d0a4046adf0
5) 8b539f3ba05fe82c4f992ffbeb6ab55151b36dce2d03b64721e966dedf82be81
6) 226a723ffb4a91d9950a8b266167c5b354ab0db1dc225578494917fe53867ef2
7) 187610fb06cc60c73c0062b593c5fe3ba29e0436e396969feb9ed25391ff7e8b
8) b26749b17ca691328ba67ee49d4d9997c101966c607ab578afad204459b7bf8f
9) e7c277aae66085f1e0c4789fe51cac50e3ea86d79c8a242ffc066ed0b0548037
10) e435a95489a4ebdfdc12031091f92a7f9c5e3f6cc9b55355ee4030d82553e9ac
11) f570d5b17671e6f3e56eae6ad87be3a6bbfac46c677e478618afd9f59bf35963

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις με hash sha256 :

1) 2c44444d207a78da7477ae1af195d4265134e895bebb476f7b2c003f1467a033
2) 2d2d2e39ccae1ff764e6618b5d7636d41ac6e752ce56d69a9acbb9cb1c8183d0
3) 7fe1686f4afb9907f880a5e77bf30bc00fae71980f57ca70b60b7b1716456a2f
4) 6143d920ebdd5e9b1db7425916417c0896139f425493a8fcd63d62dac80779f1
5) 31630d16f4564c7a214a206a58f60b7623cd1b3abb823d10ed50aa077ca33585

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Rundll32.exe				

Στην ανάλυση με hash sha256:

1) f570d5b17671e6f3e56eae6ad87be3a6bbfac46c677e478618afd9f59bf35963

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Msiexec.exe				

Στην ανάλυση με hash sha256:

1) 30050b3673c720729cd6a61803059b16dd3aa526683e7342aae0261e4c78fa83

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Msiexec.exe	Zipfldr.dll			

Από διάφορες έρευνες που διεξήγαγαν εταιρίες με ειδίκευση στον τομέα της κυβερνοασφάλειας εντοπίσαμε πως το Ransomware Mountlocker μετονομάστηκε σε Astralocker. Αυτό φαίνεται επίσης και από τις ημερομηνίες που ανέβηκαν τα δείγματα στο Malware bazaar καθώς το τελευταίο δείγμα του Mountlocker έχει ανέβει στις 09/01/2022 και το πρώτο του Astralocker στις 11/04/2022. Συνεπώς θα συμπεριλάβουμε στην έρευνα και τρία δείγματα από το AstraLocker. Τα δείγματα είναι τα :

1) cf3bdf0f8ea4c8ece5f5a76524ab4c81fea6c3a1715b5a86b3ad4d397fca76f3
2) b0a010e5a9b353a11fb664501de91fc47878d89bf97cb57bc03428c7a45981b9
3) 17ea24ce8866da7ef4a842cba16961eafba89d526d3efe5d783bb7a30c5d1565

Στα αποτελέσματα των δύο πρώτων δειγμάτων γίνεται χρήση των παρακάτω LOLBAS.

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll	Cmd.exe			

Στα αποτελέσματα του τρίτου δείγματος γίνεται χρήση των παρακάτω LOLBAS.

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll				

Λειτουργικότητα και Ροή Διεργασιών

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 00ed4c347cd62526226363a0aceb851b2ef7e3a4da78433a28f2cd6cbd5f1b99
2) 5eae13527d4e39059025c3e56dad966cf67476fe7830090e40c14d0a4046adf0
3) 8b539f3ba05fe82c4f992ffbeb6ab55151b36dce2d03b64721e966dedf82be81
4) 31630d16f4564c7a214a206a58f60b7623cd1b3abb823d10ed50aa077ca33585
5) 187610fb06cc60c73c0062b593c5fe3ba29e0436e396969feb9ed25391ff7e8b
6) e7c277aae66085f1e0c4789fe51cac50e3ea86d79c8a242ffc066ed0b0548037
7) e435a95489a4ebdfdc12031091f92a7f9c5e3f6cc9b55355ee4030d82553e9ac
8) b0a010e5a9b353a11fb664501de91fc47878d89bf97cb57bc03428c7a45981b9
9) cf3bdf0f8ea4c8ece5f5a76524ab4c81fea6c3a1715b5a86b3ad4d397fca76f3

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 0aa8099c5a65062ba4baec8274e1a0650ff36e757a91312e1755fded50a79d47
2) 4a5ac3c6f8383cc33c795804ba5f7f5553c029bbb4a6d28f1e4d8fb5107902c1
3) f570d5b17671e6f3e56eae6ad87be3a6bbfac46c677e478618afd9f59bf35963

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x64.exe → Κύριο Εκτελέσιμο

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) b26749b17ca691328ba67ee49d4d9997c101966c607ab578afad204459b7bf8f

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → Κύριο Εκτελέσιμο

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 2d2d2e39ccae1ff764e6618b5d7636d41ac6e752ce56d69a9acbb9cb1c8183d0
2) 2c44444d207a78da7477ae1af195d4265134e895bebb476f7b2c003f1467a033
3) 7fe1686f4afb9907f880a5e77bf30bc00fae71980f57ca70b60b7b1716456a2f
4) 6143d920ebdd5e9b1db7425916417c0896139f425493a8fcd63d62dac80779f1

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → rundll32.exe → rundll32.exe → Κύριο Εκτελέσιμο

Η ροή των διεργασιών στο δείγμα με hash sha256 :

1) 226a723ffb4a91d9950a8b266167c5b354ab0db1dc225578494917fe53867ef2

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python→ python → inject-x86.exe → Κύριο Εκτελέσιμο → PowerShell.exe

Η ροή των διεργασιών στο δείγμα με hash sha256 :

1) 17ea24ce8866da7ef4a842cba16961eafba89d526d3efe5d783bb7a30c5d1565

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → jusched.exe

Η ροή των διεργασιών στο δείγμα με hash sha256 :

1) 30050b3673c720729cd6a61803059b16dd3aa526683e7342aae0261e4c78fa83

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → msiexec.exe

Ξεκινώντας από το δείγμα 0aa8099c5a65062ba4baec8274e1a0650ff36e757a91312e1755fded50a79d47 παρατηρείται ότι γίνεται χρήση της εντολής vssadmin.exe delete shadows /all /Quiet" για την διαγραφή των αντιγράφων ασφαλείας. Στα δείγματα με μπλε χρωματισμό παραπάνω εμφανίζεται μια συνεχής χρήση του LOLBAS rundll32.exe καθώς εκτελεί το DLL αρχείο που δημιουργήθηκε από το κύριο εκτελέσιμο για μια πιο κρυφή λειτουργία των μηχανισμών του. Στο δείγμα 226a723ffb4a91d9950a8b266167c5b354ab0db1dc225578494917fe53867ef2 χρησιμοποιείται το κύριο εκτελέσιμο καθώς και το powershell.exe v1.0 όπου και βλέπουμε πολλές λεπτομέρειες σχετικά με την χρήση τους, πολλά καλέσματα σε API καθώς και το μήνυμα πληρωμής λύτρων. Επιπρόσθετα σημαντικό είναι ότι εντοπίζει την γλώσσα που χρησιμοποιεί το σύστημα για την ανίχνευση φιλικής χώρας και αποφυγής εκτέλεσης του ransomware.

9.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του MountLocker.

1) Allocates read-write-execute memory (usually to unpack itself)
2) Queries for the computername
3) Communicates with host for which no DNS query was performed
4) Performs some HTTP requests

5) Checks for the Locally Unique Identifier on the system for a suspicious privilege
6) Uses Windows APIs to generate a cryptographic key
7) File has been identified by 55 AntiVirus engines on VirusTotal as malicious
8) Found URLs related to Tor in process memory dump (e.g. onion services, Tor2Web, and Ransomware)
9) Creates a suspicious process
10) Terminates another process
11) This executable has a PDB path
12) Creates executable files on the filesystem
13) Uses suspicious command line tools or Windows utilities
14) Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
15) Writes a potential ransom message to disk

9.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) The executable contains unknown PE section names indicative of a packer (could be a false positive)
3) Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available
4) Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation
5) The file contains an unknown PE resource name possibly indicative of a packer
6) Changes read-write memory protection to read-execute (probably to avoid detection when setting all RWX flags at the same time)
7) One or more thread handles in other processes
8) One or more processes crashed
9) Potentially malicious URLs were found in the process memory dump
10) Detects VMWare through the in instruction feature
11) Resumed a suspended thread in a remote process potentially indicative of process injection
12) A process created a hidden window
13) Stopped Application Layer Gateway service
14) Kernel module without a name
15) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
16) Malfind detects one or more injected processes
17) Stopped Firewall service
18) Checks if process is being debugged by a debugger
19) Looks for the Windows Idle Time to determine the uptime

9.4 Κρυπτογράφηση

Η διαδικασία της κρυπτογράφησης στο Mountlocker προαπαιτεί ορισμένες διαδικασίες πριν το ξεκίνημά της. Όπως και στα περισσότερα Ransomware αρχικά πρέπει να ολοκληρωθεί η ανίχνευση πληροφοριών του συστήματος και του δικτύου. Παρακάτω φαίνεται ένα μέρος από το πεδίο strings του δείγματος 00ed4c347cd62526226363a0aceb851b2ef7e3a4da78433a28f2cd6cbd5f1b99 που βλέπουμε τι πληροφορίες εντοπίζονται και τι διεργασίες σταματάνε.

```
"========== SYS INFO ==========",
"CORE COUNT:",
"TOTAL MEM:",
"WIN VER:",
"%u.%u.%u SP%u",
"WIN ARCH:",
"USER NAME:",
"PC NAME:",
"IN DOMAIN:",
"IS ADMIN:",
"CMDLINE:",
"================================",
"          KILL PROCESS          ",
"================================",
"[ERROR] locekr.kill.process > get process list error=%s",
"timeout",
"          KILL SERVICE          ",
"[ERROR] locekr.kill.service > get services list error=%s",
"/NOLOCK=",
"/TARGET=",
```

Εικόνα 90: System Info and Process Kill

Για την κρυπτογράφηση των αρχείων και των κλειδιών το MountLocker χρησιμοποιεί υβριδική κρυπτογραφία με την αξιοποίηση του RSA-2048 και ChaCha20 [27]. Τέλος βλέπουμε και το μήνυμα πληρωμής λύτρων όπως φαίνεται στο JSON αρχείο.

```
        "/!\\ YOUR COMPANY' NETWORK HAS BEEN HACKED /!\\ <br>",
        "All your important documents have been encrypted and transferred to our premises!</b><br>",
        "ANY ATTEMPT TO RESTORE YOUR FILES WITH THIRD-PARTY SOFTWARE WILL PERMANENTLY CORRUPT IT.<br>",
        "DO NOT MODIFY ENCRYPTED FILES.<br>",
        "DO NOT RENAME ENCRYPTED FILES.<br>",
        "However there is a solution for your problem!<br>",
        "We can support you with your data decryption for a monetary reward.<br>",
        "Also we will destroy your private data from our premises.<br>",
        "And we can prove our decryption capabilities by decrypting couple of your files free of charge.<br>",
        "Here are the next steps to get your valuable data back and get it wiped out from our premises:<br>",
        "<a href=\"http://br3o5we2252csfnhotfbsfx7ch5csivuuidhdefbhmg2zmbqebs6znad.onion/?
cid=%CLIENT_ID%                                                \">http://
br3o5we2252csfnhotfbsfx7ch5csivuuidhdefbhmg2zmbqebs6znad.onion/?
cid=%CLIENT_ID%                                                </a><br>",
        "* Note that you need installed Tor Browser to open this kind of links.<br><br>",
        "Follow the instructions to install/run Tor Browser:<br>",
        "1. Go to TOR Project website https://www.torproject.org using your usual browser (Chrome, Firefox,
Internet Explorer or Edge)<br>",
        "2. Click \"Download Tor Browser\" and pick right version for your Operation System (this is Windows
in 99.9% of cases)<br>",
        "3. Download and Install Tor Browser<br>",
        "4. After installation finished, click on \"Tor Browser\" link from your Desktop<br>",
        "5. By using Tor Browser visit <a href=\"http://
br3o5we2252csfnhotfbsfx7ch5csivuuidhdefbhmg2zmbqebs6znad.onion/?
cid=%CLIENT_ID%                                                \">http://
br3o5we2252csfnhotfbsfx7ch5csivuuidhdefbhmg2zmbqebs6znad.onion/?
cid=%CLIENT_ID%                                                </a><br>",
        "6. Copy your client id from the top of this document and paste it into Authorization window if
requested <br>",
        "7. This will start a chat with our security experts.<br>",
        "<b>If you can`t use the link above, please, use the email:</b><br>",
        "<a href=\"mailto:mountman@tutanota.com\">mountman@tutanota.com</a><br>",
        "Please note, sometimes our team is away from keyboard, but make sure they will reply you back as soon
as possible.<br>",
        " Also, we kindly request you to contact with us as soon as possible.<br>",
        " We will start publishing your private data to the Internet if you don't get in touch with us within
next few days.<br>",
```

Εικόνα 89: MountLocker Ransom Message

10 BlackMatter

10.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις του BlackMatter με hash sha256 :

1) 2aad85dbd4c79bd21c6218892552d5c9fb216293a251559ba59d45d56a01437c
2) 2e50eb85f6e271001e69c5733af95c34728893145766066c5ff8708dcc0e43b2
3) 5da8d2e1b36be0d661d276ea6523760dbe3fa4f3fdb7e32b144812ce50c483fa
4) 6d4712df42ad0982041ef0e2e109ab5718b43830f2966bd9207a7fac3af883db
5) 7f6dd0ca03f04b64024e86a72a6d7cfab6abccc2173b85896fc4b431990a5984
6) 8eada5114fbbc73b7d648b38623fc206367c94c0e76cb3b395a33ea8859d2952
7) 9bae897c19f237c22b6bdc024df27455e739be24bed07ef0d409f2df87eeda58
8) 22d7d67c3af10b1a37f277ebabe2d1eb4fd25afbd6437d4377400e148bcc08d6
9) 520bd9ed608c668810971dbd51184c6a29819674280b018dc4027bc38fc42e57
10) 730f2d6243055c786d737bae0665267b962c64f57132e9ab401d6e7625c3d0a4
11) b824bbc645f15e213b4cb2628f7d383e9e37282059b03f6fe60f7c84ea1fed1f
12) c6e2ef30a86baa670590bd21acf5b91822117e0cbe6060060bc5fe0182dace99
13) daed41395ba663bef2c52e3d1723ac46253a9008b582bb8d9da9cb0044991720
14) e4fd947a781611c85ea2e5afa51b186de7f351026c28eb067ad70028acd72cda

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL

Στις αναλύσεις με hash sha256 :

1) 4e74b6733558644ee27e4c568bec821c8e2cec95c86f524999edbbbbe932a43e
2) 70344ece62a828c46ff315b3328125d8ab5f6902bbeaa24224fee97142ee6ad9

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Rundll32.exe				

Στις αναλύσεις με hash sha256 :

1) a56b41a6023f828cccaaef470874571d169fdb8f683a75edd430fbd31a2c3f6e
2) d61af007f6c792b8fb6c677143b7d0e2533394e28c50737588e40da475c040ee
3) c690148b6baec765c65fe91ea9f282d6a411ae90c08d74d600515b3e075e21b2

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll				

Στις αναλύσεις με hash sha256 :

1) 3609272795c8f8ba1275959d1457b03f6143efaaf8cd037547cd561e68763237

Χρησιμοποιούνται τα παρακάτω LOLBAS όπου με μπλε χρωματισμό τα νέα :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Zipfldr.dll	Cmd.exe			

Λειτουργικότητα και Ροή Διεργασιών

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 2aad85dbd4c79bd21c6218892552d5c9fb216293a251559ba59d45d56a01437c
2) 4e74b6733558644ee27e4c568bec821c8e2cec95c86f524999edbbbbe932a43e
3) 8eada5114fbbc73b7d648b38623fc206367c94c0e76cb3b395a33ea8859d2952
4) 730f2d6243055c786d737bae0665267b962c64f57132e9ab401d6e7625c3d0a4
5) 70344ece62a828c46ff315b3328125d8ab5f6902bbeaa24224fee97142ee6ad9
6) daed41395ba663bef2c52e3d1723ac46253a9008b582bb8d9da9cb0044991720
7) 3609272795c8f8ba1275959d1457b03f6143efaaf8cd037547cd561e68763237
8) c690148b6baec765c65fe91ea9f282d6a411ae90c08d74d600515b3e075e21b2

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 2e50eb85f6e271001e69c5733af95c34728893145766066c5ff8708dcc0e43b2
2) 5da8d2e1b36be0d661d276ea6523760dbe3fa4f3fdb7e32b144812ce50c483fa
3) 6d4712df42ad0982041ef0e2e109ab5718b43830f2966bd9207a7fac3af883db
4) 7f6dd0ca03f04b64024e86a72a6d7cfab6abccc2173b85896fc4b431990a5984
5) 9bae897c19f237c22b6bdc024df27455e739be24bed07ef0d409f2df87eeda58
6) 22d7d67c3af10b1a37f277ebabe2d1eb4fd25afbd6437d4377400e148bcc08d6
7) 520bd9ed608c668810971dbd51184c6a29819674280b018dc4027bc38fc42e57
8) b824bbc645f15e213b4cb2628f7d383e9e37282059b03f6fe60f7c84ea1fed1f
9) c6e2ef30a86baa670590bd21acf5b91822117e0cbe6060060bc5fe0182dace99
10) e4fd947a781611c85ea2e5afa51b186de7f351026c28eb067ad70028acd72cda
11) a56b41a6023f828cccaaef470874571d169fdb8f683a75edd430fbd31a2c3f6e
12) d61af007f6c792b8fb6c677143b7d0e2533394e28c50737588e40da475c040ee

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → Κύριο Εκτελέσιμο

Ενδιαφέρον σημεία παρουσιάζονται στο δείγμα 5da8d2e1b36be0d661d276ea6523760dbe3fa4f3fdb7e32b144812ce50c483fa όπου βλέπουμε την απόκτηση πολλαπλών δικαιωμάτων στο σύστημα από το BlackMatter. Αρχικά γίνεται privilege escalation μέσω της παραποίησης του δικαιώματος δημιουργίας token SeCreateTokenPrivilege από την διεργασία lsass.exe. Η παραποίηση φαίνεται από τα attributes που υφίστανται μόνο το Present και το Enabled και όχι το Default.


```
75094                {
75095                    "description": "Create a token object",
75096                    "value": 2,
75097                    "filename": "lsass.exe",
75098                    "process_id": 612,
75099                    "privilege": "SeCreateTokenPrivilege",
75100                    "attributes": "Present,Enabled"
75101                },
```


Εικόνα 91: SeTokenPriviledge

Από το πεδίο Privs data βλέπουμε το κύριο εκτελέσιμο να αποκτά προνόμια όπως :

- Increase quotas
- Manage auditing and security log

- Take ownership of files/objects
- Load and unload device drivers
- Profile system performance
- Profile a single process
- Increase scheduling priority
- Backup files and directories
- Restore files and directories
- Shut down the system
- Debug programs
- Manage the files on a volume
- Allocate more memory for user applications

Σημαντικό επίσης είναι ότι το ransomware σταματάει την διεργασία Internet Connection Sharing ώστε να μην γίνει δυνατή η ανάκτηση αρχείων από κάποιο backup. Εφαρμόζονται τεχνικές για persistence, lateral movement και ανάκτησης κωδικών των χρηστών. Επιπρόσθετα γίνεται εντοπισμός πληροφοριών του συστήματος και δεδομένων web του χρήστη καθώς στην συνέχεια τα στέλνει στον επιτιθέμενο μέσω http ή https. Τέλος γίνεται η κρυπτογράφηση και διαγραφή των backups.

Πολλές πληροφορίες στο διαδίκτυο δείχνουν πως το BlackMatter Ransomware είναι η συνέχεια των Revil και DarkSide καθώς ισχυρίζεται ότι χρησιμοποιεί τα καλύτερα χαρακτηριστικά και των δύο Ransomware. Από την ανάλυσή μας φαίνεται πως το BlackMatter χρησιμοποιεί πολύ καλή τεχνολογία για την αποφυγή ανάλυσης από εικονικές μηχανές καθώς δεν εκδηλώνεται πλήρως. Επιπρόσθετα έχει την ικανότητα τα προσβάλει συστήματα Windows αλλά και Linux σαφώς με διαφορετικό δείγμα.

10.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Ryuk.

1) Allocates read-write-execute memory (usually to unpack itself)
2) Queries for the computername
3) Communicates with host for which no DNS query was performed
4) Performs some HTTP requests
5) HTTP traffic contains suspicious features which may be indicative of malware related traffic
6) Sends data using the HTTP POST Method
7) Creates a shortcut to an executable file
8) Checks adapter addresses which can be used to detect virtual network interfaces
9) File has been identified by 55 AntiVirus engines on VirusTotal as malicious
10) Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)
11) Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)

10.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

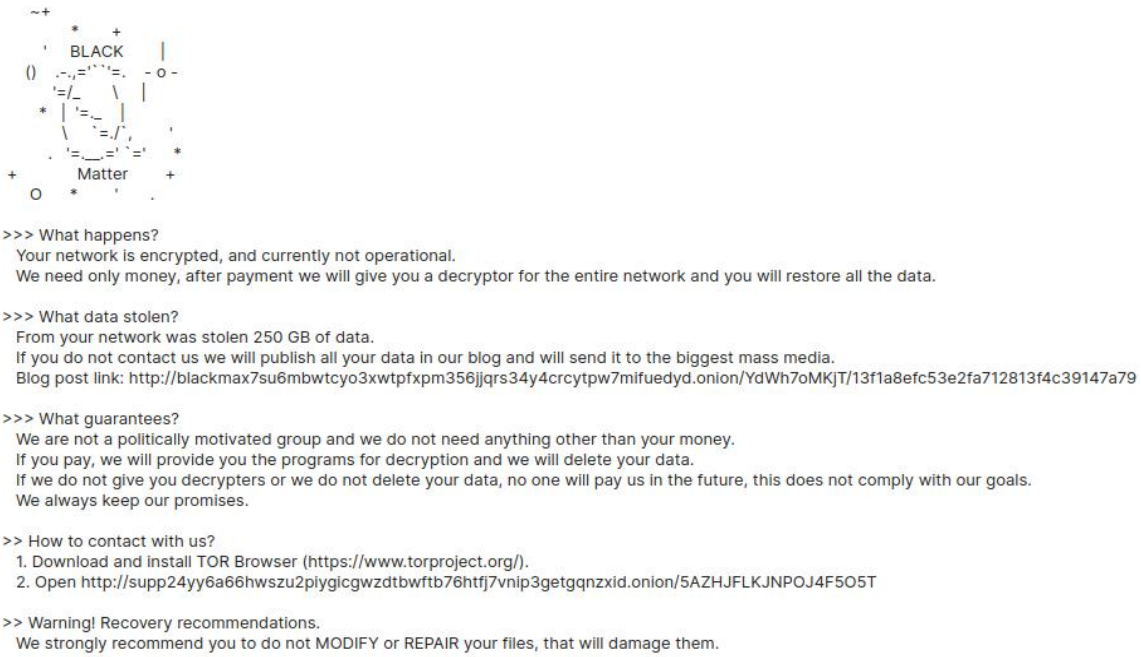
Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer

2) The executable contains unknown PE section names indicative of a packer (could be a false positive)
3) Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation
4) One or more thread handles in other processes
5) One or more processes crashed
6) Potentially malicious URLs were found in the process memory dump
7) One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.
8) Stopped Application Layer Gateway service
9) Kernel module without a name
10) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
11) Malfind detects one or more injected processes
12) Stopped Firewall service

10.4 Κρυπτογράφηση

Το BlackMatter όπως και τα υπόλοιπα Ransomware εκτελεί πρώτα τις διαδικασίες της ανίχνευσης πληροφοριών. Για την κρυπτογράφηση των αρχείων το BlackMatter χρησιμοποιεί τον αλγόριθμο Salsa20. Επιπρόσθετα γίνεται χρήση του RSA-1024 για την προστασία των κλειδιών του Salsa20 δείχνοντας την εφαρμογή της υβριδικής κρυπτογραφίας [28]. Το BlackMatter χρησιμοποιεί μερική κρυπτογράφηση, δηλαδή δεν κρυπτογραφεί ένα αρχείο ολόκληρο αλλά ένα μέρος αυτού για να συντομέψει την διαδικασία της επίθεσης. Τέλος βλέπουμε και το μήνυμα πληρωμής λύτρων.



Εικόνα 92: BlackMatter Ransom Message

11 Egregor

11.1 Living Off the Land Binaries, Scripts and Libraries

Στις αναλύσεις με hash sha256 :

1) 3aad14d200887119f316be71d71aec11735dd3698a4fcaa50902fce71bdccb07
2) 3ae02fc1fdb653997eeb9303305f1ec35dbb87eb603b573bd94895f35542f1a8
3) 3b13b6f1d7cd14dc4a097a12e2e505c0a4cff495262261e2bfc991df238b9b04
4) 3e5a6834cf6192a987ca9b0b4c8cb9202660e399ebe387af8c7407b12ae2da63
5) 004a2dc3ec7b98fa7fe6ae9c23a8b051ec30bcfcd2bc387c440c07ff5180fe9a
6) 6a441734b34cdee31a01164140b0c88966fbb4358dcb63a14ae6824f09e9476f
7) 6ad7b3e0873c9ff122c32006fdc3675706a03c4778287085a020d839b74cd780
8) 14da004cc96b910fb75abb86df09e318d92f4fb8dda39c8bd6a8e0601b6605d8
9) 14e547bebaa738b8605ba4182c4379317d121e268f846c0ed3da171375e65fe4
10) 30c18908c6f9b545dafa30edfc24f5fbd808ed69343f701c1f8d6501fe83cbdf
11) 9017c070ad6ac9ac52e361286b3ff24a315f721f488b53b7aaf6ac35de477f44
12) 765327e1dc0888c69c92203d90037c5154db9787f54d3fc8f1097830be8c76ab
13) 6713403015feb8959093f5d007bcbdbb3be9eec96dd62f517786b67506067251
14) 932778732711cd18d5c4aabc507a65180bf1d4bd2b7d2d4e5506be4b8193596e
15) a9d483c0f021b72a94324562068d8164f8cce0aa8f779faea304669390775436
16) a5989c480ec6506247325652a1f3cb415934675de3877270ae0f65edd9b14d13
17) af538ab1b8bdfbf5b7f1548d72c0d042eb14d0011d796cab266f0671720abb4d
18) c1c4e677b36a2ee6ae858546e727e73cc38c95c9024c724f939178b3c03de906
19) c3c50adcc0a5cd2b39677f17fb5f2efca52cc4e47ccd2cdbbf38815d426be9e1
20) ee06c557f1acd5c4948b1df0413e49f3885f8ac96185a9d986b91a1231444541

Χρησιμοποιούνται τα παρακάτω LOLBAS :

LOLBAS				
Explorer.exe	Dllhost.exe	Shell32.dll	Setupapi.dll	Shdocvw.dlL
Rundll32.exe				

Λειτουργικότητα και Ροή Διεργασιών

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 3b13b6f1d7cd14dc4a097a12e2e505c0a4cff495262261e2bfc991df238b9b04
2) 3e5a6834cf6192a987ca9b0b4c8cb9202660e399ebe387af8c7407b12ae2da63
3) 004a2dc3ec7b98fa7fe6ae9c23a8b051ec30bcfcd2bc387c440c07ff5180fe9a
4) 6a441734b34cdee31a01164140b0c88966fbb4358dcb63a14ae6824f09e9476f
5) 30c18908c6f9b545dafa30edfc24f5fbd808ed69343f701c1f8d6501fe83cbdf
6) 9017c070ad6ac9ac52e361286b3ff24a315f721f488b53b7aaf6ac35de477f44
7) 765327e1dc0888c69c92203d90037c5154db9787f54d3fc8f1097830be8c76ab
8) a9d483c0f021b72a94324562068d8164f8cce0aa8f779faea304669390775436
9) c3c50adcc0a5cd2b39677f17fb5f2efca52cc4e47ccd2cdbbf38815d426be9e1
10) ee06c557f1acd5c4948b1df0413e49f3885f8ac96185a9d986b91a1231444541

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 6713403015feb8959093f5d007bcbdbb3be9eec96dd62f517786b67506067251

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 3aad14d200887119f316be71d71aec11735dd3698a4fcaa50902fce71bdccb07
2) 6ad7b3e0873c9ff122c32006fdc3675706a03c4778287085a020d839b74cd780
3) 14da004cc96b910fb75abb86df09e318d92f4fb8dda39c8bd6a8e0601b6605d8
4) 14e547bebaa738b8605ba4182c4379317d121e268f846c0ed3da171375e65fe4
5) 932778732711cd18d5c4aabc507a65180bf1d4bd2b7d2d4e5506be4b8193596e
6) a5989c480ec6506247325652a1f3cb415934675de3877270ae0f65edd9b14d13
7) c1c4e677b36a2ee6ae858546e727e73cc38c95c9024c724f939178b3c03de906

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → rundll32.exe

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) 3ae02fc1fdb653997eeb9303305f1ec35dbb87eb603b573bd94895f35542f1a8

έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → rundll32.exe → rundll32.exe

Η ροή των διεργασιών στα δείγματα με hash sha256 :

1) af538ab1b8bdfbf5b7f1548d72c0d042eb14d0011d796cab266f0671720abb4d

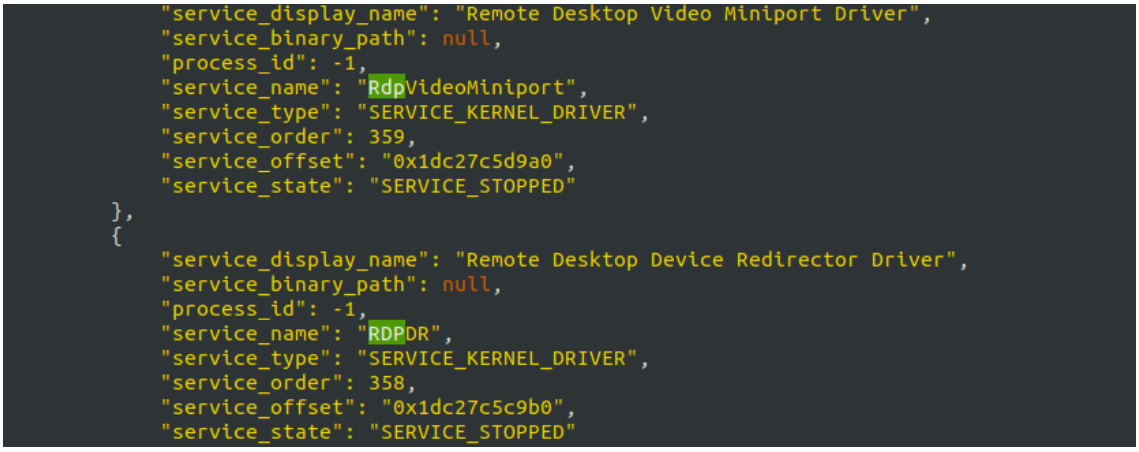
έχει ως εξής :

Smss → Winlogon → Userinit → explorer.exe → python → python → inject-x86.exe → rundll32.exe → rundll32.exe → rundll32.exe → rundll32.exe → rundll32.exe → rundll32.exe

Το Egregor δείχνει πολύ ισχυρούς μηχανισμούς για την αποφυγή ανάλυσης [29] από το Cuckoo Sandbox μας αλλά και από άλλες online λύσεις. Αυτό γίνεται φανερό από τον λιγοστό αριθμό signatures του Cuckoo που παρουσιάζουμε παρακάτω. Δοκιμάζοντας μερικές online λύσεις δεν εντόπιζαν ούτε ότι είναι κακόβουλα τα αρχεία, πράγμα που σημαίνει ότι δεν έχουν ενημερώσει τους μηχανισμούς ανάλυσης των συστημάτων τους καθώς οι απειλές αυτές είναι ήδη γνωστές.

Από θεωρητικές πηγές ανακαλύψαμε ότι χρησιμοποιείται το εργαλείο Cobalt Strike για το τον απομακρυσμένο έλεγχο και στην συνέχεια κατεβαίνει αρχείο bat μέσω του WMIC από το διαδίκτυο που περιέχει την κύρια λειτουργία του Egregor.Έμφαση δίνεται στο εφαρμόζονται προχωρημένες τεχνικές obfuscation, data exfiltration και lateral movement. Από τις παραπάνω αναλύσεις στην ροή των διεργασιών βλέπουμε ότι γίνεται εκτεταμένη χρήση του rundll32.exe LOLBAS που φανερώνει ότι τρέχει τις λειτουργίες του Egregor μέσω DLL αρχείου για να μην τραβήξει την προσοχή μηχανισμών ασφαλείας όπου το πετυχαίνει με μεγάλη επιτυχία.

Στα αποτελέσματα του Json αρχείου του δείγματος af538ab1b8bdfbf5b7f1548d72c0d042eb14d0011d796cab266f0671720abb4d εντοπίσαμε πως γίνεται διακοπή σε 425 υπηρεσίες του συστήματος όπου αρκετές είναι φανερό πως σχετίζονται με την ασφάλεια του συστήματος όπως η διακοπή λειτουργίας του Firewall και του Application Layer Gateway. Ωστόσο εντοπίσαμε την διακοπή υπηρεσιών που σχετίζονται με το RDP πρωτόκολλο που είναι συνηθισμένος στόχος των Ransomware. Στην παρακάτω εικόνα φαίνεται η διακοπή των υπηρεσιών RdpVideoMiniport και RDPDR.



Εικόνα 93: RDP Services Stopped

11.2 Μηχανισμοί Λειτουργίας

Παρακάτω παρατίθενται οι μηχανισμοί λειτουργίας που εντοπίστηκαν με βάση τα signatures του Cuckoo Sandbox στα δείγματα του Ryuk.

1) Allocates read-write-execute memory (usually to unpack itself)
2) Queries for the computername
3) This executable has a PDB path
4) Communicates with host for which no DNS query was performed
5) Performs some HTTP requests
6) File has been identified by 55 AntiVirus engines on VirusTotal as malicious
7) Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)

11.3 Μηχανισμοί Λειτουργίας για την Αποφυγή Ανίχνευσης (Evasion Techniques)

Τα Evasion Techniques που εντοπίστηκαν στα δείγματα με βάση τα signatures του Cuckoo Sandbox.

1) The binary likely contains encrypted or compressed data indicative of a packer
2) The executable contains unknown PE section names indicative of a packer (could be a false positive)
3) The executable uses a known packer
4) Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation
5) One or more thread handles in other processes
6) Resumed a suspended thread in a remote process potentially indicative of process injection
7) Stopped Application Layer Gateway service
8) Kernel module without a name
9) PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary
10) Malfind detects one or more injected processes
11) Stopped Firewall service

11.4 Κρυπτογράφηση

Πριν την εκκίνηση των μηχανισμών κρυπτογράφησης του Egregor θα πρέπει πρώτα να ολοκληρωθεί η ανίχνευση πληροφοριών καθώς και ο τερματισμός ορισμένων διεργασιών. Για την κρυπτογράφηση των αρχείων χρησιμοποιείται ο αλγόριθμος ChaCha20. Επιπρόσθετα γίνεται χρήση του RSA-2048 για την προστασία των κλειδιών του ChaCha20 [30]. Επιπρόσθετα αναφέρεται από πολλαπλές πηγές ότι το Egregor χρησιμοποιεί τεχνικές Obfuscation παρόμοιες με τα Ransomware Sekhmet και Maze. Τέλος βλέπουμε και το μήνυμα πληρωμής λύτρων από το δείγμα c3c50adcc0a5cd2b39677f17fb5f2efca52cc4e47ccd2cdbbf38815d426be9e1.

```
------------------
| What happened? |
------------------

Your network was ATTACKED, your computers and servers were LOCKED,
Your private data was DOWNLOADED.
----------------------
| What does it mean? |
----------------------
It means that soon mass media, your partners and clients WILL KNOW about your PROBLEM.
-------------------------
| How it can be avoided? |
-------------------------
In order to avoid this issue,
you are to COME IN TOUCH WITH US no later than within 3 DAYS and conclude the data recovery and breach fixing AGREEMENT.
-----------------------------------------
| What if I do not contact you in 3 days? |
-----------------------------------------
If you do not contact us in the next 3 DAYS we will begin DATA publication.
-----------------------------
| I can handle it by myself |
-----------------------------
It is your RIGHT, but in this case all your data will be published for public USAGE.
------------------------------
| I do not fear your threats! |
------------------------------
That is not the threat, but the algorithm of our actions.
If you have hundreds of millions of UNWANTED dollars, there is nothing to FEAR for you.
That is the EXACT AMOUNT of money you will spend for recovery and payouts because of PUBLICATION.
-------------------------
| You have convinced me! |
-------------------------
Then you need to CONTACT US, there is few ways to DO that.

I. Recommended (the most secure method)

   a) Download a special TOR browser: https://www.torproject.org/
   b) Install the TOR browser
   c) Open our website with LIVE CHAT in the TOR browser: http://egregor4u5ipdzhv.onion/55DA7C733EAD630
   d) Follow the instructions on this page.

II. If the first method is not suitable for you

   a) Open our website with LIVE CHAT: https://egregor.top/55DA7C733EAD630
   b) Follow the instructions on this page.

Our LIVE SUPPORT is ready to ASSIST YOU on this website.
---------------------------------------
| What will I get in case of agreement |
---------------------------------------
You WILL GET full DECRYPTION of your machines in the network, FULL FILE LISTING of downloaded data,
confirmation of downloaded data DELETION from our servers, RECOMMENDATIONS for securing your network perimeter.

And the FULL CONFIDENTIALITY ABOUT INCIDENT.

--------------------------------------------------------------------------------
Do not redact this special technical block, we need this to authorize you.
---EGREGOR---
1K/j2P6gIA6OwqS73T8EkiQ9dpyyFqNSfrHrnNh/63uq3eOO8yBRwLRNRzQwMOlXlnBArhViB4+wSlQYlyZJBv37AzKDDpqEfaH2uI1Wl81aDBF/suQ/mTm5
QQryDBNVAdjjOaefbUbbbhe3vWTC1ji3zEs03kTINOjLuZc/CoNfjfApmy7y3FEQ3oISD6B8Y6L7hJncI7LLe3lpwiqUI3v6FiBKwHRmzd8mb9fIoh6jtpU5
W6uxCYO6BByv5fgNCiT+UjZDU7JtU6t8W+STaBybDubIEGv57rtJiDyS+zY1V3L/547v8QOKmHDwzLgXqL1VQ1PvG7SCZXYB1leNtbtf6BqYytdpOMDB45kmz
---EGREGOR---
```

Εικόνα 94: Egregor Ransom Message

12 Στατιστικά Στοιχεία Αναλύσεων

Παραπάνω σε κάθε ενότητα του εκάστοτε Ransomware, δείχνουμε τα Signatures [16] που εντοπίστηκαν στα είκοσι δείγματά του συνολικά ώστε να καταλάβουμε όχι την συχνότητα εμφάνισης του κάθε Signature αλλά το τι μπορεί το συγκεκριμένο Ransomware να κάνει. Δεν αναφέρουμε πόσες φορές εμφανίστηκε το κάθε Signature από τα είκοσι δείγματα του καθενός, καθώς οι μηχανισμοί anti-analysis είναι διαφορετικοί σε κάθε δείγμα και μερικοί μπορεί να μην έχουν ακόμα ανακαλυφθεί από το Cuckoo. Με ισχυρούς μηχανισμούς anti-analysis τα δείγματα εκτελούνται μερικώς και κρύβουν ορισμένες λειτουργίες τους.

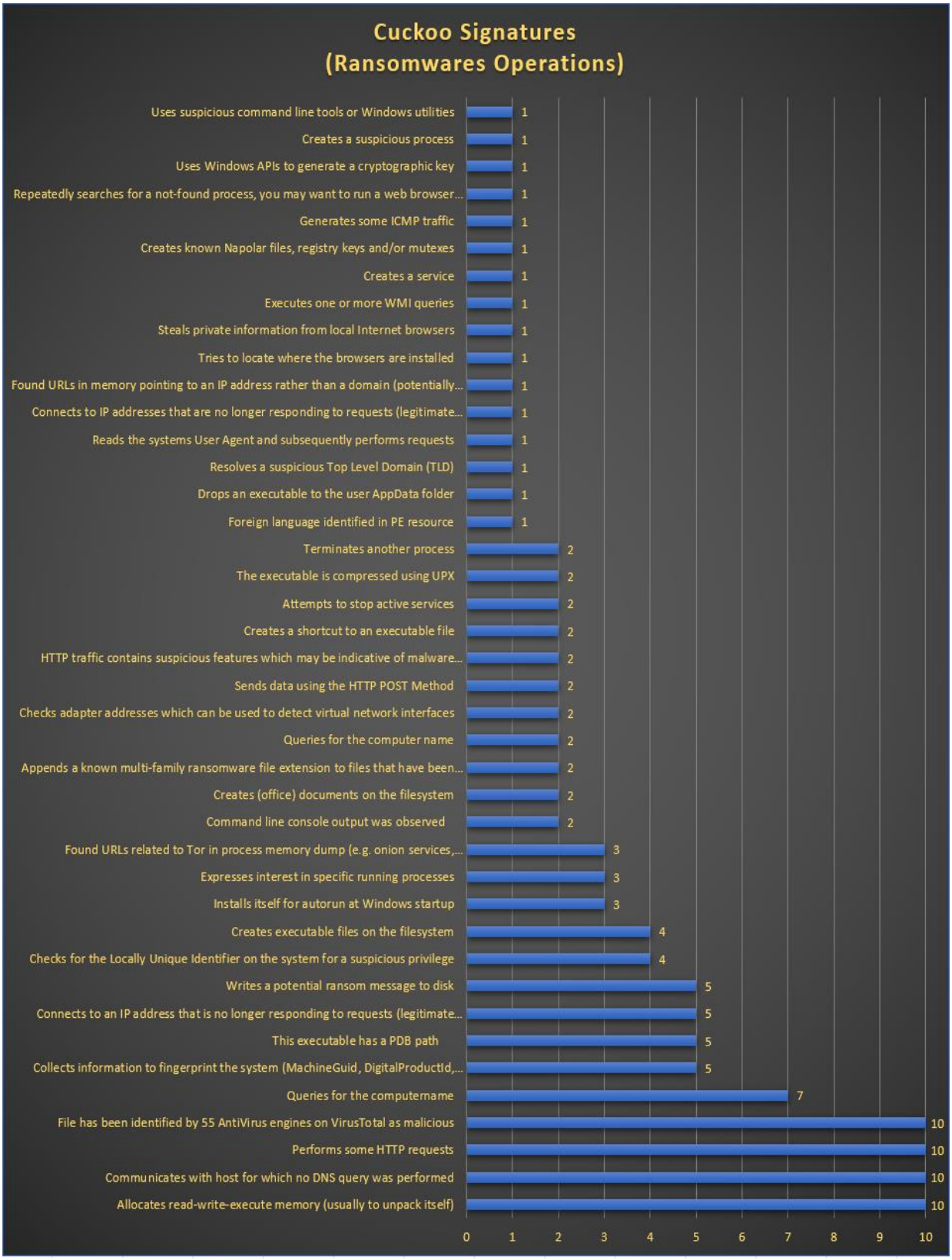
Στην παρούσα ενότητα βλέπουμε τα στατιστικά στοιχεία σχετικά με την λειτουργικότητα των Ransomware αλλά και τις τεχνικές αποφυγής ανάλυσης. Στους παρακάτω δύο πίνακες αναφέρουμε ποια Cuckoo Signatures εντοπίσαμε και σε πόσα από τα δέκα Ransomware που αναλύσαμε εντοπίστηκαν. Αυτό το κάνουμε για να δούμε ποιες τεχνικές χρησιμοποιούν περισσότερο τα Ransomware ώστε να μπορούμε να τα αναγνωρίζουμε εύκολα και να ξέρουμε με τι έχουμε να κάνουμε. Σε έναν οργανισμό που έχει παραβιαστεί αν εντοπίσουμε αρκετές από τις συμπεριφορές των πινάκων παρακάτω μπορούμε να διαπιστώσουμε ότι χτυπήθηκε από Ransomware και ίσως να προλάβουμε την κρυπτογράφηση των αρχείων αν γίνουν έγκαιρα συγκεκριμένες ενέργειες. Επιπρόσθετα ορισμένες λειτουργίες τις εντοπίσαμε μόνο σε μερικά Ransomware ή και σε μόνο ένα. Αυτό σημαίνει ότι με την εμφάνιση της συγκεκριμένης λειτουργίας ακόμα και αν δεν χρησιμοποιούμε το Cuckoo Sandbox θα μπορούμε να αναγνωρίσουμε τον ακριβή τύπο του Ransomware. Ακόμα από τις τεχνικές αποφυγής θα μπορούμε να δούμε πιο λειτουργικό εικονικών μηχανήματων ανιχνεύεται ευκολότερα από τα Ransomware [31].

No	Cuckoo Signatures (Ransomware Operations)	Ποσοστό εμφάνισης
1.	Allocates read-write-execute memory (usually to unpack itself)	10/10
2.	Communicates with host for which no DNS query was performed	10/10
3.	Performs some HTTP requests	10/10
4.	File has been identified by 55 AntiVirus engines on VirusTotal as malicious	10/10
5.	Queries for the computername	7/10
6.	Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)	5/10
7.	This executable has a PDB path	5/10
8.	Connects to an IP address that is no longer responding to requests (legitimate services will remain up-and-running usually)	5/10
9.	Writes a potential ransom message to disk	5/10
10.	Checks for the Locally Unique Identifier on the system for a suspicious privilege	4/10
11.	Creates executable files on the filesystem	4/10
12.	Installs itself for autorun at Windows startup	3/10
13.	Expresses interest in specific running processes	3/10
14.	Found URLs related to Tor in process memory dump (e.g. onion services, Tor2Web, and Ransomware)	3/10
15.	Command line console output was observed	2/10
16.	Creates (office) documents on the filesystem	2/10
17.	Appends a known multi-family ransomware file extension to files that have been encrypted	2/10
18.	Queries for the computer name	2/10
19.	Checks adapter addresses which can be used to detect virtual network interfaces	2/10

20.	Sends data using the HTTP POST Method	2/10
21.	HTTP traffic contains suspicious features which may be indicative of malware related traffic	2/10
22.	Creates a shortcut to an executable file	2/10
23.	Attempts to stop active services	2/10
24.	The executable is compressed using UPX	2/10
25.	Terminates another process	2/10
26.	Foreign language identified in PE resource	1/10
27.	Drops an executable to the user AppData folder	1/10
28.	Resolves a suspicious Top Level Domain (TLD)	1/10
29.	Reads the systems User Agent and subsequently performs requests	1/10
30.	Connects to IP addresses that are no longer responding to requests (legitimate services will remain up-and-running usually)	1/10
31.	Found URLs in memory pointing to an IP address rather than a domain (potentially indicative of Command & Control traffic)	1/10
32.	Tries to locate where the browsers are installed	1/10
33.	Steals private information from local Internet browsers	1/10
34.	Executes one or more WMI queries	1/10
35.	Creates a service	1/10
36.	Creates known Napolar files, registry keys and/or mutexes	1/10
37.	Generates some ICMP traffic	1/10
38.	Repeatedly searches for a not-found process, you may want to run a web browser during analysis	1/10
39.	Uses Windows APIs to generate a cryptographic key	1/10
40.	Creates a suspicious process	1/10
41.	Uses suspicious command line tools or Windows utilities	1/10

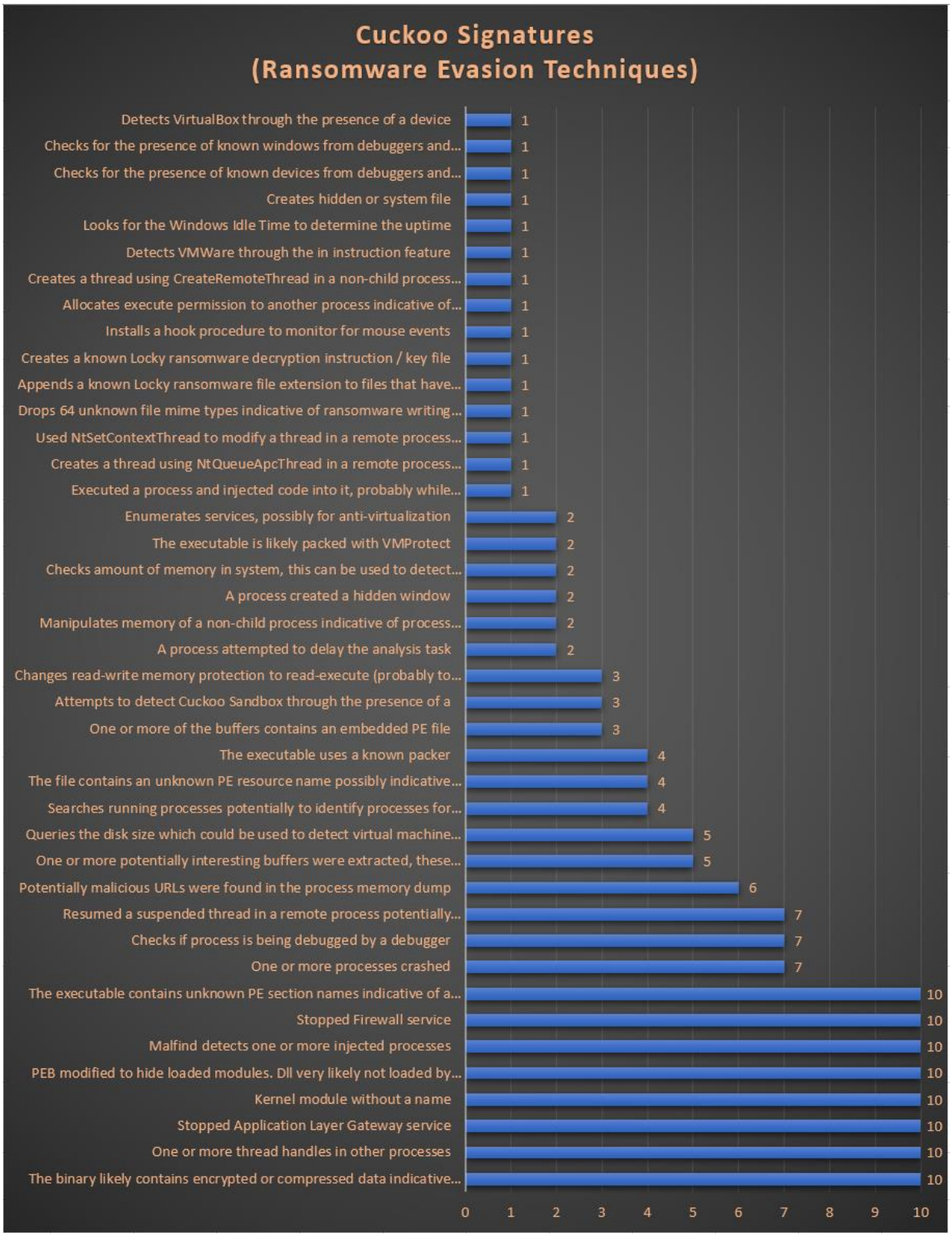
No	Cuckoo Signatures (Ransomware Evasion Techniques)	Ποσοστό εμφάνισης
1.	The binary likely contains encrypted or compressed data indicative of a packer	10/10
2.	One or more thread handles in other processes	10/10
3.	Stopped Application Layer Gateway service	10/10
4.	Kernel module without a name	10/10
5.	PEB modified to hide loaded modules. Dll very likely not loaded by LoadLibrary	10/10
6.	Malfind detects one or more injected processes	10/10
7.	Stopped Firewall service	10/10
8.	The executable contains unknown PE section names indicative of a packer (could be a false positive)	10/10
9.	One or more processes crashed	7/10
10.	Checks if process is being debugged by a debugger	7/10
11.	Resumed a suspended thread in a remote process potentially indicative of process injection	7/10
12.	Potentially malicious URLs were found in the process memory dump	6/10

13.	One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc	**5/10**
14.	Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation	**5/10**
15.	Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping	**4/10**
16.	The file contains an unknown PE resource name possibly indicative of a packer	**4/10**
17.	The executable uses a known packer	**4/10**
18.	One or more of the buffers contains an embedded PE file	**3/10**
19.	Attempts to detect Cuckoo Sandbox through the presence of a	**3/10**
20.	Changes read-write memory protection to read-execute (probably to avoid detection when setting all RWX flags at the same time)	**3/10**
21.	A process attempted to delay the analysis task	**2/10**
22.	Manipulates memory of a non-child process indicative of process injection	**2/10**
23.	A process created a hidden window	**2/10**
24.	Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available	**2/10**
25.	The executable is likely packed with VMProtect	**2/10**
26.	Enumerates services, possibly for anti-virtualization	**2/10**
27.	Executed a process and injected code into it, probably while unpacking	**1/10**
28.	Creates a thread using NtQueueApcThread in a remote process potentially indicative of process injection	**1/10**
29.	Used NtSetContextThread to modify a thread in a remote process indicative of process injection	**1/10**
30.	Drops 64 unknown file mime types indicative of ransomware writing encrypted files back to disk	**1/10**
31.	Appends a known Locky ransomware file extension to files that have been encrypted	**1/10**
32.	Creates a known Locky ransomware decryption instruction / key file	**1/10**
33.	Installs a hook procedure to monitor for mouse events	**1/10**
34.	Allocates execute permission to another process indicative of possible code injection	**1/10**
35.	Creates a thread using CreateRemoteThread in a non-child process indicative of process injection	**1/10**
36.	Detects VMWare through the in instruction feature	**1/10**
37.	Looks for the Windows Idle Time to determine the uptime	**1/10**
38.	Creates hidden or system file	**1/10**
39.	Checks for the presence of known devices from debuggers and forensic tools	**1/10**
40.	Checks for the presence of known windows from debuggers and forensic tools	**1/10**
41.	Detects VirtualBox through the presence of a device	**1/10**

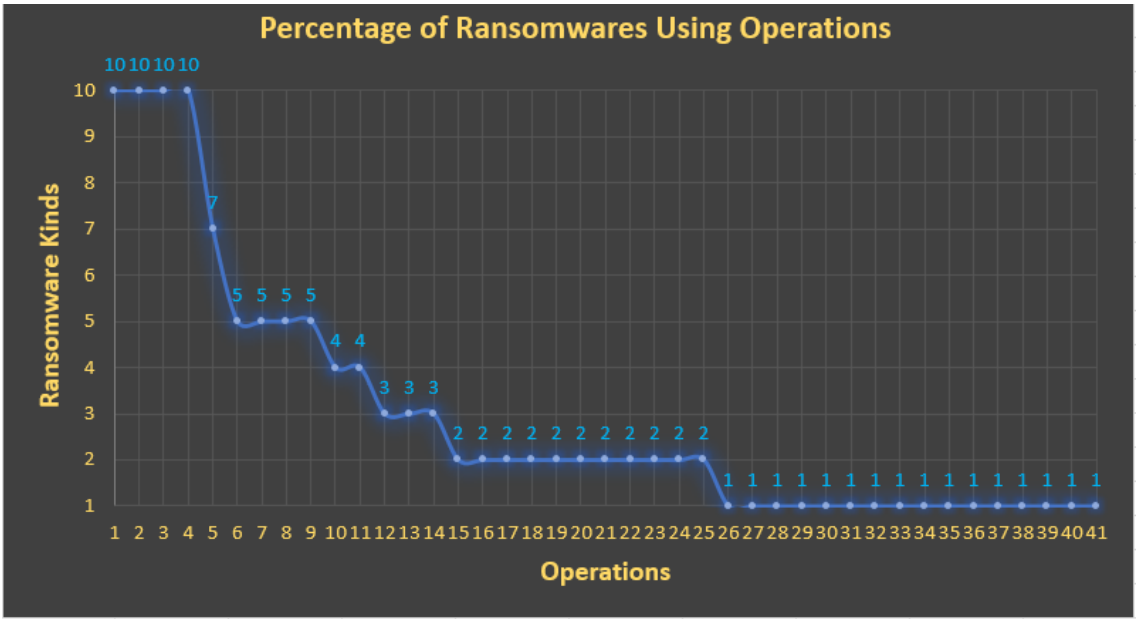


Εικόνα 95: Ransomware Operations

Παρακάτω βλέπουμε τα γραφήματα των παραπάνω πινάκων σχετικά με τις λειτουργίες και τις τεχνικές αποφυγής της ανάλυσης ώστε να κατανοήσουμε καλύτερα την συχνότητα εμφάνισής τους.

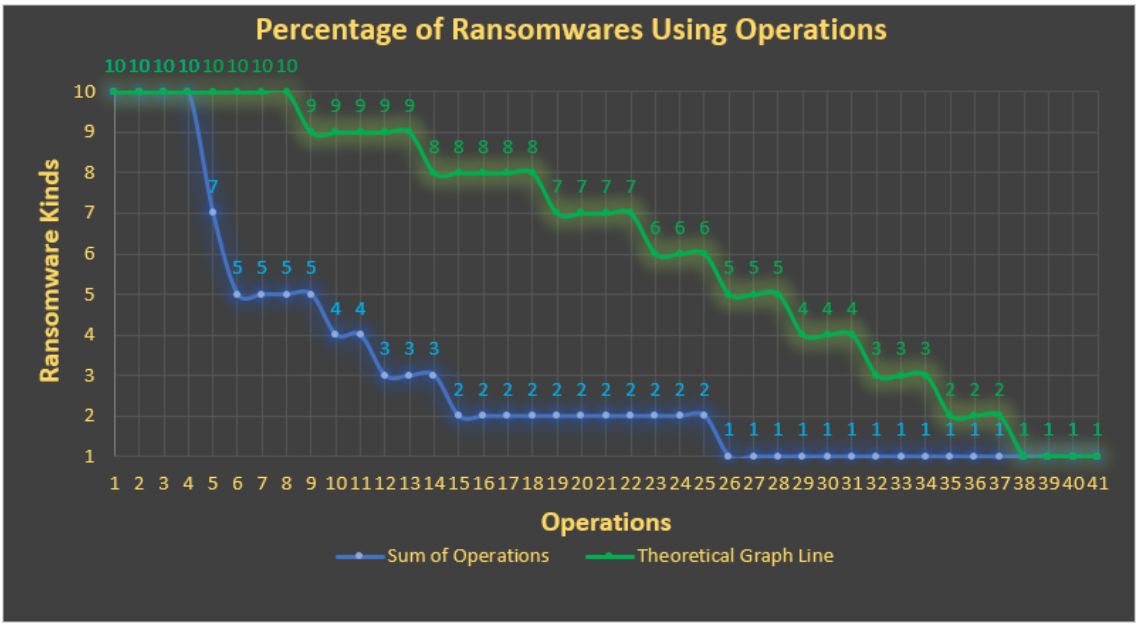


Εικόνα 96: Ransomware Evasion Techniques



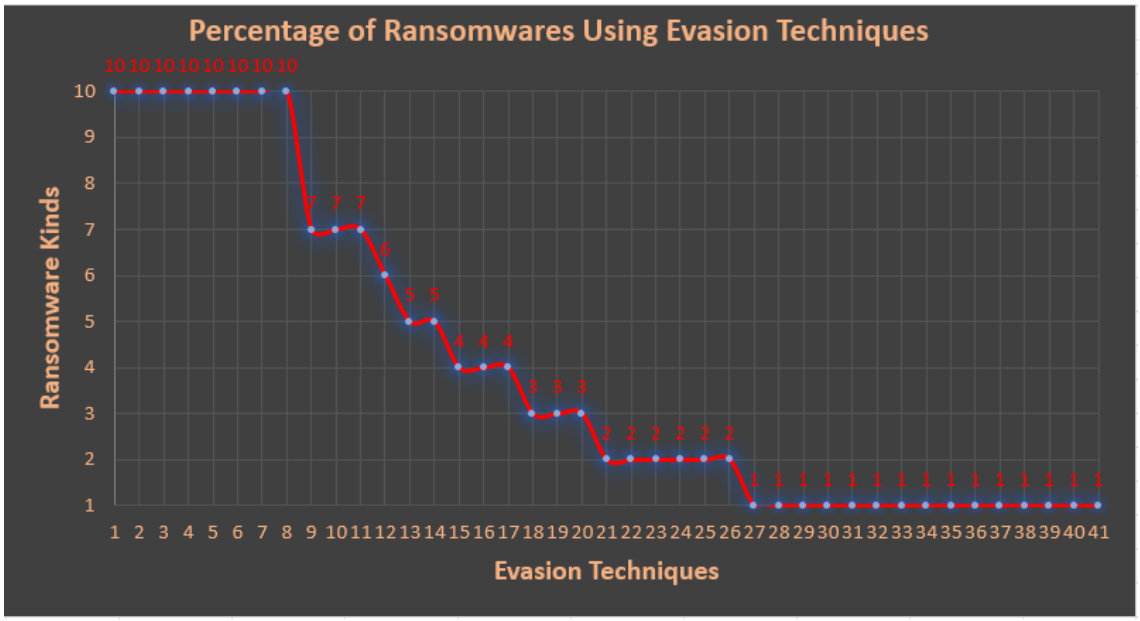
Εικόνα 97: Percentage of Ransomware Using Operations

Από το γράφημα των λειτουργιών μπορούμε να καταλήξουμε σε αρκετά συμπεράσματα. Καταρχάς βλέπουμε ότι οι τέσσερις πρώτες λειτουργίες με δέκα στα δέκα εκτελούνται σε όλα τα είδη Ransomware είτε τα αυτά εκτελεστούν πλήρως ή μερικώς, δηλαδή δεν αποκρύπτονται από την ανάλυση. Όπως είδαμε παραπάνω σε αναλύσεις αρκετά Ransomware χρησιμοποιούσαν ισχυρούς μηχανισμούς anti-analysis οπότε αρκετές λειτουργίες δεν εκδηλώθηκαν και δεν εντοπίστηκαν από το Cuckoo. Οπότε όσο καλύτερα έχουμε στήσει το εικονικό μας μηχάνημα και όσο πιο προηγμένες τεχνολογίες χρησιμοποιούμε [32] όλο και περισσότερες λειτουργίες θα εντοπίζονται και τόσο η γραφική μας παράσταση θα τείνει να υψώνεται προς τα πάνω όπως φαίνεται στο θεωρητικό μας γράφημα παρακάτω με πράσινο χρώμα. Επίσης η χρήση ενός bare metal περιβάλλοντος θα μπορούσε να δείξει περισσότερα αποτελέσματα Το πράσινο γράφημα δεν υπολογίστηκε με βάση κάποιον υπολογισμό καθώς χρησιμοποιήθηκε για την καλύτερη δυνατή επεξήγηση.



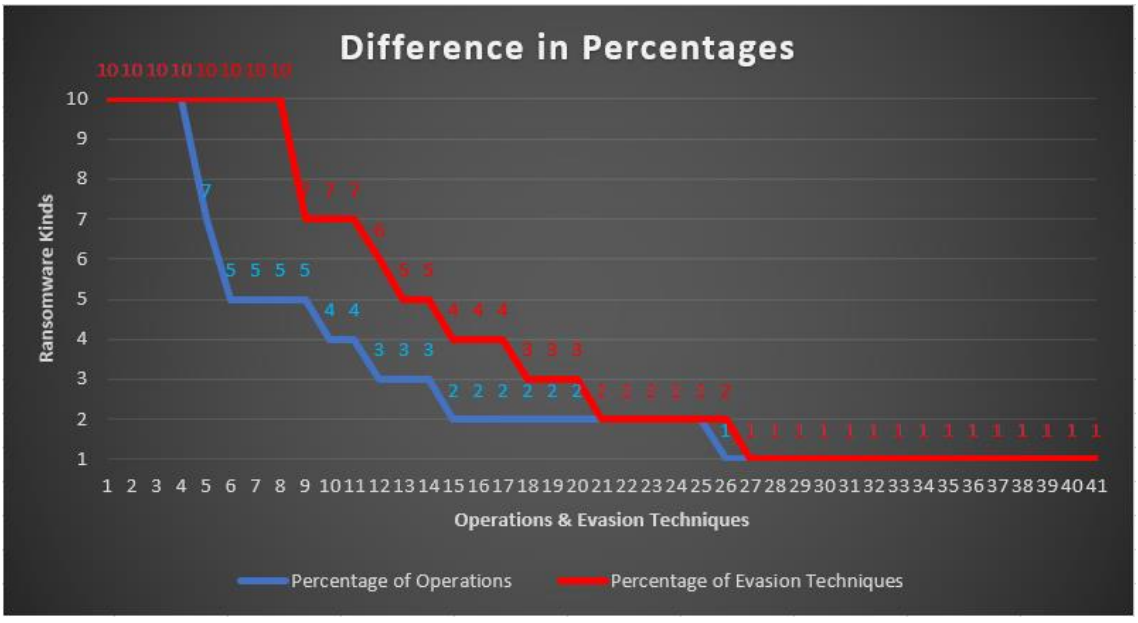
Εικόνα 98: Percentage of Ransomware Using Operations 2

Στην παρακάτω γραφική φαίνεται πως οι οκτώ πρώτοι μηχανισμοί anti-analysis δεν μπορούν να αποφύγουν τον εντοπισμό από το Cuckoo. Επίσης αν μετρήσουμε το πλήθος των τεχνικών αποφυγής ανάλυσης στα δέκα είδη μας βλέπουμε ότι είναι εκατόν εξήντα πέντε ενώ οι λειτουργίες είναι εκατόν είκοσι. Αυτό σημαίνει ότι έχει δοθεί ιδιαίτερη έμφαση στις τεχνικές αποφυγής. Επίσης βλέπουμε ότι ένα μεγάλο μέρος τους εκτελείται στα πρώτα βήματα εκτέλεσης του Ransomware.



Εικόνα 99: Percentage of Ransomware Using Evasion Techniques

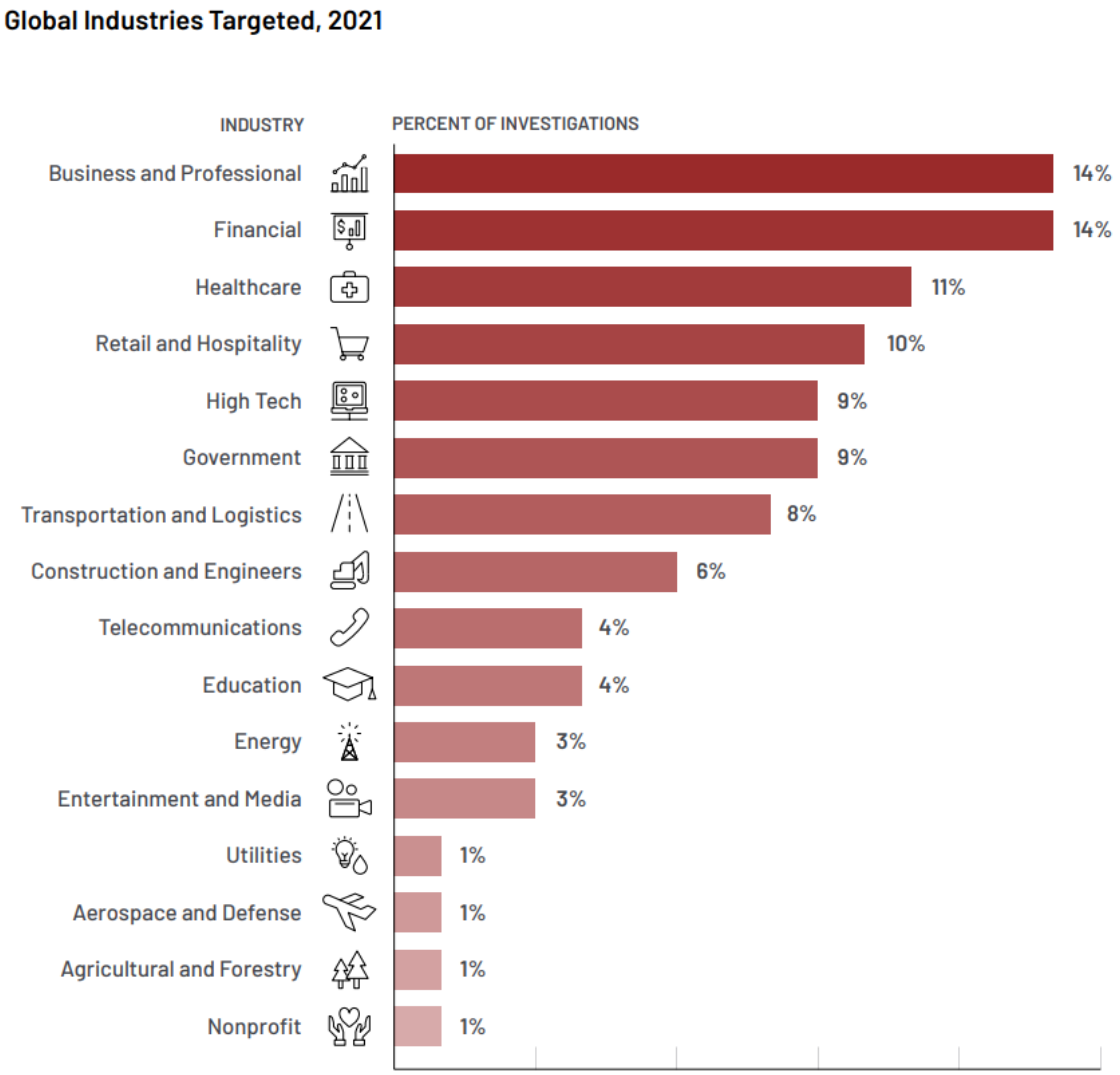
Στα γραφήματα του Difference in Percentages βλέπουμε σχηματικά την διαφορά στην εμφάνιση των λειτουργιών σε σχέση με των τεχνικών anti-analysis.



Εικόνα 100: Difference in Percentages

13 Στατιστικά Στοιχεία Επιθέσεων Ransomware

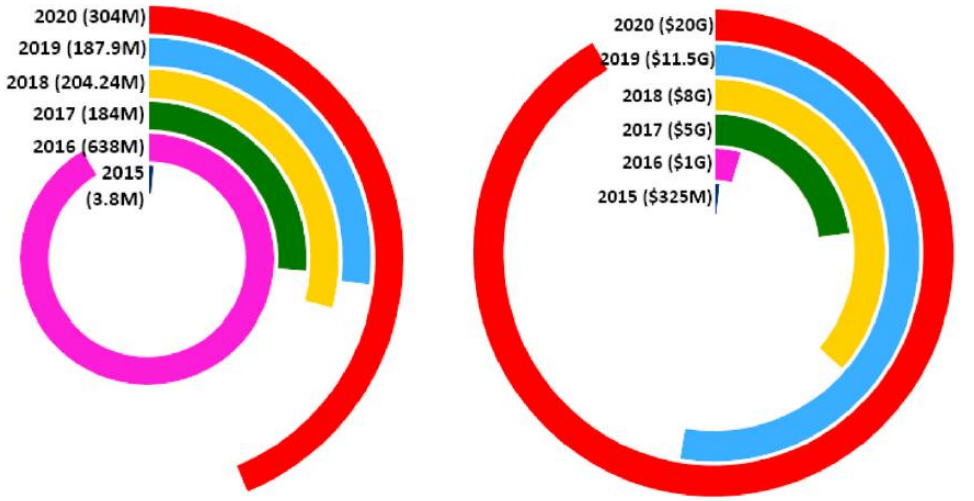
Μερικά από τα παραπάνω Ransomware που αναλύσαμε φημολογείται ότι έχουν φτάσει στο τέλος της ζωής τους και ότι οι ομάδες πίσω από αυτά έχουν διαλυθεί. Πολλοί ίσως να επαναπαυθούν καθώς πιστεύουν ότι αυτές η απειλές δεν υπάρχουν πια. Εμείς ωστόσο θα δείξουμε ότι δεν ισχύει αυτό καθώς μπορεί να σταμάτησε μια Ransomware ομάδα να λειτουργεί, αλλά δεν σημαίνει ότι τα μέλη της δεν θα ενταχτούν σε άλλες ομάδες ή ότι δεν θα αναγεννηθούν κάτω από ένα πιο ισχυρό κακόβουλο λογισμικό. Τα κίνητρα είναι πολλά καθώς εκτός από την ίδια την ομάδα του Ransomware κερδίζουν και άλλοι συνεργάτες εγκληματίες του διαδικτύου, που υποστηρίζουν το μοντέλο RaaS (Ransomware as a service) με σκοπό να πάρουν την αρχική πρόσβαση και να εισάγουν το Ransomware μέσα στα μηχανήματα στόχους. Στην εικόνα [33] εδώ βλέπουμε τους προτιμώμενους στόχους των Ransomware ομάδων για το προηγούμενο έτος.



Εικόνα 101: Προτιμώμενοι στόχοι Ransomware επιθέσεων

Από την παραπάνω εικόνα γίνεται εύκολα κατανοητό ότι κύριο στόχος τους είναι οργανισμοί που έχουν μεγάλα κέρδη και άρα σημαντικές υποδομές που δεν θα μπορούν να ανεχτούν την επίθεση καθώς αν γίνει παύση σε ορισμένους μηχανισμούς μπορούν να χαθούν μεγάλα χρηματικά ποσά μέχρι και ανθρώπινες ζωές. Στην επόμενη εικόνα [34] δείχνουμε τον

εκτιμώμενο αριθμό των επιθέσεων από Ransomware και την οικονομική καταστροφή που προκάλεσαν μέχρι το έτος δύο χιλιάδες είκοσι.



a) Estimated number of ransomware attacks **b) The amount of damages caused by ransomware**

Εικόνα 102: Στατιστικά κατά προσέγγιση των επιθέσεων και του κόστους

14 Τακτικές Αντιμετώπισης Ransomware και LOLBAS

Αρχικά θα αναφέρουμε γενικές τακτικές που μπορούν να λειτουργήσουν σε Ransomwares και στην συνέχεια θα επεκταθούμε σε ειδικότερες τακτικές αντιμετώπισης των LOLBAS που χρησιμοποιούν. Για την γενικότερη αντιμετώπιση Ransomware επιθέσεων μπορούμε να κάνουμε:

1. Διατήρηση πολλαπλών Backup σε εξωτερικούς δίσκους ή σε Cloud Server.
2. Συνεχής αναβάθμιση και ενημέρωση των συστημάτων και λειτουργικών.
3. Χρήση ενός συστήματος IDS (Intrusion Detection System).
4. Δημιουργία σχεδίου αντιμετώπισης από την ομάδα ασφαλείας σε περίπτωση επίθεσης.
5. Ανασκόπηση των ρυθμίσεων των Ports ώστε να μην υπάρχουν ανοιχτές πόρτες που δεν χρησιμοποιούνται για κάποιο σκοπό.
6. Ρύθμιση των τελικών (Endpoint) συστημάτων με βάση την ασφάλεια για ελαχιστοποίηση των αδυναμιών.
7. Εκπαίδευση των εργαζομένων.
8. Χρήση λύσεων ασφαλείας για την προστασία των emails.
9. Whitelisting των εφαρμογών ώστε μόνο οι εξουσιοδοτημένες να μπορούν να κατεβάσουν αρχεία και να τα εκτελέσουν.
10. Περιορισμός πρόσβασης των χρηστών ώστε να κάνουν χρήση μόνο των υπηρεσιών που χρειάζονται.
11. Τακτική εκτέλεση δοκιμών ασφαλείας στα συστήματα από εξειδικευμένους τεχνικούς για την εύρεση νέων αδυναμιών.

Όσον αφορά την αντιμετώπιση των κακόβουλων συμπεριφορών των LOLBAS αρχείων θα χρειαστούν εξειδικευμένες τεχνικές. Η πρώτη λύση που μπορούμε να χρησιμοποιήσουμε είναι μέσω Yara κανόνων και παρεμφερή εργαλείων όπως το Elastic και Splunk [35]. Δηλαδή μπορούμε να δημιουργήσουμε κανόνες ειδικά σχεδιασμένους για την ανίχνευση LOLBAS. Έπειτα μπορούμε να κάνουμε ανάλυση στα αρχεία log του Command Line και του Powershell και αν εντοπίζεται η χρήση LOLBAS να ενημερώνει τον χρήστη. Επίσης αυτό θα μπορούσε να γίνει και την ώρα της εκτέλεσης κάποιας τέτοιας εντολής. Συνεχίζοντας θα μπορούσε να γίνεται ανάλυση της σχέσης γονέα και παιδιού (Parent and Child) των διεργασιών ώστε να αναγνωρίζονται οι φυσιολογικές και οι ύποπτες σχέσεις, σε περίπτωση που μια διεργασία έχει ως γονέα μια μη επιτρεπτή διεργασία. Μια ακόμα ενδιαφέρουσα τακτική είναι μέσω της συλλογής διάφορων μετρητικών στοιχείων και χρήσης αυτών όπως bytes που διαβάστηκαν και γράφτηκαν, αριθμός φακέλων που ανοίχτηκαν, δικαιώματα χρήσης και χρόνος χρήσης του επεξεργαστή. Μέσω αυτών τον στοιχείων δημιουργούνται γραφήματα προέλευσης και γίνονται συγκρίσεις ώστε αν ένα από αυτά ξεπεράσει ένα ορισμένο όριο, τότε αυτό και όλα τα υπόλοιπα στοιχεία που συνδέονται μαζί του να θεωρηθούν κακόβουλα. Τέλος εντοπίσαμε πως προτείνεται η χρήση ενός Active Learning Framework ονόματος LOLAL. Το LOLAL κάνει χρήση μοντέλων Machine Learning ώστε να εντοπίσει επιθέσεις. Ξεκινώντας το LOLAL τροφοδοτείται από πολλαπλά δείγματα και ταξινομούνται από τον αναλυτή ως κακόβουλα η καλόβουλα ώστε να εκπαιδευτεί στο να τα ξεχωρίζει. Με την ολοκλήρωση της εκπαίδευσης μπορεί και εντοπίζει της κακόβουλες δραστηριότητες και να κάνει παρουσίαση των κακόβουλων εντολών που εντοπίστηκαν στα δείγματα. Ο εντοπισμός γίνεται σε εντολές στο Powershell και σε διεργασίες μέσω μοτίβων και δεδομένων προέλευσης.

15 Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή παρουσιάστηκαν δέκα από τα πιο δημοφιλή Ransomware μέσω διακοσίων δειγμάτων, εξετάσαμε την λειτουργικότητά τους, την ροή των διεργασιών τους, τους μηχανισμούς αποφυγής της ανίχνευσης και ανάλυσης τους, εντοπίσαμε LOLBAS αρχεία, δώσαμε έμφαση σε ξεχωριστές λειτουργίες του καθενός, είδαμε τα μηνύματα λύτρων και τα είδη των κρυπτογραφικών μεθόδων που χρησιμοποιεί το κάθε Ransomware και τέλος γραφήματα συνόλων συγκρίσεων και στατιστικών. Οι αναλύσεις πραγματοποιήθηκαν μέσω του Cuckoo Sandbox σε Windows 10 ειδικά διαμορφωμένο για να μην εντοπιστεί από τα Ransomware. Εντοπίσαμε μια πληθώρα τεχνικών όπως process injection, παρεμβάσεις στα δικαιώματα χρήσης των αρχείων, κάλεσμα ύποπτων API, το κατέβασμα ύποπτων αρχείων, μολυσμένα αρχεία office και άλλα.

Μέσω της παρουσίασης αυτής της διατριβής μπορεί να γίνει κατανοητή από τον αναγνώστη η βαρύτητα μιας Ransomware επίθεσης, τα όρια που μπορεί να φτάσει το Cuckoo Sandbox καθώς και οι περιορισμοί της δυναμικής ανάλυσης μέσω εικονικών μηχανημάτων και ιδιαίτερα μέσω του Virtual Box. Επίσης μέσω της επίδειξης των επικίνδυνων λειτουργιών μπορούν να σχεδιαστούν μηχανισμοί ασφαλείας ώστε να αποφευχθούν αρκετές λειτουργίες του Ransomware η έστω να χτυπήσει συναγερμός στο κέντρο ασφαλείας ώστε να γίνει πρόληψη και άμεση εξυγίανση του συστήματος. Το θεμελιώδες ζήτημα των επαναλαμβανόμενων επιθέσεων από Ransomware είναι η έλλειψη κατάλληλου ελέγχου πρόσβασης σε αρχεία και πόρους λειτουργικού συστήματος, και για να διασφαλιστεί αυτό οι συμπεριφορές κώδικα θα πρέπει να συνάδουν με τις προθέσεις των χρηστών. Τα Ransomware θα συνεχίσουν να εξελίσσονται με νέα χαρακτηριστικά και βελτιωμένες προσπάθειες μείωσης ή πλήρους παράκαμψης των υπαρχόντων μηχανισμών ασφαλείας, οπότε σαν μελλοντικές προεκτάσεις του θέματος προτείνονται συνεχείς έρευνες για την παροχή επαρκούς και κατάλληλης πρόσβασης, ώστε να μην περιορίζεται η λειτουργία από τον χρήστη αλλά να εμποδίζονται πλήρως οι Ransomware επιθέσεις.

Χρήσιμοι σύνδεσμοι

https://cuckoosandbox.org/
https://github.com/d4rksystem/VBoxCloak
https://github.com/a0rtega/pafish
https://attack.mitre.org/
https://www.enisa.europa.eu/
https://docs.microsoft.com/en-us/windows/
https://blog.malwarebytes.com/101/2015/12/an-introduction-to-image-file-execution-options/
https://gchq.github.io/CyberChef/
https://www.varonis.com/blog/darkside-ransomware
https://www.hhs.gov/sites/default/files/demystifying-blackmatter.pdf
https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=conti%20ransomware
https://any.run/
https://tria.ge/
https://www.joesandbox.com/#windows
https://www.mandiant.com/
https://research.checkpoint.com/
http://essay.utwente.nl/93265/

Βιβλιογραφικές Αναφορές

[1] Digit Oktavianto, Iqbal Muhardianto, Cuckoo Malware Analysis, Birmingham B3 2PB, UK. 35 Livery Street: Packt Publishing Ltd., 2013.

[2] Bukhteyev, Alexey, «research.checkpoint.com,» 7 2 2022. [Ηλεκτρονικό]. Available: https://research.checkpoint.com/2022/invisible-cuckoo-cape-sandbox-evasion/.

[3] Vasilios Koutsokostas, Constantinos Patsakis, «Python and Malware: Developing Stealth and Evasive Malware Without Obfuscation,» σε *18th International Conference on Security and Cryptography*, Athens Research Center, Artemidos 6, Marousi 15125, Greece, 2021.

[4] Columbus, Louis, «www.Forbes.com,» 30 7 2019. [Ηλεκτρονικό]. Available: https://www.forbes.com/sites/louiscolumbus/2019/07/30/how-to-deal-with-ransomware-in-a-zero-trust-world/.

[5] Sense, Secure, «https://securesense.ca/ransomware-works-via-carbon-black/,» 20 9 2016. [Ηλεκτρονικό]. Available: https://securesense.ca/wp-content/uploads/2016/09/Ransomware_Anatomy_of_Attack_Carbon_Black.jpg.

[6] Vasilios Koutsokostas, Nikolaos Lykousas, Theodoros Apostolopoulos, Gabriele Orazi, Amrita Ghosal, Fran Casino, Mauro Conti, Constantinos Patsakis, «Automated analysis of malicious Microsoft Office documents,» *Computers & Security,* p. 13, 2022.

[7] Talha Ongun, Jack W. Stokes, Jonathan Bar Or, Ke Tian, Farid Tajaddodianfar, Joshua Neil, Christian Seifert, Alina Oprea, John C. Platt, «Living-Off-The-Land Command

Detection Using Active Learning,» *ACM International Conference Proceeding Series,* Τόμ. %1 από %2pp. 442-455, p. 14, 2021.

[8] Stamp, Ryan, «Living-off-the-Land Abuse Detection Using Natural Language Processing and Supervised Learning,» *School of Computer Science, University of Guelph, Canada,* p. 7, 2022.

[9] Pavel Yosifovich, Alex Ionescu, Mark E. Russinovich, David A. So, «Windows Internals, Part 1,» σε *System architecture, processes, threads, memory management, and more*, Microsoft Press, 2017, p. 801.

[10] A, Monnappa K, Learning Malware Analysis, 35 Livery Street Birmingham B3 2PB, UK.: Packt Publishing Ltd, 2018.

[11] DiMaggio Jon, The Art Of Cyberwarfare (An Investigator's Guide to Espionage), San Francisco: No Starch Press, 2022.

[12] Michael Sikorski, Andrew Honig, Practical Malware Analysis (The Hands-On Guide to Dissecting Malicious Software), San Francisco: no strarch press, 2012.

[13] Anthony Cheuk Tung Lai, Ping Fan Ke, Kelvin Chan, Siu Ming Yiu, Dongsun Kim, Wai Kin Wong, Shuai Wang, Joseph Muppala, Alan Ho, «RansomSOC: A More Effective Security Operations Center to Detect and Respond to Ransomware Attacks,» *Journal of Internet Services and Information Security (JISIS),* τόμ. 12, p. 13, 2022.

[14] Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters, The Art of Memory Forensics, 10475 Crosspoint Boulevard Indianapolis, Indiana IN 46256: John Wiley & Sons, Inc., 2014.

[15] Theodoros Apostolopoulos, Vasilios Katos, Kim-Kwang Raymond Choo, Constantinos Patsakis, «Resurrecting Anti-virtualization and Anti-debugging: Unhooking your Hooks,» *Future Generation Computer Systems,* pp. 116, pp. 393-405, 2021.

[16] Umara Urooj, Bander Ali Saleh Al-rimy, Anazida Zainal, Fuad A. Ghaleb, Murad A. Rassam, «Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions,» MDPI, Johor Bahru, 2021.

[17] Lifars, «A Detailed Analysis of The Last Version of Conti Ransomware,» 244 Fifth Avenue, Suite 2035, New York, 2021.

[18] Saleh Alzahrani, Yang Xiao, Wei Sun, «An Analysis of Conti Ransomware Leaked Source Codes,» *IEEEAccess,* p. 16, 2022.

[19] McAfee, «Ransomware-NetWalker,» 27 1 2021. [Ηλεκτρονικό]. Available: https://kcm.trellix.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/94000/KB94078/en_US/McAfee_Labs_Threat_Advisory_Netwalker_Ransomware.pdf.

[20] Gallo, Allan Liska and Timothy, Ransomware Defending Against Digital Extortion, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2016.

[21] Hasherezade, «Malwarebytes,» 1 3 2016. [Ηλεκτρονικό]. Available: https://www.malwarebytes.com/blog/news/2016/03/look-into-locky.

[22] Abhijit Mohanta, Anoop Saldanha, Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect, New York: Apress, 2020.

[23] A. Brandt, «Sophos News,» 11 6 2021. [Ηλεκτρονικό]. Available: https://news.sophos.com/en-us/2021/06/11/relentless-revil-revealed/.

[24] Jordan Nuce, Jeremy Kennelly, Kimberly Goody, Andrew Moore, Alyssa Rahman, Matt Williams, Brendan Mckeague, Jared Wilson, «Mandiant,» 11 5 2021. [Ηλεκτρονικό]. Available: https://www.mandiant.com/resources/blog/shining-a-light-on-darkside-ransomware-operations.

[25] Elias, Marc, «McAfee,» 21 7 2021. [Ηλεκτρονικό]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/new-ryuk-ransomware-sample%E2%80%AFtargets-webservers/.

[26] Acronis, «Acronis,» 4 6 2021. [Ηλεκτρονικό]. Available: https://www.acronis.com/en-us/blog/posts/ragnar-locker/.

[27] N. Digital, «Digital NHS UK,» 28 9 2020. [Ηλεκτρονικό]. Available: https://digital.nhs.uk/cyber-alerts/2020/cc-3624.

[28] USA, Department of Health & Human Services, «hhs.gov,» 9 2 2021. [Ηλεκτρονικό]. Available: https://www.hhs.gov/sites/default/files/demystifying-blackmatter.pdf.

[29] Amir Afianian, Salman Niksefat, Babak Sadeghiyan, David Baptiste, «Malware Dynamic Analysis Evasion Techniques: A Survey,» ACM Comput., 2019.

[30] ANSSI, «Egregor Ransomware,» 2 3 2021. [Ηλεκτρονικό]. Available: https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-007.pdf.

[31] Oyama, Yoshihiro, «Trends of anti-analysis operations of malwares observed,» *Journal of Computer Virology and Hacking Techniques,* p. 17, 2018.

[32] Alexei Bulazel, Bülent Yener, «A survey on automated dynamic malware analysis evasion and counter-evasion: PC, Mobile, and Web,» σε *ACM International Conference Proceeding Series*, Vienna, 2017.

[33] Mandiant, «Special Report | Mandiant M-Trends 2022,» 31 12 2021. [Ηλεκτρονικό]. Available: https://www.mandiant.com/sites/default/files/2022-04/M-Trends%202022%20Executive%20Summary.pdf.

[34] Masoudeh Keshavarzi, Hamid Reza Ghaffary, «Computers in Human Behavior,» *An ontology-driven framework for knowledge representation of digital extortion attacks,* τόμ. 139, p. 16, 2022.

[35] Utz Nisslmueller, «LOLBin detection through unsupervised learning, An approach based on explicit featurization of the command line and parent-child,» University of Twente, Enschede, Netherlands, 2022.