Διεθνές Πανεπιστήμιο Ελλάδος
Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Command and Control (C2) Περιβάλλοντος για Εκπαιδευτικούς Σκοπούς»

Του φοιτητή:
Αρβανιτίδη Φώτιου
Αρ. Μητρώου: 174961

Επιβλέπων:
Χαράλαμπος Μπράτσας
Επίκουρος Καθηγητής

Μάιος 2025

Τίτλος Π.Ε.: *Ανάπτυξη Command and Control (C2) Περιβάλλοντος για Εκπαιδευτικούς Σκοπούς*
Κωδικός Π.Ε.: 25149
Ονοματεπώνυμο φοιτητή: Φώτιος Αρβανιτίδης
Ονοματεπώνυμο εισηγητή: Χαράλαμπος Μπράτσας
Ημερομηνία ανάληψης Δ.Ε.: 07-03-2025
Ημερομηνία περάτωσης Δ.Ε.: 23-05-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένεια και τους φίλους μου,

για την αμέριστη αγάπη και υποστήριξη που μου έδειξαν.»

Πρόλογος

Έπειτα από αρκετή σκέψη σχετικά με την επιλογή θέματος για την πτυχιακή μου εργασία, λαμβάνοντας υπ'όψιν το ενδιαφέρον μου για την επιθετική κυβερνοασφάλεια, τα δίκτυα υπολογιστών και τη διαχείριση συστημάτων, αποφάσισα να συνδυάσω τα παραπάνω γνωστικά αντικείμενα, καταλήγωντας έτσι στο συγκεκριμένο θέμα. Πέρα από την προσωπική μου επιθυμία να εμβαθύνω στις μεθοδολογίες και στρατηγικές που εφαρμόζονται σε ένα ήδη παραβιασμένο σύστημα, βασικός μου στόχος ήταν να επεξηγήσω στο κοινό τα γεγονότα που ακολουθούν μετά την παραβίαση ενός συστήματος. Θεωρώ ότι συχνά, η προσοχή στρέφεται προς την αρχική εισβολή και την εκμετάλλευση ευπαθειών, ωστόσο είναι κρίσιμο να γίνει αντιληπτό το ότι η κύρια ζημιά δεν προκαλείται από την ίδια την παραβίαση, αλλά από τις κακόβουλες ενέργειες που ακολουθούν μέσα στο δίκτυο και τις σοβαρές συνέπειές τους.

Περίληψη

Έχοντας διανύσει σχεδόν το ένα τέταρτο του 21ου αιώνα, είναι πλέον φανερό ότι οι κυβερνοεπιθέσεις πληθαίνουν, καθιστώντας επιτακτική την κατανόηση των μεθόδων που χρησιμοποιούν οι επιτιθέμενοι, ειδικά μετά την αρχική παραβίαση ενός συστήματος. Η παρούσα πτυχιακή εργασία εστιάζει ακριβώς σε αυτή τη φάση, διερευνώντας θεωρητικά και πρακτικά τις υποδομές Διοίκησης και Ελέγχου (C2). Αρχικά, γίνεται μια επισκόπηση του θεωρητικού πλαισίου, εξετάζοντας την εξέλιξη των C2, τις αρχιτεκτονικές τους, τα κανάλια επικοινωνίας, και τον ρόλο εργαλείων όπως το PowerShell και των τεχνικών κρυπτογράφησης. Στη συνέχεια, η εργασία παρουσιάζει τον σχεδιασμό και την υλοποίηση ενός εκπαιδευτικού C2 framework. Το σύστημα αυτό, βασισμένο σε αρχιτεκτονική Server-Agent, περιλαμβάνει έναν server σε Python με διαδικτυακή διεπαφή Flask και έναν agent σε PowerShell για περιβάλλοντα Windows. Η μεταξύ τους επικοινωνία προστατεύεται με κρυπτογράφηση AES σε επίπεδο εφαρμογής, αξιοποιώντας ένα προκαθορισμένο κλειδί και ένα απλό πρωτόκολλο για την ανταλλαγή δεδομένων μέσω TCP.

Το αναπτυχθέν πλαίσιο δοκιμάστηκε επιτυχώς, επιδεικνύοντας βασικές και προχωρημένες λειτουργίες C2, όπως η απομακρυσμένη εκτέλεση εντολών, η συλλογή πληροφοριών συστήματος, η λήψη στιγμιότυπων οθόνης, η περιήγηση στο σύστημα αρχείων του στόχου και η αμφίδρομη μεταφορά αρχείων. Τα αποτελέσματα αυτά αναλύθηκαν και αντιστοιχίστηκαν με τις τακτικές του πλαισίου MITRE ATT&CK. Η εργασία συμπεραίνει ότι ακόμη και ένα εκπαιδευτικό C2 framework μπορεί να αναδείξει τις σημαντικές δυνατότητες που αποκτά ένας επιτιθέμενος, τονίζοντας τις προκλήσεις που ενέχει η ανάπτυξη τέτοιων συστημάτων και την αξία της πρακτικής κατανόησης για την ενίσχυση της κυβερνοάμυνας.

«Developing a Command and Control (C2) Environment for Educational Purposes»

«Fotios Arvanitidis»

Abstract

In an era of escalating cyberattacks, understanding the methods employed by adversaries, particularly in the post-exploitation phase, is paramount. This undergraduate thesis focuses on this critical stage by theoretically exploring and practically implementing an educational Command and Control (C2) framework. Initially, an overview of the theoretical landscape is provided, examining the evolution of C2 systems, their architectures, communication channels, and the role of tools like PowerShell and encryption techniques. Subsequently, the thesis details the design and implementation of this educational C2 framework. The system, based on a Server-Agent architecture, comprises a Python-based server with a Flask web user interface and a PowerShell agent for Windows environments. Communication between these components is secured using application-level AES encryption with a pre-shared key, employing a simple protocol for data exchange over TCP.

The developed framework was successfully tested, demonstrating core and advanced C2 functionalities, including remote command execution, system information gathering, screen capture, agent file system browsing, and bidirectional file transfer. These results were analyzed and mapped to tactics within the MITRE ATT&CK framework. The thesis concludes that even an educational C2 framework can reveal the significant capabilities afforded to an attacker, underscoring the challenges inherent in developing such systems and the value of practical understanding for enhancing cybersecurity defenses.

Ευχαριστίες

Αρχικά, ευχαριστώ από τα βάθη της καρδιάς μου τον Θεό, τους γονείς και τους φίλους μου για την πολυεπίπεδη υποστήριξη τους και τα πολύτιμα εφόδια που μου παρέχουν όλα αυτά τα χρόνια. Επιπλέον, οφείλω να ευχαριστήσω τον κ. Χαράλαμπο Μπράτσα για την ειλικρίνεια, τη συνεργασιμότητα και τη βοήθεια που μου παρείχε πριν και κατά τη διάρκεια της εργασίας αυτής. Ακόμα, θα ήθελα να ευχαριστήσω τον κ. Αργύρη Χατζόπουλο και τον κ. Αντώνη Σαρηγιαννίδη, επειδή μου έδωσαν τα κατάλληλα εναύσματα για να ανακαλύψω το ενδιαφέρον μου για τα δίκτυα και την κυβερνοασφάλεια, δίνοντας μου έτσι μια κατεύθυνση στον χαοτικό κλάδο της πληροφορικής.

Χωρίς αυτούς, πιστεύω ότι δε θα είχα πετύχει όσα έχω καταφέρει μέχρι τώρα, και γι' αυτό νιώθω πραγματικά ευγνώμων και τους εκτιμώ πολύ.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Συντομογραφίες

ACK Acknowledgement (Επιβεβαίωση)

AES Advanced Encryption Standard

API Application Programming Interface (Διεπαφή Προγραμματισμού Εφαρμογών)

APT Advanced Persistent Threat (Προηγμένη Επίμονη Απειλή)

ATT&CK Adversarial Tactics, Techniques, and Common Knowledge

AV Antivirus

BOM Byte Order Mark (Σήμανση Σειράς Byte)

C2 Command and Control (Διοίκηση και Έλεγχος)

CBC Cipher Block Chaining

CERT Certificate

CLI/ΔΓΕ Command-line Interface (Διεπαφή Γραμμής Εντολών)

CN Common Name

DNS Domain Name System

EDR Endpoint Detection and Response

EOF End of File

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IoC Indicator of Compromise

IP Internet Protocol

IPS Intrusion Prevention System

IV Initialization Vector

JSON JavaScript Object Notation

MITRE Όνομα Οργανισμού

OS Operating System (Λειτουργικό Σύστημα)

PEM Privacy-Enhanced Mail

PS PowerShell

RSA Rivest-Shamir-Adleman

SIEM Security Information and Event Management

SMB Server Message Block

SSL Secure Sockets Layer

SYN Synchronize (Συγχρονισμός - μέρος της χειραψίας TCP)

TLS Transport Layer Security (Ασφάλεια Επιπέδου Μεταφοράς)

UI User Interface (Διεπαφή Χρήστη)

URL Uniform Resource Locator (Ενιαίος Προσδιοριστής Πόρων)

VM Virtual Machine (Εικονική Μηχανή)

Κεφάλαιο 1

Εισαγωγή

1.1 Πρόλογος

Στην αυγή της ψηφιακής εποχής, ο προσωπικός υπολογιστής μετατράπηκε από ένα απλό εργαλείο σε μια προέκταση της ύπαρξής μας - πύλη επικοινωνίας, εργασίας και ψυχαγωγίας. Τι συμβαίνει, όμως, όταν αυτή η πύλη παραβιάζεται και ο έλεγχος περνά, αθέατα, σε ξένα χέρια; Φανταστείτε ένα σενάριο όπου, ενώ εσείς εργάζεστε ανυποψίαστοι, ένας άγνωστος χειριστής δίνει εντολές στο σύστημά σας, παρακολουθεί τις κινήσεις σας και αποσπά τα δεδομένα σας. Αυτή η ανησυχητική πραγματικότητα βρίσκεται στον πυρήνα της φάσης μετά-εκμετάλλευσης στις κυβερνοεπιθέσεις, μια φάση που ενορχηστρώνεται μέσω εξειδικευμένων υποδομών διοίκησης και ελέγχου (C2). Η παρούσα εργασία επιχειρεί να εισέλθει σε αυτό το σκιώδες πεδίο, όχι για να εξοπλίσει το κοινό με κακόβουλα εργαλεία, αλλά για να ρίξει φως στους μηχανισμούς πίσω από αυτόν τον έλεγχο, αποκαλύπτοντας τη δύναμη που ασκείται σε ένα παραβιασμένο σύστημα και υπενθυμίζοντας την επιτακτική ανάγκη για κατανόηση και άμυνα στον 21ο αιώνα.

1.2 Το Ψηφιακό Τοπίο και οι Προκλήσεις Ασφάλειας

Στις μέρες μας, είναι ευρέως αντιληπτό πως η ενασχόληση του κοινού με τον ψηφιακό χώρο αυξάνεται εκθετικά. Νέοι χρήστες από κάθε υπόβαθρο, νέες συσκευές και νέα δεδομένα παντός τύπου και σημασίας, εισέρχονται και αλληλεπιδρούν μεταξύ τους στο μεγαλύτερο δίκτυο (υπολογιστών) ευρείας πρόσβασης του κόσμου, γνωστό και ως διαδίκτυο, κάθε δευτερόλεπτο που περνά [1]. Αυτή η αυξημένη διασύνδεση και εξάρτηση από την ψηφιακή τεχνολογία, ενώ προσφέρει πρωτοφανείς δυνατότητες, δημιουργεί ταυτόχρονα, σε συνδυασμό με το αυξανόμενο χάσμα ανάμεσα στη ζήτηση και στην προσφορά ειδικών κυβερνοασφάλειας [2], εύφορο έδαφος για καινοτόμες, ατομικές ή συλλογικές δράσεις κακόβουλων παραγόντων εναντίον νόμιμων επιχειρήσεων, οργανισμών και τελικών χρηστών, με τα κίνητρα να ποικίλλουν από οικονομικά έως και ιδεολογικά ή κρατικά υποκινούμενα. [3], [4], [5]

Δυστυχώς, η τεχνολογική ανάπτυξη, εξέλιξη και βελτίωση καινούριων ή υφιστάμενων συστημάτων, συσκευών, μηχανισμών κ.ο.κ. δεν είναι απαραίτητα συνυφασμένη με την εξασφάλιση της προστατευμένης,

εξουσιοδοτημένης και αδιάκοπης χρήσης των τελευταίων. Συχνά, τα ζητήματα ασφαλείας είτε παραλείπονται τελείως, είτε τίθενται σε «δεύτερη μοίρα» από τους σχεδιαστές ή/και τους δημιουργούς ενός έργου [6], [7]. Αυτό μπορεί να οφείλεται σε παράγοντες όπως η πίεση για γρήγορη κυκλοφορία στην αγορά (time-to-market), η έλλειψη εξειδικευμένων γνώσεων ασφαλείας ή η αντίληψη της ασφάλειας ως κόστους παρά ως επένδυσης [7], [8]. Για πολλούς η ασφάλεια είναι το «αναγκαίο κακό» που προλαμβάνει ή περιορίζει τις απώλειες που προκύπτουν έπειτα από κάποιο περιστατικό, και δεν επιφέρει κέρδη - τουλάχιστον άμεσα.

Συνεπώς, κάπου ανάμεσα στο χρονικό διάστημα μεταξύ υλοποίησης και ασφάλισης ενός προϊόντος ή συστήματος, «ευδοκιμούν» οι ευπάθειες (vulnerabilities), τα κενά ασφαλείας και κατ' επέκταση, η δυνατότητα για μη εξουσιοδοτημένη είσοδο και χρήση του, γνωστή και ως κυβερνοεπίθεση.

1.3 Ο Κύκλος Ζωής μιας Κυβερνοεπίθεσης και η Φάση Μετά-Εκμετάλλευσης

Μια τυπική κυβερνοεπίθεση μπορεί να θεωρηθεί ότι ακολουθεί μια σειρά από στάδια, όπως αυτά που περιγράφονται σε μοντέλα σαν το Cyber Kill Chain της Lockheed Martin [9] και το ATT&CK της MITRE Corporation [10]. Αυτά τα στάδια περιλαμβάνουν την αναγνώριση (reconnaissance), την προετοιμασία του παράγοντα που θα διατελέσει χρέη όπλου (weaponization), την παράδοση (delivery), την εκμετάλλευση (exploitation), την εγκατάσταση (installation) κακόβουλου λογισμικού, τη Διοίκηση & Έλεγχο (Command & Control) και τέλος, τις ενέργειες προς την επίτευξη του στόχου (actions on objectives).

Η παρούσα εργασία θα επικεντρωθεί στη φάση που ακολουθεί την επιτυχή αρχική παραβίαση (exploitation/ installation) και περιλαμβάνει τη Διοίκηση & Έλεγχο και τις ενέργειες προς επίτευξη στόχου. Αυτή η φάση συχνά αναφέρεται συνολικά ως φάση μετά-εκμετάλλευσης (post-exploitation). Στο στάδιο αυτό, παρατηρείται πως οι επιτιθέμενοι χρησιμοποιούν την αρχική τους πρόσβαση (initial foothold) για να εγκαθιδρύσουν συνεχή και επίμονη παρουσία (persistency), να εκτελέσουν απομακρυσμένα διάφορες εντολές, προγράμματα και scripts (execution), να κινηθούν πλευρικά εντός του δικτύου (lateral movement), να κλιμακώσουν τα δικαιώματά τους (privilege escalation) και έτσι να επιτύχουν τους τελικούς τους στόχους – στο σύστημα-«ξενιστή» (host system) ή και σε ολόκληρο το δίκτυο.

1.4 Ορισμός και Ρόλος των Πλαισίων Διοίκησης και Ελέγχου (C2)

Φυσικά, για να πραγματοποιηθούν οι ενέργειες μετά-εκμετάλλευσης που αναφέρθηκαν προηγουμένως, ιδανικά με αθόρυβο, γρήγορο και αποτελεσματικό τρόπο, κρίνεται απαραίτητη η οργάνωση και η προετοιμασία του επιτιθέμενου, προτού ακόμα ληφθεί η οποιαδήποτε κακόβουλη ενέργεια. Μια ποικιλία εργαλείων, με το καθένα να επιτελεί τον δικό του κρίσιμο σκοπό, θα κληθεί να συγκεντρωθεί για να αποτελέσει το λεγόμενο πλαίσιο (framework) Διοίκησης και Ελέγχου (ή Εντολών και Ελέγχου, οι μεταφράσεις ποικίλλουν ωστόσο το νόημα παραμένει το ίδιο) - τον σημαντικότερο σύμμαχο ενός επιτιθέμενου που σκοπεύει να εκμεταλλευθεί στο έπακρο την αρχική του πρόσβαση σε ένα σύστημα. Αυτά τα πλαίσια παρέχουν την κεντρική υποδομή για τη διαχείριση των παραβιασμένων συστημάτων (agents) και την ενορχήστρωση των ενεργειών μετά-εκμετάλλευσης. Θα μπορούσαν να παρομοιαστούν ως το «κεντρικό

νευρικό σύστημα» μιας κακόβουλης επιχείρησης. Συνοπτικά, το C2 πλαίσιο, είναι μια προηγμένη πλατφόρμα που χρησιμοποιείται από χάκερς για τον απομακρυσμένο έλεγχο και τη διαχείριση συστημάτων που έχουν παραβιαστεί. Λειτουργεί ως ένας κεντρικός κόμβος ή κέντρο εντολών, από όπου ένας επιτιθέμενος μπορεί να διαχειρίζεται εκατοντάδες παραβιασμένα συστήματα μέσα σε ένα δίκτυο-στόχο [11]. Η ανάγκη για βαθύτερη κατανόηση αυτών των πολύπλοκων αλλά κρίσιμων πλαισίων και των κινδύνων που εγκυμονούν αποτέλεσε το κύριο κίνητρο για την εκπόνηση της παρούσας πτυχιακής εργασίας.

1.5 Το Πρόβλημα: Κίνδυνοι Μετά-Εκμετάλλευσης (Post-Exploitation)

Η επιτυχής εγκαθίδρυση ενός C2 καναλιού μετά από μια αρχική παραβίαση ανοίγει την πόρτα για μια πληθώρα κακόβουλων ενεργειών στη φάση μετά-εκμετάλλευσης, καθιστώντας την συχνά την πιο καταστροφική φάση μιας επίθεσης. Ο έλεγχος που παρέχει το C2 πλαίσιο, επιτρέπει στον επιτιθέμενο να μετατρέψει μια απλή παραβίαση σε μια βαθιά και διαρκή διείσδυση σε ένα σύστημα, με σοβαρές συνέπειες τις περισσότερες φορές. Οι κύριοι κίνδυνοι περιλαμβάνουν: Αρχικά, ένας από τους κυριότερους κινδύνους που υφίστανται, αποτελεί η εδραίωση παρουσίας ή αλλιώς, «μονιμότητα»/ επιμονή (persistence) κατά την οποία ο επιτιθέμενος μπορεί επί της ουσίας να χρησιμοποιήσει το πλαίσιο C2 για να εγκαταστήσει μηχανισμούς που εξασφαλίζουν ότι ο agent θα εκτελείται αυτόματα ακόμα και μετά από κάποια επανεκκίνηση του συστήματος, έτσι ώστε ο επιτιθέμενος να μη χάνει την πρόσβαση του χωρίς κάποια ειδική παρέμβαση. Έπειτα, ο επόμενος κίνδυνος είναι σίγουρα η συλλογή πληροφοριών, η οποία επιτυγχάνεται μέσω μιας «αλυσίδας» ενεργειών από τον επιτιθέμενο, όπως εκτέλεση εντολών που φανερώνουν κρίσιμες και ευαίσθητες πληροφορίες σχετικά με το σύστημα και το δίκτυο του στόχου (διεργασίες, usergroups, διευθυνσιοδότηση IP, έκδοση λειτουργικού συστήματος, πληροφορίες υλικολογισμικού). Οι πληροφορίες αυτές μπορούν να φανούν αρκετά χρήσιμες στον επιτιθέμενο για διάφορους λόγους.

Επιπλέον, ένας ακόμη κίνδυνος είναι η απομακρυσμένη εκτέλεση εντολών, η οποία περιγράφει τη δυνατότητα εκτέλεσης εντολών, σεναρίων κελύφους (scripts), ή ακόμη και ολόκληρων εκτελέσιμων αρχείων (binaries) από τη «φαρέτρα» του επιτιθέμενου, για την απόκτηση σχεδόν πλήρους ελέγχου του συστήματος-στόχου. Η κλιμάκωση δικαιωμάτων σε χρήστη με αυξημένα δικαιώματα (π.χ. Administrator) είναι επίσης αναπόσπαστο κομμάτι της διαδικασίας, στην περίπτωση που η αρχική πρόσβαση έγινε σε σύστημα με περιορισμένα δικαιώματα. Στη συνέχεια, δεν πρέπει να αποκλείεται ο κίνδυνος της «πλευρικής» κίνησης (lateral movement- διαφορετικό από το pivoting), κατά την οποία γίνεται χρήση του παραβιασμένου συστήματος ως ορμητήριου με στόχο την απόκτηση πρόσβασης σε άλλα συστήματα εντός του ίδιου δικτύου, καταλήγοντας σε ραγδαία και συνήθως καταστρεπτική εξάπλωση.

Τέλος, σοβαροί κίνδυνοι της φάσης της μετα-εκμετάλλευσης αποτελούν η συλλογή και υποκλοπή δεδομένων (collection and exfiltration) και η ανάπτυξη επιπλέον κακόβουλου λογισμικού. Ουσιαστικά, ενας επιτιθέμενος είναι σε θέση να αναζητήσει, να συλλέξει και να εξάγει ευαίσθητα δεδομένα όπως αρχεία με κωδικούς πρόσβασης, cookies, στιγμιότυπα οθόνης, ακόμα και καταγραφές πληκτρολόγησης (keylogging), ωστόσο μπορεί και να στείλει και να εγκαταστήσει δικά του έξτρα λογισμικά όπως ransomware, trojans, keyloggers κ.ο.κ, αν αυτά προσφέρουν λειτουργικότητα που δεν υπάρχει ήδη ενσωματωμένη στην υλοποίηση C2 που χρησιμοποιείται. Άρα, η κατανόηση αυτών των κινδύνων υπογραμμίζει τη σημασία της έγκαιρης ανίχνευσης και διακοπής της C2 επικοινωνίας.

1.6 Ερευνητικός Στόχος και Αντικειμενικοί Σκοποί

Παρακάτω λοιπόν, θα εξερευνηθούν και θα αναλυθούν εκτενώς τα πλαίσια διοίκησης και ελέγχου, οι μεθοδολογίες και τεχνικές που ενσωματώνουν στον πυρήνα τους, η ιστορία τους, η συμπεριφορά τους και ο ρόλος τους ως κεντρικό νευρικό σύστημα για την ενορχήστρωση προηγμένων κυβερνοεπιθέσεων αλλά και δοκιμών παρείσδυσης (*penetration tests*) σε ελεγχόμενα, εταιρικά και μη, περιβάλλοντα. Επιπροσθέτως, σε μια προσπάθεια βαθύτερης κατανόησης των πλαισίων διοίκησης και ελέγχου, θα αναπτυχθεί ένα παραδειγματικό c2 πλαίσιο, το οποίο θα εκτελεί κάποιες βασικές, μα διόλου ακίνδυνες λειτουργίες. Τέλος, παραμένοντας πάντοτε στα ακαδημαϊκά και ηθικά όρια, θα μελετηθεί μια τυπική περίπτωση χρήσης ενός πλαισίου c2 από έναν κακόβουλο παράγοντα που στοχεύει έναν τελικό σταθμό για να αποσπάσει κρίσιμες πληροφορίες δίχως να γίνει αντιληπτός κατα τη διάρκεια των δράσεων του. Η επίδειξη αυτή, θα γίνει όχι μόνο για την αποσαφήνηση της περιπλοκότητας ολόκληρης της διαδικασίας και την έμφαση στους μηχανισμούς και τους κινδύνους της φάσης της μετα-εκμετάλλευσης, αλλά και με την ελπίδα για ευαισθητοποίηση και ενημέρωση του κοινού προς τα κρίσιμα ζητήματα ασφαλείας που το αφορούν είτε άμεσα είτε έμμεσα.

Για να επιτευχθούν οι παραπάνω στόχοι, τέθηκαν οι ακόλουθοι αντικειμενικοί σκοποί: Αρχικά, πρέπει να γίνει η κατασκευή του πρωτοτύπου. Αυτό σημαίνει πως πρέπει να σχεδιαστεί και να υλοποιηθεί ένα λειτουργικό πρωτότυπο πλαίσιο C2 που θα περιλαμβάνει τα βασικά στοιχεία, τα οποία είναι ο server, ένας agent για λειτουργικό σύστημα Windows (το πιο δημοφιλές στο κοινό) και μια διεπαφή ιστού για την εύκολη και αποτελεσματική διαχείριση του πλαισίου από τον χειριστή του.

Στη συνέχεια, κρίθηκε απαραίτητη η εφαρμογή κρυπτογράφησης στην επικοινωνία μεταξύ του server και του agent για την εξασφάλιση της εμπιστευτικότητας των δεδομένων. Η υλοποίηση αυτή πραγματοποιήθηκε με χρήση του συμμετρικού αλγορίθμου AES (Advanced Encryption Standard) σε επίπεδο εφαρμογής, με ένα προκαθορισμένο κοινό κλειδί και αρχικοποίησης (IV). Η προσέγγιση αυτή, αν και διαφορετική από την αρχική πρόθεση χρήσης TLS, επιλέχθηκε για λόγους απλοποίησης και αντιμετώπισης τεχνικών προκλήσεων, παρέχοντας ωστόσο το απαιτούμενο επίπεδο κρυπτογράφησης για τους εκπαιδευτικούς σκοπούς της εργασίας και την επίδειξη ενός μηχανισμού προστασίας της C2 επικοινωνίας.

Ακολουθούν οι βασικές λειτουργίες προς υλοποίηση εντός του συνολικού πλαισίου, οι οποίες είναι το επιτυχές και συχνό check-in του agent με τον server για την άμεση παρακολούθηση της κατάστασης του παραβιασμένου συστήματος, το tasking (αποστολή εντολών από τον χειριστή και σωστή διαχείριση τους από τον agent) και η επιτυχής και ευανάγνωστη συλλογή αποτελεσμάτων των αντολών που στάλθηκαν προς εκτέλεση. Αυτές οι λειτουργίες αποτελούν τον πυρήνα κάθε C2 συστήματος και είναι απαραίτητες για την επίδειξη των δυνατοτήτων του.

Τέλος, πολύ σημαντικό είναι να διερευνηθεί η χρήση του πλαισίου για την εκτέλεση θεμελιωδών τεχνικών αναγνώρισης (reconnaissance) και εκτέλεσης (execution). Ο στόχος είναι να φανεί πώς ένα απλό C2 μπορεί να διευκολύνει ενέργειες που συναντώνται συχνά σε πραγματικές επιθέσεις.

1.7 Εύρος και Περιορισμοί της Εργασίας

Η σκοπιμότητα της παρούσας πτυχιακής εργασίας έγκειται στην αναγνώριση της ανάγκης για βαθύτερη, πρακτική κατανόηση των σύγχρονων επιθετικών τεχνικών που χαρακτηρίζουν το τοπίο της κυβερνοασφάλειας. Η εστίαση στην κατασκευή ενός λειτουργικού πλαισίου Διοίκησης και Ελέγχου (C2) δεν αποσκοπεί στη δημιουργία ενός επιχειρησιακού εργαλείου, αλλά στην προσφορά μιας μοναδικής μαθησιακής εμπειρίας. Η πρωτοτυπία της προσέγγισης εντοπίζεται στην εκπαιδευτική σύνθεση και παρουσίαση των θεμελιωδών αρχών ενός C2, αξιοποιώντας ευρέως διαθέσιμα και κατανοητά εργαλεία, με στόχο την αποσαφήνιση των μηχανισμών μετά-εκμετάλλευσης.

Είναι, ωστόσο, κρίσιμο να οριοθετηθεί με σαφήνεια το εύρος του εγχειρήματος και να αναγνωριστούν οι εγγενείς του περιορισμοί, οι οποίοι απορρέουν άμεσα από τον πρωταρχικά εκπαιδευτικό του χαρακτήρα. Το πλαίσιο που αναπτύχθηκε, σχεδιάστηκε αποκλειστικά για σκοπούς επίδειξης και ακαδημαϊκής μελέτης, και ως εκ τούτου, απέχει σημαντικά από την πολυπλοκότητα και τις δυνατότητες απόκρυψης που διαθέτουν τα πραγματικά κακόβουλα ή επαγγελματικά εργαλεία διείσδυσης. Πιο συγκεκριμένα, δεν ενσωματώθηκαν τεχνικές αποφυγής ανίχνευσης (evasion) από σύγχρονα συστήματα ασφαλείας όπως antivirus (AV) ή Endpoint Detection and Response (EDR). Η επικοινωνία μεταξύ του εξυπηρετητή και του πράκτορα (agent) προστατεύεται μέσω κρυπτογράφησης AES σε επίπεδο εφαρμογής, χρησιμοποιώντας ένα προκαθορισμένο κοινό κλειδί. Αυτή η μέθοδος, αν και παρέχει εμπιστευτικότητα, στερείται των μηχανισμών αυθεντικοποίησης και της στιβαρότητας ενός πλήρους πρωτοκόλλου όπως το TLS, και η ασφάλειά της εξαρτάται από τη μυστικότητα του ενσωματωμένου κλειδιού. Η επιλογή αυτή έγινε για λόγους απλοποίησης της υλοποίησης και αντιμετώπισης τεχνικών δυσκολιών που προέκυψαν κατά την αρχική προσπάθεια χρήσης TLS.

Σε επίπεδο λειτουργικότητας, ο σχεδιασμός δεν περιλαμβάνει μηχανισμούς εδραίωσης παρουσίας, με αποτέλεσμα ο agent να μην εκκινεί αυτόματα μετά από επανεκκίνηση του συστήματος-στόχου. Η διαχείριση των συνδεδεμένων agents και των ανατιθέμενων εργασιών (tasks) παραμένει σε βασικό επίπεδο, βασιζόμενη σε δομές εντός μνήμης (in-memory), χωρίς τη χρήση βάσης δεδομένων για πιο ανθεκτική αποθήκευση. Βέβαια αυτό, μερικές φορές αποδεικνύεται χρήσιμο σε περιπτώσεις συλλογής πειστηρίων για ενοχοποίηση του χειριστή. Τέλος, η επιλογή χρήσης της εντολής `Invoke-Expression` στο Powershell για την εκτέλεση κώδικα, ενώ παρέχει λειτουργική ευελιξία στο πλαίσιο της επίδειξης, είναι γνωστό ότι εισάγει σημαντικούς κινδύνους ασφάλειας και αυξάνει δραστικά την πιθανότητα ανίχνευσης από μηχανισμούς παρακολούθησης, όπως κάποιο antivirus ή EDR.

Κατά συνέπεια, τονίζεται εμφατικά ότι η οποιαδήποτε χρήση του πλαισίου που αναπτύχθηκε περιορίζεται αυστηρά και αποκλειστικά σε ελεγχόμενα εργαστηριακά ή εκπαιδευτικά περιβάλλοντα, υπό την πλήρη συμμόρφωση με τους κανόνες ηθικής δεοντολογίας και με πλήρη επίγνωση των προαναφερθέντων περιορισμών.

1.8 Διάρθρωση της Εργασίας

Για την συστηματική και ολοκληρωμένη παρουσίαση της έρευνας, του σχεδιασμού, της υλοποίησης και των αποτελεσμάτων, η παρούσα πτυχιακή εργασία οργανώνεται σε έξι κεφάλαια συνολικά.

Στο τρέχον κεφάλαιο, γίνεται εισαγωγή στις πολύ βασικές έννοιες του αντικειμένου το οποίο μελετάται στην εργασία, θέτονται οι στόχοι και οι ευρύτεροι σκοποί της εργασίας, αναφέρονται μερικές τεχνικές λεπτομέρειες και περιορισμοί που προέκυψαν, έτσι ώστε να υπάρξει μια ομαλή μετάβαση στα επόμενα κεφάλαια. Έγινε προσπάθεια περιορισμού των υπερβολικά τεχνικών λεπτομερειών διότι το πρώτο κεφάλαιο οφείλει να διατηρήσει τον εισαγωγικό του χαρακτήρα. Εν συνεχεία, στο δεύτερο κεφάλαιο αναπτύσσεται το θεωρητικό υπόβαθρο του αντικειμένου της εργασίας, εμβαθύνοντας περαιτέρω στις θεωρητικές γνώσεις που κρίθηκαν απαραίτητες για την καλύτερη κατανόηση των πλαισίων διοίκησης και ελέγχου. Καλύπτεται η ιστορική εξέλιξη των πλαισίων C2, αναλύονται οι θεμελιώδεις αρχές λειτουργίας τους, οι κοινές αρχιτεκτονικές και τα κανάλια επικοινωνίας που χρησιμοποιούν, παρουσιάζονται παραδείγματα από πραγματικά συμβάντα κυβερνοεπιθέσεων ανά τον κόσμο και αναφέρονται μερικά από τα πιο αξιοσημείωτα C2 πλαίσια που υπάρχουν. Επιπλέον, δίνεται προσοχή στις τεχνικές μετά-εκμετάλλευσης με αναφορά στο μοντέλο MITRE ATT&CK [10] και στο τέλος, εξετάζονται οι σχετικές τεχνολογίες Powershell, οι μέθοδοι κρυπτογράφησης επικοινωνίας, καθώς και οι βασικές αρχές ανίχνευσης απειλών.

Προχωρόντας στο τρίτο κεφάλαιο, το επίκεντρο της προσοχής αποτελεί το πλαίσιο που αναπτύχθηκε καθαρά και μόνο για τις ανάγκες της εργασίας αυτής. Το τρίτο κεφάλαιο περιγράφει λεπτομερώς τις σχεδιαστικές αποφάσεις που λήφθηκαν, απαντά δηλαδή στο «γιατί» επιλέχθηκαν οι τεχνολογίες που απαρτίζουν το αναπτυχθέν εκπαιδευτικό πλαίσιο. Αναλύεται η αρχιτεκτονική του συστήματος, αιτιολογείται η επιλογή του πρωτοκόλλου επικοινωνίας που ορίστηκε και φυσικά, δείχνεται ο σχεδιασμός των στοιχείων που απαρτίζουν το πλαίσιο, τα οποία δεν είναι άλλα από τον server, τον agent και της διεπαφής ιστού.

Σειρά έχει το κεφάλαιο 4, στο οποίο παρουσιάζεται η πρακτική εφαρμογή και αξιολόγηση του πλαισίου που αναπτύχθηκε. Περιγράφεται αναλυτικά το περιβάλλον δοκιμών που χρησιμοποιήθηκε και επιδεικνύεται βήμα-βήμα η χρήση του πλαισίου C2 για την εκτέλεση συγκεκριμένων σεναρίων μετά-εκμετάλλευσης (συλλογή πληροφοριών, εκτέλεση εντολών), συνδέοντας τις ενέργειες αυτές με τις αντίστοιχες τεχνικές που περιγράφονται στο μοντέλο ATT&CK [10].

Το πρότελευταιο κεφάλαιο, προβαίνει σε μια κριτική ανάλυση της λειτουργικότητας του πλαισίου και των αποτελεσμάτων της μελέτης περίπτωσης που προηγήθηκε. Εδώ, ερμηνεύεται η σημασία των ευρημάτων, γίνεται εκτενής συζήτηση για τους περιορισμούς που εντοπίστηκαν κατά τον σχεδιασμό και την υλοποίηση του πλαισίου, θίγονται τα ηθικά ζητήματα που προκύπτουν από την ενασχόληση με τέτοια εργαλεία και τέλος, προτείνονται μερικές πιθανές κατευθύνσεις για μελλοντική εργασία και βελτιώσεις στο πλαίσιο. Το τελευταίο κεφάλαιο συνοψίζει τα βασικά σημεία της εργασίας, επαναδιατυπώνει τα κύρια ευρήματα και τη συνεισφορά της στην κατανόηση των C2 πλαισίων και των κινδύνων μετά-εκμετάλλευσης, και καταλήγει με ορισμένες τελικές παρατηρήσεις.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο: Διοίκηση, Έλεγχος και Τεχνικές Μετεκμετάλλευσης

2.1 Εισαγωγή

Εφόσον στο πρώτο κεφάλαιο ορίστηκε το πλαίσιο των σύγχρονων απειλών που εγκυμονεί ο κυβερνοχώρος, καθώς και ο κρίσιμος ρόλος των υποδομών διοίκησης και ελέγχου στην φάση της μετεκμετάλλευσης, το συγκεκριμένο κεφάλαιο πρόκειται να εμβαθύνει στο απαραίτητο θεωρητικό υπόβαθρο. Σκοπός του κεφαλαίου αυτού, είναι να παρέχει μια ολοκληρωμένη εικόνα των εννοιών, των τεχνολογιών και των τακτικών που σχετίζονται με τα C2 πλαίσια αλλά και τις ενέργειες μετεκμετάλλευσης. Θα αναφερθεί και θα εξεταστεί η ιστορική εξέλιξη αυτών των συστημάτων, οι θεμελιώδεις αρχές λειτουργίας τους, οι κοινές τους αρχιτεκτονικές και οι μέθοδοι επικοινωνίας που χρησιμοποιούν, αναφέροντας παράλληλα και μερικά πραγματικά συμβάντα που συνέβησαν ανά τα χρόνια. Έπειτα, θα αναλυθούν αρκετές από τις τεχνικές που συναντώνται στη φάση του post-exploitation όπως κατηγοριοποιούνται από το πλαίσιο MITRE ATT&CK. Μερικά παραδείγματα αυτών των τεχνικών αποτελούν τα εξής παρακάτω: η χρήση του Powershell ως επιθετικό εργαλείο, οι θεμελιώδεις αρχές της κρυπτογραφημένης επικοινωνίας (όπως η συμμετρική κρυπτογράφηση που αξιοποιείται στην παρούσα εργασία), αλλά και οι βασικές έννοιες ανίχνευσης και αποφυγής C2 (detection/evasion). Η κατανόηση των παραπάνω εννοιών είναι απαραίτητη για την πλήρη και σωστή ερμηνεία του σχεδιασμού και της λειτουργίας του εκπαιδευτικού περιβάλλοντος διοίκησης και ελέγχου που θα παρουσιαστεί στα επόμενα κεφάλαια.

Η αποτελεσματική διαχείριση μιας κυβερνοεπίθεσης, είτε από την πλευρά του επιτιθέμενου είτε του αμυνόμενου, προϋποθέτει βαθιά κατανόηση των μηχανισμών που διέπουν τη φάση αφότου προκύψει η αρχική παραβίαση. Όπως τονίστηκε στην εισαγωγή, οι υποδομές C2 αποτελούν τον ακρογωνιαίο λίθο αυτής της φάσης, επιτρέποντας τον απομακρυσμένο έλεγχο και την ενορχήστρωση περαιτέρω ενεργειών. Το παρόν κεφάλαιο αποσκοπεί στην οικοδόμηση μιας στέρεας θεωρητικής βάσης για την κατανόηση των πλαισίων C2 και των σχετικών τεχνικών post-exploitation. Ξεκινώντας από την ιστορική τους εξέλιξη, θα αναλυθούν οι βασικές αρχές, οι διαφορετικές αρχιτεκτονικές και τα κανάλια επικοινωνίας που χρησιμοποιούνται, αντλώντας παραδείγματα από σημαντικά κυβερνοσυμβάντα και γνωστά πλαίσια διοίκησης και ελέγχου. Στη συνέχεια, θα γίνει εστίαση στις τακτικές και τεχνικές που εφαρμόζονται κατά τη

μετά-εκμετάλλευση, χρησιμοποιώντας το μοντέλο MITRE ATT&CK ως οδηγό, και θα εξεταστούν συγκεκριμένες τεχνολογίες όπως το Powershell και οι μέθοδοι κρυπτογράφησης επικοινωνίας που παίζουν καθοριστικό ρόλο. Τέλος, θα υπάρξει ενασχόληση με τις βασικές αρχές πίσω από την ανίχνευση και την αποφυγή αυτών των μηχανισμών.

2.2 Ιστορική Εξέλιξη των C2 Frameworks

Η έννοια της απομακρυσμένης διαχείρισης και ελέγχου υπολογιστικών συστημάτων δεν είναι καινούρια. Ωστόσο, η εξέλιξη των υποδομών διοίκησης και ελέγχου όπως είναι γνωστές στο πλαίσιο της κυβερνοασφάλειας αποτελεί μια συναρπαστική και αξιοσημείωτη ιστορία προσαρμογής, καινοτομίας και ενός διαρκή αγώνα μεταξύ επιτιθέμενων και αμυνόμενων. Η πορεία από απλά, μεμονωμένα εργαλεία απομακρυσμένης πρόσβασης σε πολύπλοκα, αρθρωτά και συχνά δυσδιάκριτα πλαίσια C2 αντικατοπτρίζει τις ευρύτερες αλλαγές στην τεχνολογία δικτύων, στα λειτουργικά συστήματα και στις αμυντικές στρατηγικές. Η κατανόηση αυτής της εξέλιξης παρέχει πολύτιμο χώρο για την ανάλυση των σύγχρονων C2 frameworks και των τεχνικών που χρησιμοποιούν. Αξίζει να αναφερθεί πως η ιστορική εξέλιξη των πλαισίων C2 έχει διαμορφωθεί σημαντικά τόσο στον στρατιωτικό χώρο όσο και στον κυβερνοχώρο, επηρεαζόμενη από τις τεχνολογικές εξελίξεις και τις μεταβαλλόμενες επιχειρησιακές ανάγκες. Η κατανόηση αυτής της διπλής φύσης είναι ουσιαστική, καθώς οι καινοτομίες σε έναν τομέα συχνά επηρεάζουν τις εξελίξεις στον άλλο [12].

2.2.1 Οι Πρόδρομοι και οι Πρώιμες Μορφές (Δεκαετίες 1980 - 1990)

Οι ρίζες των πλαισίων C2 μπορούν να ανιχνευθούν στα πρώιμα εργαλεία απομακρυσμένης διαχείρισης και στα πρώτα κακόβουλα προγράμματα τύπου «Δούρειος Ίππος» (Trojan Horse). Εργαλεία όπως το telnet [13] ή απλά scripts που επέτρεπαν την εκτέλεση εντολών σε έναν απομακρυσμένο υπολογιστή μέσω δικτύου αποτέλεσαν τους εννοιολογικούς προδρόμους. Τα πρώτα Trojans, αν και συχνά σχεδιασμένα για απλή καταστροφή ή ενόχληση, σταδιακά άρχισαν να ενσωματώνουν λειτουργίες «κερκόπορτας» (backdoor), επιτρέποντας στον δημιουργό τους να επανασυνδεθεί στο μολυσμένο σύστημα. Χαρακτηριστικά παραδείγματα αυτής της περιόδου περιλαμβάνουν εργαλεία όπως το Back Orifice (κυκλοφόρησε το 1998) [14], το οποίο παρείχε εκτεταμένες δυνατότητες απομακρυσμένου ελέγχου σε συστήματα Windows, συχνά μέσω μιας γραφικής διεπαφής. Η επικοινωνία σε αυτές τις πρώιμες μορφές ήταν συνήθως απευθείας μεταξύ του επιτιθέμενου και του θύματος, συχνά μέσω συγκεκριμένων TCP ή UDP πορτών, και τις περισσότερες φορές χωρίς κρυπτογράφηση, καθιστώντας την σχετικά εύκολη στην ανίχνευση, στην περίπτωση που παρακολουθούνταν η δικτυακή κίνηση. Ο έλεγχος αφορούσε κυρίως μεμονωμένους στόχους.

2.2.2 Η Εποχή των Botnets και του IRC (Αρχές - Μέσα δεκαετίας 2000)

Η διάδοση του διαδικτύου και η αύξηση των ευπαθειών σε ευρέως χρησιμοποιούμενα λειτουργικά συστήματα όπως τα Windows XP οδήγησαν στην εμφάνιση και ραγδαία εξάπλωση των botnets. Ένα botnet

είναι ένα δίκτυο από παραβιασμένους υπολογιστές (bots) που ελέγχονται κεντρικά από έναν χειριστή (botmaster). Η ανάγκη για διαχείριση εκατοντάδων ή χιλιάδων bots ταυτόχρονα οδήγησε στην υιοθέτηση υπαρχόντων πρωτοκόλλων επικοινωνίας για τον συντονισμό. Το πρωτόκολλο Internet Relay Chat (IRC) [15] αναδείχθηκε ως η ντε φάκτο πλατφόρμα C2 για πολλά από τα πρώτα μεγάλα botnets (βλ. Agobot/Gaobot, SDBot, SpyBot) [16], [17], [18], [19]. Οι λόγοι ήταν πρακτικοί: το IRC ήταν ένα καθιερωμένο, απλό, βασισμένο σε κείμενο πρωτόκολλο, υπήρχαν ήδη πολλοί δημόσιοι IRC servers, και η δομή των καναλιών επέτρεπε στον botmaster να στέλνει εντολές μαζικά σε όλα τα bots που είχαν συνδεθεί σε ένα συγκεκριμένο κανάλι. Τα bots συνδέονταν σε έναν προκαθορισμένο IRC server και κανάλι, περίμεναν εντολές (π.χ. για αποστολή spam, εκτέλεση DDoS επιθέσεων) και μερικές φορές έστελναν πίσω απλές αναφορές κατάστασης. Αν και αποτελεσματικό για μαζικό έλεγχο, το IRC C2 είχε μειονεκτήματα: η επικοινωνία ήταν συχνά ακρυπτογράφητη και η χρήση μη τυπικών ports (δηλαδή ό,τιδήποτε πέρα από το εύρος 0 έως 1023 που δεν αντιστοιχεί σε κάποια γνωστή υπηρεσία) για το IRC μπορούσε να μπλοκαριστεί εύκολα από firewalls. Επίσης, η εξάρτηση από δημόσιους IRC εξυπηρετητές τους καθιστούσε ευάλωτους σε takedowns, δηλαδή κατασχέσεις ή αλλιώς «ρίξιμο» [18], [20], [21].

2.2.3 Η Μετάβαση στα Πρωτόκολλα Ιστού (Μέσα δεκαετίας 2000 - Αρχές δεκαετίας 2010)

Με την πάροδο των χρόνων, καθώς οι διαχειριστές δικτύων άρχισαν να μπλοκάρουν πιο συστηματικά τα ports που χρησιμοποιούσε το IRC και άλλα μη-τυπικά πρωτόκολλα, οι δημιουργοί κακόβουλου λογισμικού αναζήτησαν πιο συγκεκαλυμμένες μεθόδους επικοινωνίας. Τα πρωτόκολλα του παγκόσμιου ιστού, HTTP (πόρτα 80) και αργότερα HTTPS (πόρτα 443), αποτέλεσαν την ιδανική λύση. Αυτά τα ports είναι σχεδόν πάντα ανοικτά στα firewalls για να επιτρέπουν την περιήγηση στο διαδίκτυο, καθιστώντας την C2 κίνηση που τα χρησιμοποιεί πολύ πιο δύσκολο να διακριθεί από την κανονική ή «νόμιμη» δικτυακή κίνηση. Τα C2 frameworks άρχισαν να υιοθετούν ένα μοντέλο «beaconing», όπου ο agent περιοδικά «καλεί σπίτι» (beacons ή «φάροι») στον C2 server στέλνοντας ένα HTTP GET ή POST request, όπου η απάντηση του server μπορεί να περιείχε τυχόν νέες εντολές προς εκτέλεση από τον agent. Δεδομένα που αποστέλλονταν από τον agent στον server (δηλαδή, κλεμμένα διαπιστευτήρια, πληροφορίες συστήματος) συχνά κωδικοποιούνταν σε κάποια κωδικοποίηση όπως είναι και η base64 [22] και ενσωματώνονταν στο σώμα ενός POST request ή ακόμα και στα URL parameters ή HTTP headers. Η υιοθέτηση του HTTPS πρόσθεσε ένα επίπεδο κρυπτογράφησης (TLS/SSL), καθιστώντας την επιθεώρηση του περιεχομένου της επικοινωνίας πολύ πιο δύσκολη για τους αμυνόμενους. Εμβληματικά παραδείγματα κακόβουλου λογισμικού που χρησιμοποίησαν εκτενώς HTTP/HTTPS C2 είναι τα τραπεζικά Trojans όπως το Zeus [23] και το SpyEye [24], τα οποία είχαν εξαιρετικά εξελιγμένες δυνατότητες C2 για την εποχή τους [25], [26].

2.2.4 Προηγμένες Επίμονες Απειλές (APTs) και Εξειδίκευση (Τέλη δεκαετίας 2000 - Σήμερα)

Η εμφάνιση των Προηγμένων Επίμονων Απειλών (Advanced Persistent Threats - APTs), συχνά συνδεδεμένων με κρατικούς φορείς, οδήγησε την εξέλιξη των C2 σε νέα, πρωτόγνωρα επίπεδα πολυπλοκότητας, με κύριο γνώμονα την απόλυτη μυστικότητα και την ανθεκτικότητα. Οι ομάδες πίσω από τα APT συχνά αναπτύσσουν τα δικά τους, κατά παραγγελία (bespoke) C2 πλαίσια και πρωτόκολλα έτσι ώστε να είναι

απρόβλεπτα και δύσκολα στην ανίχνευση.

Την περίοδο αυτή, παρατηρήθηκαν μερικές νέες τάσεις όπως τα προσαρμοσμένα πρωτόκολλα, ισχυρή κρυπτογράφηση, χρήση νόμιμων υπηρεσιών, ανθεκτικότητα και στοχευμένη λειτουργικότητα. Για τα προσαρμοσμένα πρωτόκολλα, γίνεται χρήση «δυαδικών» (binary) πρωτοκόλλων αντί του απλού HTTP, καθιστώντας την ανάλυση της κίνησης από κάποιον συλλέκτη πακέτων, όπως το Wireshark, αισθητά δυσκολότερη, χωρίς προηγούμενη γνώση και κατανόηση της δομής του πρωτοκόλλου εν δράσει. Ένα δυαδικό πρωτόκολλο είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιεί δυαδική κωδικοποίηση δεδομένων για τη μετάδοση δεδομένων μέσω ενός δικτύου. Αντίθετα με τα πρωτόκολλα που βασίζονται σε κείμενο (όπως το HTTP), τα οποία χρησιμοποιούν αναγνώσιμες μορφές κειμένου (όπως το κείμενο ASCII) για την επικοινωνία, τα δυαδικά πρωτόκολλα αναπαριστούν τα δεδομένα σε δυαδική μορφή. Αυτή η δυαδική μορφή είναι πιο αποδοτική για τους υπολογιστές να την αναλύουν και να την επεξεργάζονται [27]. Όσον αφορά την εφαρμογή ισχυρής κρυπτογράφησης, πέρα από το TLS, παρατηρείται χρήση ισχυρών αλγορίθμων κρυπτογράφησης με σωστή διαχείριση και ανταλλαγή κλειδιών [28].

Επιπλέον, φαίνεται να αξιοποιούνται έξυπνα διάφορες νόμιμες και δημοφιλείς διαδικτυακές υπηρεσίες για την απόκρυψη της κίνησης και κατ' επέκταση επικοινωνίας C2. Για παράδειγμα, οι κακόβουλοι φορείς πραγματοποιούν DNS tunneling, δηλαδή την ενσωμάτωση έξτρα δεδομένων σε DNS ερωτήματα ή απαντήσεις[29], [30], [31]. Άλλα παραδείγματα περιλαμβάνουν τη χρήση πολυσύχναστων πλατφορμών κοινωνικής δικτύωσης, όπως το Twitter (πλέον X), το Facebook (πλέον Meta), το Discord και το Github (*ναι, το Github θεωρείται κι αυτό κοινωνικό δίκτυο πλέον*), υπηρεσιών νέφους όπως το Dropbox και το Google Drive, ή ακόμα και υπηρεσιών ηλεκτρονικού ταχυδρομείου, για την ανταλλαγή εντολών και δεδομένων [28].

Συνεχίζοντας, η ανθεκτικότητα έγκειται στην υλοποίηση πολλαπλών C2 servers, μηχανισμών fallback σε περίπτωση που ο κύριος server τεθεί εκτός λειτουργίας, και τέλος, χρήση αλγορίθμων παραγωγής ονομάτων τομέα (Domain Generation Algorithms- DGAs) για τη δυναμική εύρεση ενεργών C2 εξυπηρετητών. Ακόμη, η στοχευμένη λειτουργικότητα αφορά τη δημιουργία και χρήση εξειδικευμένων (custom) πλαισίων C2, κατασκευασμένα καθαρά και μόνο για πολύ συγκεκριμένους σκοπούς όπως κατασκοπεία ή σαμποτάζ. Σε αυτή την κατηγορία εντάσσεται το πασίγνωστο Stuxnet [32], [33] καθώς και τα Flame [34] και Duqu [35], τα οποία απέδειξαν τις χαοτικές δυνατότητες των εξειδικευμένων C2 πλαισίων και υπήρξαν το επίκεντρο πολύωρων ερευνών μέχρι και σήμερα.

2.2.5 Εμπορευματοποίηση και Πολλαπλασιασμός Πλαισίων (Δεκαετία 2010 - Σήμερα)

Παράλληλα με την εξέλιξη των APTs, η δεκαετία του 2010 είδε την άνοδο της βιομηχανίας επιθετικής ασφάλειας (penetration testing, red teaming) και την εμπορευματοποίηση ή διάθεση ως open-source ισχυρών C2 πλαισίων. Αυτά τα εργαλεία, αν και πολλές φορές αναπτύχθηκαν για νόμιμους σκοπούς (προσομοίωση επιθέσεων για βελτίωση της άμυνας), αναπόφευκτα υιοθετήθηκαν και από κακόβουλους παράγοντες, μειώνοντας σημαντικά το τεχνικό εμπόδιο για την πραγματοποίηση εξελιγμένων επιθέσεων. Κύρια χαρακτηριστικά αυτής της εποχής περιλαμβάνουν:

- Αρθρωτά (Modular) Frameworks: Εργαλεία όπως το Metasploit Framework (με το Meterpreter payload) [36], το Cobalt Strike [37], το Empire (βασισμένο σε Powershell) [38], και πιο πρόσφατα

το Sliver [39], προσφέρουν μια κεντρική κονσόλα διαχείρισης και μια πληθώρα από modules για διάφορες ενέργειες post-exploitation όπως, για παράδειγμα: συλλογή διαπιστευτηρίων, πλευρική κίνηση, καταγραφή οθόνης.

- Ευκολία Χρήσης: Συχνά διαθέτουν γραφικές διεπαφές χρήστη (GUI) και εκτενές documentation.

- Προσαρμοστικότητα: Πλαίσια όπως το Cobalt Strike εισήγαγαν την έννοια των «Malleable C2 Profiles» [40], επιτρέποντας στους χρήστες να προσαρμόσουν τα χαρακτηριστικά της δικτυακής κίνησης του C2 ώστε να μιμούνται νόμιμο λογισμικό ή συγκεκριμένους τύπους απειλών, καθιστώντας την ανίχνευση βάσει υπογραφών πιο δύσκολη.

- Διττή Χρήση (Dual-Use): Η ίδια η ισχύς και ευελιξία αυτών των εργαλείων τα καθιστά ελκυστικά τόσο για τους «καλούς» (red teamers) όσο και για τους «κακούς» της υπόθεσης [41], [42].

2.2.6 Σύγχρονες Τάσεις και Μελλοντικές Κατευθύνσεις

Στη σήμερον ημέρα, η εξέλιξη των πλαισίων C2 συνεχίζεται αμείωτα, προσαρμοζόμενη στις νέες τεχνολογίες και στις αμυντικές στρατηγικές. Ξεκινώντας από τα πλαίσια C2 στο «σύννεφο», παρατηρείται αυξανόμενη χρήση νόμιμων υπηρεσιών cloud σαν το Azure Functions, AWS Lambda ή Google Cloud Platform για τη φιλοξενία C2 υποδομών, καθιστώντας τον εντοπισμό και το blacklisting από διάφορους φορείς, πιο δύσκολο [30]. Όσον αφορά τη χρήση Content Delivery Networks (CDNs), αξιοποιούνται CDNs για την απόκρυψη της πραγματικής διεύθυνσης IP του C2 εξυπηρετητή, καθώς και για domain fronting (αν και αυτή η τεχνική γίνεται πιο δύσκολη) [43]. Σαν να μην έφταναν τα παραπάνω, οι κακόβουλοι φορείς πειραματίζονται με λιγότερο κοινά κανάλια επικοινωνίας, όπως μηνύματα σε πλατφόρμες gaming όπως για παράδειγμα το Discord, ή ενσωμάτωση σε λιγότερο παρακολουθούμενα πρωτόκολλα [44]. Επιπλέον, έμφαση δίνεται στα «Living Off The Land» Binaries and Scripts (LOLBAS). Αυτό σημαίνει πως γίνεται χρήση ενσωματωμένων εργαλείων του λειτουργικού συστήματος για την εκτέλεση κακόβουλων ενεργειών, μειώνοντας την ανάγκη για εναπόθεση εξωτερικών αρχείων και κάνοντας την ανίχνευση βάσει αρχείων λιγότερο αποτελεσματική [45], [46]. Τέλος, στο κοντινό μέλλον, ίσως ανακαλυφθούν πλαίσια C2 που χρησιμοποιούν τεχνητή νοημοσύνη για πιο αυτόνομη λήψη αποφάσεων, βελτιστοποίηση της επικοινωνίας ή προσαρμογή της συμπεριφοράς για αποφυγή ανίχνευσης [47], [48].

Συνοψίζοντας, η ιστορία των πλαισίων C2 είναι μια διαρκής διαδικασία προσαρμογής. Από απλά scripts απομακρυσμένης εκτέλεσης, εξελίχθηκαν σε μαζικά συστήματα διαχείρισης botnets μέσω του IRC και έπειτα, μετατοπίστηκαν στα πανταχού παρόντα πρωτόκολλα ιστού για κάλυψη, εξειδικεύτηκαν για τις ανάγκες των APTs, και τελικά έγιναν διαθέσιμα στο ευρύ κοινό μέσω ισχυρών εμπορικών και ανοικτού κώδικα πλαισίων. Οπότε, η υψηλή σημασία αυτής της πορείας έγκειται στην χρήση της για την καλύτερη εκτίμηση της τρέχουσας κατάστασης που επικρατεί στον χώρο της κυβερνοασφάλειας, όπως και για την πρόβλεψη μελλοντικών εξελίξεων σε αυτό τον κρίσιμο τομέα της κυβερνοασφάλειας οπου κατατάσονται τα πλαίσια C2.

2.3 Θεμελιώδεις Αρχές, Αρχιτεκτονικές και Κανάλια Επικοινωνίας C2

Μετά την ιστορική αναδρομή στην εξέλιξη των C2 frameworks στην προηγούμενη ενότητα, είναι απαραίτητο να εξεταστούν οι θεμελιώδεις αρχές που διέπουν τη λειτουργία τους, ανεξάρτητα από τη συγκεκριμένη εποχή ή υλοποίηση. Η κατανόηση του σκοπού, κύκλου ζωής, αρχιτεκτονικών δομών και καναλιών επικοινωνίας είναι θεμελιώδης για την ανάλυση οποιουδήποτε συστήματος C2, συμπεριλαμβανομένου του εκπαιδευτικού πλαισίου της εργασίας.

2.3.1 Σκοπός και Κύκλος Ζωής

Ο πρωταρχικός σκοπός μιας υποδομής διοίκησης και ελέγχου είναι να παρέχει στον επιτιθέμενο (ή χειριστή) έναν αξιόπιστο και ιδανικά, συγκεκαλυμμένο μηχανισμό για να αλληλεπιδρά με τα παραβιασμένα συστήματα μετά την αρχική «μόλυνση». Οι επιμέρους σκοποί που εξυπηρετεί ένα πλαίσιο C2 περιλαμβάνουν τους εξής παρακάτω:

- **Διατήρηση Πρόσβασης (Maintaining Access)**: Η εξασφάλιση της δυνατότητας συνεχούς ή περιοδικής επικοινωνίας με το παραβιασμένο σύστημα, συχνά μέσω μηχανισμών εδραίωσης επίμονης παρουσίας (persistence).
- **Απομακρυσμένη Εκτέλεση (Remote Execution)**: Η δυνατότητα αποστολής εντολών, scripts ή εκτελέσιμων αρχείων προς εκτέλεση στο σύστημα-στόχο.
- **Συλλογή Δεδομένων (Data Collection)**: Η συγκέντρωση πληροφοριών από το παραβιασμένο σύστημα. Για παράδειγμα, πληροφορίες συστήματος, διαπιστευτήρια, αρχεία χρήστη.
- **Υποκλοπή Δεδομένων (Data Exfiltration)**: Η ασφαλής μεταφορά των συλλεγμένων δεδομένων από το παραβιασμένο σύστημα πίσω στον C2 server.
- **Διαχείριση Πολλαπλών Agents (Management)**: Ο συντονισμός και η διαχείριση ενός δικτύου από παραβιασμένα συστήματα (botnet).
- **Εγκατάσταση «Ωφέλιμου Φορτίου» (Payload Deployment)**: Η χρήση του υπάρχοντος καναλιού C2 για τη λήψη και εγκατάσταση επιπρόσθετου κακόβουλου λογισμικού όπως ransomware ή/ και keyloggers.
- **Πλατφόρμα για Post-Exploitation**: Η παροχή μιας διαδραστικής ή αυτοματοποιημένης πλατφόρμας για την εκτέλεση διαφόρων τακτικών μετά-εκμετάλλευσης όπως lateral movement και privilege escalation.

Ο τυπικός κύκλος ζωής ενός C2 agent, από τη στιγμή της αρχικής εκτέλεσης στο σύστημα-στόχο, περιλαμβάνει συνήθως τα ακόλουθα στάδια:

1. **Εκκίνηση/Εγκατάσταση (Initialization/Installation)**: Ο «πράκτορας» (agent) εκτελείται για πρώτη φορά, είτε μέσω εκμετάλλευσης ευπάθειας, είτε μέσω κοινωνικής μηχανικής (social engineering),

είτε ως μέρος ενός άλλου κακόβουλου λογισμικού. Μπορεί να προσπαθήσει να εγκαταστήσει μηχανισμούς εδραίωσης παρουσίας.

2. **Επικοινωνία/Καταχώρηση (Beaconing/Registration)**: Ο agent προσπαθεί να επικοινωνήσει με τον προκαθορισμένο C2 server. Αυτή η πρώτη επικοινωνία (beacon) συχνά περιλαμβάνει βασικές πληροφορίες για το σύστημα (όνομα υπολογιστή, όνομα χρήστη, έκδοση ΛΣ) ώστε ο server να το καταχωρήσει και να του αποδώσει ένα μοναδικό αναγνωριστικό. Όπως θα γίνει φανερό σε επόμενο κεφάλαιο, στο εκπαιδευτικό πλαίσιο που αναπτύχθηκε για τις ανάγκες της εργασίας, αυτό αντιστοιχεί στο μήνυμα SYN.

3. **Λήψη Εργασιών (Tasking)**: Μετά την επιτυχή επικοινωνία, ο agent ζητά (ή λαμβάνει περιοδικά) οδηγίες από τον C2 server. Ο server απαντά είτε με μια συγκεκριμένη εντολή προς εκτέλεση, είτε με μια εντολή αναμονής.

4. **Εκτέλεση Εργασίας (Execution)**: Εάν ο agent λάβει μια εντολή, προσπαθεί να την εκτελέσει χρησιμοποιώντας τους διαθέσιμους πόρους του λειτουργικού συστήματος.

5. **Αναφορά Αποτελεσμάτων (Result Reporting)**: Μετά την εκτέλεση ή αν η εκτέλεση αποτύχει, ο agent συλλέγει την έξοδο ή την κατάσταση σφάλματος και την προετοιμάζει για αποστολή πίσω στον C2 server κατά την επόμενη προγραμματισμένη επικοινωνία.

6. **Επανάληψη (Looping)**: Ο κύκλος *Λήψη Εργασιών, Εκτέλεση, Αναφορά Αποτελεσμάτων* επαναλαμβάνεται, συχνά με περιόδους αδράνειας μεταξύ των επικοινωνιών για μείωση του «θορύβου» δικτύου και για αποφυγή ανίχνευσης.

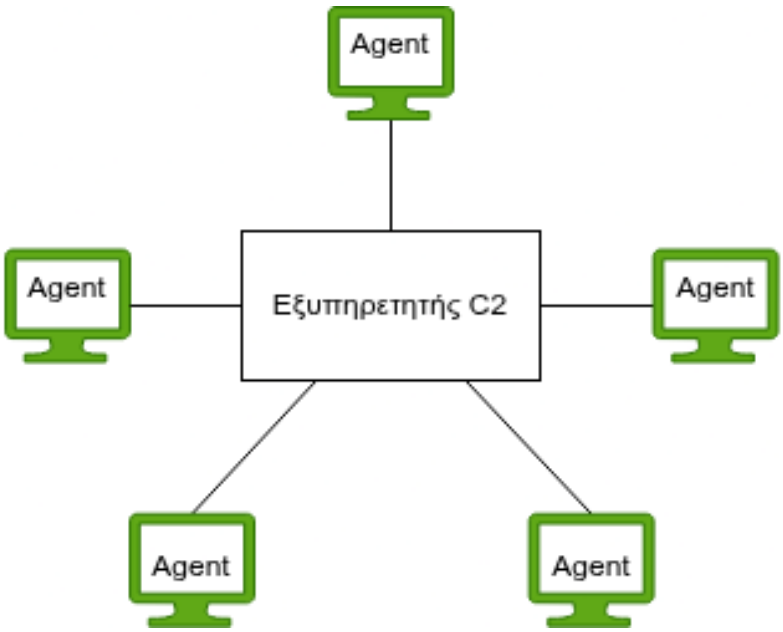
7. **(Προαιρετικά) Ενημέρωση/Τερματισμός (Update/Termination)**: Ο C2 server μπορεί να στείλει εντολές για την ενημέρωση του agent με νεότερη έκδοση ή για τον πλήρη τερματισμό και αυτοδιαγραφή του από το σύστημα. Αρκετα χρήσιμο για την εξαφάνιση στοιχείων/ πειστηρίων.

2.3.2 Κοινές Αρχιτεκτονικές C2

Τα πλαίσια διοίκησης και ελέγχου δεν υιοθετούν όλα την ίδια δομική προσέγγιση. Η αρχιτεκτονική ενός C2 παίζει καθοριστικό ρόλο, επηρεάζοντας σημαντικά παραμέτρους όπως την ανθεκτικότητά του έναντι προσπαθειών κατάργησης (takedown), την ταχύτητα με την οποία οι εντολές διαδίδονται στους παραβιασμένους agents, την πολυπλοκότητα της διαχείρισής του, καθώς και την ευκολία με την οποία μπορεί να ανιχνευθεί. Διακρίνονται τρεις κυρίαρχες αρχιτεκτονικές τάσεις.

Η πιο κλασική και εννοιολογικά απλή είναι η **κεντροποιημένη αρχιτεκτονική**, συχνά αναφερόμενη και ως **τοπολογία αστέρα (star topology)**. Σε αυτό το μοντέλο, όλοι οι agents επικοινωνούν απευθείας με έναν ή παραπάνω κεντρικούς C2 εξυπηρετητές. Οι εξυπηρετητές αυτοί λειτουργούν ως το κεντρικό νευρικό σύστημα, συγκεντρώνοντας δεδομένα και διανέμοντας εντολές. Το εκπαιδευτικό πλαίσιο που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας υιοθετεί ακριβώς αυτήν την κεντροποιημένη δομή. Τα πλεονεκτήματα μιας τέτοιας προσέγγισης έγκεινται στην απλότητα του σχεδιασμού και της διαχείρισης, στον άμεσο έλεγχο που παρέχεται στον χειριστή επί των agents, και στην ταχεία διάδοση των εντολών. Ωστόσο, παρουσιάζει ένα σημαντικό μειονέκτημα, καθώς συνιστά ένα μοναδικό σημείο αποτυχίας (Single Point of Failure - SPoF). Εάν οι διευθύνσεις IP ή τα ονόματα τομέα (domains) των κεντρικών

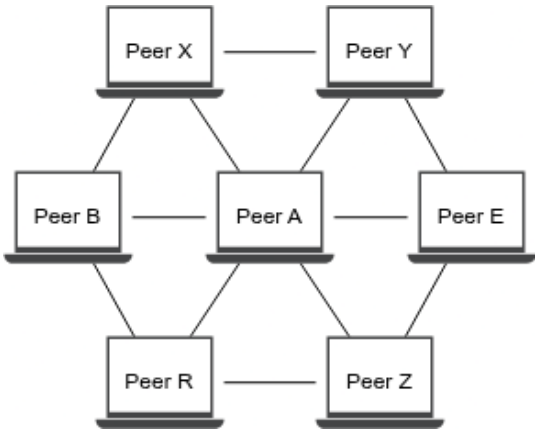
C2 εξυπηρετητών αναγνωριστούν και μπλοκαριστούν, ολόκληρο το δίκτυο των παραβιασμένων συστημάτων (botnet) μπορεί να καταστεί ανενεργό. Επιπλέον, η συγκεντρωμένη δικτυακή κίνηση προς λίγους κεντρικούς εξυπηρετητές είναι ενδεχομένως πιο εύκολα ανιχνεύσιμη.



Σχήμα 2.1: Τοπολογία Αστέρα

Μια διαφορετική προσέγγιση υιοθετείται στην **ομότιμη (Peer-to-Peer - P2P) αρχιτεκτονική**. Πρόκειται για ένα αποκεντρωμένο μοντέλο όπου απουσιάζει ο κεντρικός εξυπηρετητής. Αντ' αυτού, οι ίδιοι οι agents, λειτουργώντας ως ομότιμοι κόμβοι (peers), επικοινωνούν απευθείας μεταξύ τους, σχηματίζοντας ένα αυτο-οργανωμένο δίκτυο. Οι εντολές και τα δεδομένα διαχέονται από κόμβο σε κόμβο εντός αυτού του P2P δικτύου, και ο χειριστής μπορεί να εισάγει εντολές συνδεόμενος σε οποιονδήποτε από τους συμμετέχοντες κόμβους. Το κύριο πλεονέκτημα αυτής της δομής είναι η εξαιρετική της ανθεκτικότητα σε προσπάθειες κατάργησης, δεδομένου ότι δεν υφίσταται κεντρικό σημείο που να μπορεί να στοχευθεί και να εξουδετερωθεί. Παράλληλα, καθίσταται δυσκολότερος ο εντοπισμός του συνόλου του δικτύου. Αντίθετα, η υλοποίηση και διαχείριση μιας P2P αρχιτεκτονικής είναι σημαντικά πιο πολύπλοκη, και η διάδοση των εντολών ενδέχεται να είναι πιο αργή. Ένα επιπλέον πιθανό μειονέκτημα είναι η δημιουργία σημαντικού όγκου δικτυακής κίνησης μεταξύ των agents, η οποία μπορεί να γίνει αντιληπτή, ιδίως εντός ενός εσωτερικού, παρακολουθούμενου δικτύου. Ιστορικά παραδείγματα botnets που αξιοποίησαν P2P αρχιτεκτονικές περιλαμβάνουν το Slime και το Storm Worm [49], [50].

Τέλος, η **αρχιτεκτονική πολλαπλών επιπέδων ή υβριδική (Multi-Tier / Hybrid)** επιδιώκει να συνδυάσει στοιχεία από τις προαναφερθείσες δομές. Μια συνήθης υλοποίηση περιλαμβάνει τη χρήση ενδιάμεσων κόμβων, οι οποίοι μπορεί να είναι είτε ειδικοί διακομιστές μεσολάβησης (proxies/relays) είτε άλλα παραβιασμένα συστήματα που λειτουργούν ως ενδιάμεσα επίπεδα (tiers). Οι τελικοί agents συνδέονται με αυτούς τους ενδιάμεσους κόμβους, οι οποίοι με τη σειρά τους δρομολογούν την επικοινωνία προς τον κύριο, συχνά πιο κρυφό, C2 εξυπηρετητή. Τα πλεονεκτήματα αυτής της μεθόδου περιλαμβάνουν την απόκρυψη της πραγματικής τοποθεσίας του κεντρικού C2 εξυπηρετητή και την αυξημένη ανθεκτικότητα, καθώς η απώλεια ενός ενδιάμεσου κόμβου δεν συνεπάγεται απαραίτητα την απώλεια ολόκληρου του δικτύου. Επιπλέον, μπορεί να χρησιμοποιηθεί για γεωγραφική κατανομή των ενδιάμεσων κόμβων ή



Σχήμα 2.2: Τοπολογία P2P

για εξισορρόπηση του φόρτου επικοινωνίας. Ωστόσο, η διαχείριση μιας τέτοιας πολυεπίπεδης υποδομής είναι πιο πολύπλοκη και μπορεί να εισάγει καθυστερήσεις στην επικοινωνία.

Αξίζει να σημειωθεί ότι η τεχνική της περιοδικής επικοινωνίας (beaconing), όπου ο agent εκκινεί την επικοινωνία προς τον C2 server σε τακτά ή ακανόνιστα χρονικά διαστήματα, δεν συνιστά από μόνη της μια διακριτή αρχιτεκτονική. Αντίθετα, αποτελεί ένα μοτίβο συμπεριφοράς που εφαρμόζεται κατά κόρον σε κεντροποιημένες ή υβριδικές αρχιτεκτονικές. Ο agent παραμένει αδρανής («κοιμάται») για ένα ορισμένο χρονικό διάστημα, εν συνεχεία αφυπνίζεται για να συνδεθεί στον C2 εξυπηρετητή, να αποστείλει ή να λάβει δεδομένα και εντολές, και κατόπιν επανέρχεται σε κατάσταση αδράνειας. Αυτή η προσέγγιση δυσχεραίνει την παθητική ανίχνευση του C2 εξυπηρετητή και μπορεί να συμβάλει στην παράκαμψη τειχών προστασίας, αν και συνεπάγεται μια εγγενή καθυστέρηση στην εκτέλεση των εντολών [51], [52].

2.3.3 Κοινά Κανάλια Επικοινωνίας C2

Η επιλογή του καναλιού επικοινωνίας, δηλαδή του συγκεκριμένου πρωτοκόλλου και της θύρας που θα χρησιμοποιηθεί, αποτελεί μια απόφαση κρίσιμης σημασίας για την επιτυχία ενός πλαισίου διοίκησης και ελέγχου. Η επιλογή αυτή επηρεάζει άμεσα την ικανότητα του C2 να παρακάμπτει τα υφιστάμενα μέτρα ασφαλείας, όπως τα τείχη προστασίας (firewalls), τα συστήματα ανίχνευσης και αποτροπής εισβολών (IDS/IPS) και τους διακομιστές μεσολάβησης (proxies), καθώς και τη δυνατότητά του να παραμένει απαρατήρητο εντός της ευρύτερης δικτυακής κίνησης.

Μεταξύ των πιο διαδεδομένων καναλιών που αξιοποιούνται για την C2 επικοινωνία είναι τα πρωτόκολλα **HTTP και HTTPS**, τα οποία λειτουργούν συνήθως στις θύρες 80 και 443 αντίστοιχα. Ο μηχανισμός περιλαμβάνει την ενσωμάτωση της C2 επικοινωνίας σε τυπικά αιτήματα HTTP GET ή POST. Τα δεδομένα μπορούν να κωδικοποιηθούν και να μεταφερθούν μέσω παραμέτρων URL, κεφαλίδων HTTP (όπως οι User-Agent ή Cookies), ή απευθείας στο σώμα του αιτήματος ή της απάντησης. Η χρήση του HTTPS προσθέτει το πλεονέκτημα της κρυπτογράφησης μέσω TLS. Το κύριο πλεονέκτημα αυτής της μεθόδου είναι η εξαιρετική της ικανότητα να «αναμειγνύεται» (blending in) με τη νόμιμη κίνηση του παγκόσμιου ιστού, καθώς οι θύρες 80 και 443 είναι σχεδόν πάντοτε ανοιχτές στα περισσότερα τείχη προστασίας.

Επιπλέον, το TLS παρέχει ισχυρή και τυποποιημένη κρυπτογράφηση. Ωστόσο, η κίνηση αυτή δεν είναι απρόσβλητη στην ανίχνευση μιας και μπορεί να εντοπιστεί από web proxies και συστήματα ασφαλείας που αναλύουν τις κεφαλίδες HTTP, τα μοτίβα της κίνησης (όπως η σταθερή περιοδικότητα των beacons), τον όγκο των δεδομένων, ή τη φήμη του domain name ή της διεύθυνσης IP του C2 εξυπηρετητή. Η επιθεώρηση της κρυπτογραφημένης κίνησης TLS (TLS inspection) από ενδιάμεσες συσκευές ασφαλείας μπορεί, αν και με δυσκολία, να αποκαλύψει την κακόβουλη φύση της επικοινωνίας [51], [52].

Ένα άλλο συχνά χρησιμοποιούμενο κανάλι είναι το πρωτόκολλο **DNS (Domain Name System)**, που λειτουργεί τυπικά στη θύρα 53 μέσω UDP ή TCP. Σε αυτή την περίπτωση, η C2 επικοινωνία ενθυλακώνεται (tunnels) εντός των ερωτήσεων και απαντήσεων DNS. Για παράδειγμα, δεδομένα που προορίζονται για τον εξυπηρετητή C2 μπορούν να κωδικοποιηθούν ως υποτομείς (subdomains) σε μια DNS ερώτηση (π.χ. `base64data.c2domain.com`), ενώ οι εντολές από τον εξυπηρετητή μπορούν να επιστραφούν μέσω εγγραφών τύπου TXT, CNAME ή άλλων. Τα πλεονεκτήματα αυτής της προσέγγισης περιλαμβάνουν την εξαιρετική της δυνατότητα απόκρυψης, καθώς η κίνηση DNS είναι πανταχού παρούσα και συχνά υπόκειται σε λιγότερο αυστηρό έλεγχο σε σύγκριση με την HTTP. Η θύρα 53 είναι σχεδόν πάντοτε ανοιχτή, επιτρέποντας την παράκαμψη ακόμη και πολύ περιοριστικών τειχών προστασίας. Παρόλα αυτά, το DNS ως κανάλι C2 παρουσιάζει σημαντικά μειονεκτήματα, όπως το πολύ χαμηλό εύρος ζώνης (bandwidth), καθώς τα δεδομένα πρέπει να συμμορφώνονται με τα όρια μεγέθους των DNS εγγραφών. Επιπλέον, απαιτεί πιο πολύπλοκη υλοποίηση και μπορεί να ανιχνευθεί μέσω εξειδικευμένης ανάλυσης της κίνησης DNS, η οποία αναζητά ανωμαλίες όπως ασυνήθιστα μεγάλο όγκο ερωτήσεων, μακροσκελείς ή παράξενους υποτομείς, συχνές ερωτήσεις για TXT records, ή ανάλυση της φήμης του ονόματος τομέα [29].

Για επικοινωνία κυρίως εντός εσωτερικών δικτύων, αξιοποιείται συχνά το πρωτόκολλο **SMB (Server Message Block)** [53], στις θύρες 445 ή 139 TCP. Η επικοινωνία μπορεί να πραγματοποιηθεί μέσω μηχανισμών όπως τα «named pipes» ή με την εγγραφή και ανάγνωση αρχείων σε κοινόχρηστους φακέλους (file shares). Το κύριο πλεονέκτημα είναι ότι η C2 κίνηση ενσωματώνεται πλήρως με τη νόμιμη κίνηση διαμοιρασμού αρχείων των Windows εντός ενός δικτύου, καθιστώντας την αποτελεσματική για πλευρική κίνηση και έλεγχο agents στο ίδιο τοπικό δίκτυο. Ωστόσο, το SMB είναι εντελώς ακατάλληλο για επικοινωνία μέσω του διαδικτύου, καθώς αυτές οι θύρες συνήθως μπλοκάρονται στην περίμετρο του δικτύου, και η εκτεταμένη χρήση του μπορεί να δημιουργήσει ανιχνεύσιμο «θόρυβο».

Το πρωτόκολλο **ICMP (Internet Control Message Protocol)**, αν και δεν χρησιμοποιεί συγκεκριμένη θύρα, αποτελεί ένα ακόμη κανάλι. Ο μηχανισμός περιλαμβάνει την ενσωμάτωση δεδομένων C2 μέσα στο ωφέλιμο φορτίο (payload) των πακέτων ICMP echo request/reply, παρόμοια με αυτά που χρησιμοποιεί η εντολή `ping`. Το πλεονέκτημά του έγκειται στη δυνατότητα να είναι πολύ συγκεκαλυμμένο, εάν η κίνηση ICMP δεν φιλτράρεται ή δεν επιθεωρείται αυστηρά από τα συστήματα ασφαλείας, και στην απλότητα της ιδέας του. Τα μειονεκτήματα, όμως, είναι σημαντικά: προσφέρει εξαιρετικά χαμηλό εύρος ζώνης και η κίνηση ICMP συχνά μπλοκάρεται ή περιορίζεται αυστηρά (rate-limiting) από τα τείχη προστασίας. Μπορεί, επίσης, να ανιχνευθεί από συστήματα που εκτελούν βαθιά επιθεώρηση πακέτων (Deep Packet Inspection - DPI) και αναζητούν μη τυπικά payloads στα πακέτα ICMP.

Συγκριτικά, η προσέγγιση που υιοθετήθηκε στην παρούσα εργασία, δηλαδή η απευθείας σύνδεση TCP σε μια μη τυπική θύρα (όπως η 9999) με κρυπτογράφηση AES σε επίπεδο εφαρμογής, προσφέρει μεν

τον πλήρη έλεγχο του πρωτοκόλλου και την εμπιστευτικότητα των δεδομένων μέσω του AES, αλλά παρουσιάζει σημαντικά μειονεκτήματα όσον αφορά την απόκρυψη και την παράκαμψη τειχών προστασίας σε σύγκριση με κανάλια όπως το HTTP/S ή το DNS. Τα πλεονεκτήματά της, σε σχέση με ένα εντελώς μη κρυπτογραφημένο προσαρμοσμένο πρωτόκολλο TCP, είναι η παροχή εμπιστευτικότητας στο περιεχόμενο. Ωστόσο, σε σύγκριση με HTTP/S ή DNS, η χρήση μιας μη τυπικής θύρας την καθιστά πιθανό στόχο για μπλοκάρισμα από εξερχόμενα (egress) τείχη προστασίας. Η δομή της σύνδεσης (TCP handshake, ακολουθούμενο από την ανταλλαγή κρυπτογραφημένων με AES δεδομένων που ακολουθούν ένα συγκεκριμένο πρωτόκολλο εφαρμογής) δημιουργεί ένα μοτίβο που, αν και το περιεχόμενο είναι κρυπτογραφημένο, μπορεί δυνητικά να ανιχνευθεί από συστήματα παρακολούθησης δικτύου που αναζητούν μη τυπική κίνηση. Δεν επιτυγχάνει την πλήρη «ανάμειξη» με τη νόμιμη κίνηση όπως τα προαναφερθέντα πρωτόκολλα που χρησιμοποιούν τυπικές θύρες και δομές. Παρ' όλα αυτά, η υλοποίηση της κρυπτογράφησης AES σε επίπεδο εφαρμογής επιλέχθηκε για την απλότητα και τον έλεγχο που παρέχει στο πλαίσιο των εκπαιδευτικών σκοπών της εργασίας.

Εν κατακλείδι, η επιλογή του καναλιού επικοινωνίας αποτελεί έναν κρίσιμο παράγοντα στον σχεδιασμό ενός C2 πλαισίου και συχνά περιλαμβάνει έναν συμβιβασμό μεταξύ της ευκολίας υλοποίησης, της ταχύτητας, του διαθέσιμου εύρους ζώνης και, πρωτίστως, της δυνατότητας απόκρυψης και της ανθεκτικότητας έναντι των αμυντικών μέτρων.

2.4 Μελέτες Περίπτωσης: C2 σε Σημαντικά Κυβερνο-Συμβάντα

Η θεωρητική κατανόηση των C2 frameworks αποκτά πλήρες νόημα όταν αναλύονται παραδείγματα από την πραγματική, σύγχρονη ιστορία των κυβερνοαπειλών. Σημαντικά κυβερνο-συμβάντα, ειδικά αυτά που αποδίδονται σε προηγμένες επίμονες απειλές (APTs) ή σε μεγάλες εγκληματικές οργανώσεις, συχνά βασίζονται σε εξελιγμένες υποδομές διοίκησης και ελέγχου για την επιτυχή διεξαγωγή των επιχειρήσεών τους. Η μελέτη συγκεκριμένων περιπτώσεων αναδεικνύει την ποικιλία στις αρχιτεκτονικές και τα κανάλια επικοινωνίας, τις τεχνικές απόκρυψης που χρησιμοποιούνται, και τον κρίσιμο ρόλο που διαδραματίζει το C2 στην επίτευξη των τελικών στόχων του επιτιθέμενου. Σε αυτή την ενότητα, θα εξετασθούν επιλεγμένα παραδείγματα σημαντικών κυβερνο-συμβάντων για να αναδειχθεί ο τρόπος χρήσης των πλαισίων C2 στην πράξη.

2.4.1 Stuxnet (Περίπου 2010)

Το Stuxnet αποτελεί ένα από τα πιο γνωστά και αναλυθέντα παραδείγματα malware, κυρίως λόγω της στοχευμένης του φύσης και της πολυπλοκότητας του. Σχεδιάστηκε για να σαμποτάρει συγκεκριμένες βιομηχανικές μονάδες ελέγχου (SCADA systems) της Siemens, που χρησιμοποιούνταν στο πυρηνικό πρόγραμμα του Ιράν. Η υποδομή C2 του Stuxnet ήταν πολλαπλών επιπέδων και χρησιμοποίησε διάφορα κανάλια για να εξασφαλίσει τόσο την αρχική μετάδοση εντολών όσο και την ανθεκτικότητα. Αρχικά, το Stuxnet διέδιδε τον εαυτό του μέσω μολυσμένων USB drives (ένα σπάνιο αρχικό φορέα για malware τέτοιας πολυπλοκότητας). Μόλις μόλυνε ένα σύστημα, προσπαθούσε να επικοινωνήσει με C2 servers για να αναφέρει την κατάστασή του και να λάβει ενημερώσεις ή περαιτέρω εντολές. Χρησιμοποίησε

κυρίως HTTP και P2P επικοινωνία. Η HTTP επικοινωνία γινόταν προς συγκεκριμένα domains και IPs. Το P2P στοιχείο του botnet επέτρεπε στα μολυσμένα συστήματα να επικοινωνούν μεταξύ τους και να μεταδίδουν εντολές ή να αναζητούν ενεργούς C2 servers, αυξάνοντας σημαντικά την ανθεκτικότητα σε περίπτωση που οι κύριοι servers απομακρύνονταν. Το C2 του Stuxnet ήταν καίριας σημασίας για τη φάση της «λογικής βόμβας» (logic bomb), επιτρέποντας στους χειριστές να ενεργοποιούν ή να τροποποιούν τη σαμποταριστική συμπεριφορά του malware βάσει των συνθηκών που έβρισκε στα συστήματα SCADA. Η ανακάλυψη και ανάλυση των C2 servers του Stuxnet ήταν ένα κρίσιμο βήμα για την κατανόηση των στόχων και των δυνατοτήτων του [32], [54].

2.4.2 SolarWinds Supply Chain Attack (Ανακαλύφθηκε το 2020)

Η επίθεση στην αλυσίδα εφοδιασμού της SolarWinds είναι ένα από τα πιο εκτεταμένα και επιζήμια κυβερνο-συμβάντα των τελευταίων ετών, αποδιδόμενο σε κρατικό φορέα (συνήθως την APT29 ή Cozy Bear). Οι επιτιθέμενοι μόλυναν το λογισμικό παρακολούθησης δικτύου Orion της SolarWinds με μια «κερκόπορτα» (γνωστή ως SUNBURST). Όταν οι πελάτες της SolarWinds εγκατέστησαν την μολυσμένη έκδοση, το SUNBURST malware εγκαταστάθηκε στα συστήματά τους.

Το SUNBURST χρησιμοποιούσε μια εξελιγμένη και συγκεκαλυμμένη μέθοδο C2. Αρχικά, παρέμενε αδρανές για έως και δύο εβδομάδες για να αποφύγει την ανίχνευση. Στη συνέχεια, επικοινωνούσε με το C2 server χρησιμοποιώντας HTTP προς domains που έμοιαζαν νόμιμα, αλλά είχαν δημιουργηθεί δυναμικά με βάση πληροφορίες του μολυσμένου συστήματος. Η κίνηση αυτή προσπαθούσε να μιμηθεί το πρωτόκολλο διαχείρισης της SolarWinds. Το πιο αξιοσημείωτο χαρακτηριστικό του C2 στο SolarWinds ήταν η χρήση νόμιμων υπηρεσιών cloud (όπως Azure, AWS) ως μέρος της υποδομής ή ως «dead drops», καθώς και η πολυπλοκότητα του πρωτοκόλλου επικοινωνίας, το οποίο ήταν δύσκολο να αναλυθεί χωρίς αντίστροφη μηχανική (reverse engineering). Η δυνατότητα του C2 να επιτρέπει στους επιτιθέμενους να επιλέγουν ποιους από τους μολυσμένους στόχους θα προχωρούσαν σε περαιτέρω φάσεις (hands-on-keyboard activity) ήταν καίρια για την επιτυχία της επίθεσης, διατηρώντας το αποτύπωμα σε μη ενδιαφέροντες στόχους, χαμηλό [55], [56].

2.4.3 WannaCry Ransomware (2017)

Το **WannaCry** ήταν μια μαζική επίθεση ransomware που εξαπλώθηκε ραγδαία παγκοσμίως, εκμεταλλευόμενη μια ευπάθεια (EternalBlue [57]) στα συστήματα Windows. Αν και πρωτίστως γνωστό για τη λειτουργία κρυπτογράφησης αρχείων, το WannaCry είχε επίσης μηχανισμούς C2, αν και λιγότερο εξελιγμένους από τα παραδείγματα APT. Ο C2 μηχανισμός του WannaCry χρησιμοποιήθηκε για δύο κυρίως σκοπούς. Πρώτον, για τη λήψη κλειδιών κρυπτογράφησης: Αν και υπήρχε μια αποτυχία σε αυτό το κομμάτι (τα κλειδιά δεν στέλνονταν πάντα σωστά), ο σκοπός ήταν να στέλνονται τα κλειδιά αποκρυπτογράφησης στον server των επιτιθέμενων. Δεύτερον, για την υποδομή πληρωμής. Το C2 χρησιμοποιήθηκε για να ενημερώνει τους επιτιθέμενους για επιτυχείς πληρωμές λύτρων από τα θύματα, ώστε να τους παρέχουν το εργαλείο αποκρυπτογράφησης.

Η επικοινωνία γινόταν κυρίως μέσω HTTP/HTTPS προς συγκεκριμένα hardcoded domains. Ωστόσο,

ένα από τα πιο γνωστά χαρακτηριστικά του WannaCry ήταν η ύπαρξη ενός «kill switch» domain: αν ο κακόβουλος agent μπορούσε να συνδεθεί σε ένα συγκεκριμένο, μη καταχωρημένο domain, διέκοπτε τη λειτουργία κρυπτογράφησης. Ένας ερευνητής ασφαλείας ανακάλυψε αυτό το domain, το καταχώρησε, και έτσι σταμάτησε (ή επιβράδυνε σημαντικά) την εξάπλωση των νέων μολύνσεων, δείχνοντας πώς η ανάλυση C2 μπορεί να χρησιμοποιηθεί για την άμυνα. Αν και το C2 του WannaCry ήταν σχετικά απλό, ήταν αποτελεσματικό για τον σκοπό του (πληρωμές) και ανέδειξε την ανάγκη παρακολούθησης ακόμα και απλών C2 καναλιών [58], [59], [60].

2.4.4 Άλλα Παραδείγματα

Πολυάριθμα άλλα κυβερνο-συμβάντα αναδεικνύουν τη σημασία των πλαισίων C2:

- **Emotet**: Ένα εξελιγμένο botnet που χρησιμοποιούνταν κυρίως για τη διανομή άλλου malware. Είχε μια ιδιαίτερα ανθεκτική P2P αρχιτεκτονική και χρησιμοποιούσε κρυπτογραφημένη επικοινωνία [61].

- **Mirai**: Ένα botnet που στόχευσε κυρίως συσκευές IoT για την πραγματοποίηση μαζικών DDoS επιθέσεων. Χρησιμοποιούσε ένα απλό TCP C2 πρωτόκολλο, συχνά σε μη τυπικές θύρες, αλλά η ισχύς του βρισκόταν στον μεγάλο αριθμό ευπαθών συσκευών IoT [62].

- **Ryuk Ransomware**: Συχνά αναπτυσσόταν χειροκίνητα μετά από αρχική παραβίαση δικτύου (hands-on-keyboard activity), χρησιμοποιώντας C2 frameworks (όπως το Cobalt Strike) για την πλευρική κίνηση, την αναγνώριση του δικτύου και την τελική ανάπτυξη του ransomware [63].

Αυτές οι μελέτες περίπτωσης καταδεικνύουν ότι, παρά τις διαφορές στην πολυπλοκότητα και τις τεχνικές, η ύπαρξη μιας λειτουργικής υποδομής διοίκησης και ελέγχου είναι θεμελιώδης για τη διεξαγωγή, τη διατήρηση και την κλιμάκωση μιας κυβερνοεπίθεσης. Η ανάλυση των C2 καναλιών και υποδομών αποτελεί έτσι έναν από τους κρισιμότερους τομείς της σύγχρονης κυβερνοάμυνας και πληροφοριών (Cyber Threat Intelligence).

2.5 Επισκόπηση Αξιοσημείωτων Πλαισίων C2

Πέρα από τα κατά παραγγελία (bespoke) C2 που αναπτύσσονται από συγκεκριμένες ομάδες APT ή εγκληματικές οργανώσεις, υπάρχει ένα ευρύ οικοσύστημα από έτοιμα πλαίσια C2, διαθέσιμα είτε εμπορικά είτε ως λογισμικό ανοιχτού κώδικα (open-source). Αυτά τα πλαίσια παρέχουν ένα σύνολο εργαλείων και υποδομών που διευκολύνουν σημαντικά την ανάπτυξη και διαχείριση C2 επιχειρήσεων. Αρχικά, πολλά από αυτά αναπτύχθηκαν για νόμιμους σκοπούς, όπως η προσομοίωση επιθέσεων από ομάδες Red Team για την αξιολόγηση της άμυνας ενός οργανισμού. Ωστόσο, λόγω της ισχύος και της ευελιξίας τους, έχουν υιοθετηθεί ευρέως και από κακόβουλους παράγοντες. Η εξέταση μερικών από τα πιο αξιοσημείωτα και ευρέως χρησιμοποιούμενα πλαίσια παρέχει μια εικόνα των σύγχρονων δυνατοτήτων C2.

2.5.1 Cobalt Strike

Το Cobalt Strike [37] διακρίνεται για μια σειρά από ισχυρά χαρακτηριστικά που το καθιστούν ένα εξαιρετικά δημοφιλές, αν και αμφιλεγόμενο, εργαλείο. Στον πυρήνα του βρίσκεται το κύριο implant, γνωστό ως «*beacon*» ή *«φάρος»* στα ελληνικά, το οποίο επιδεικνύει αξιοσημείωτη ευελιξία. Το beacon μπορεί να επικοινωνεί μέσω πολλαπλών διαύλων, όπως HTTP/S, DNS, SMB Named Pipes και TCP Sockets, ενώ λειτουργεί κυρίως με μια ασύγχρονη μέθοδο επικοινωνίας (beaconing), όπου ο agent συνδέεται περιοδικά με τον server.

Ένα από τα πιο καθοριστικά και ισχυρά γνωρίσματα του Cobalt Strike είναι τα λεγόμενα «Malleable C2 Profiles» [40], τα οποία δίνουν τη δυνατότητα στους χρήστες να παραμετροποιούν σχεδόν κάθε πτυχή της δικτυακής επικοινωνίας του beacon. Μπορούν να αλλάξουν τις κεφαλίδες HTTP, τα μοτίβα των URL, τον τρόπο κωδικοποίησης των δεδομένων, τις μεθόδους κρυπτογράφησης, τους χρόνους αναμονής μεταξύ των επικοινωνιών (sleep times) και την τυχαιότητα αυτών των χρόνων (jitter). Αυτή η παραμετροποίηση επιτρέπει στο Beacon να μιμείται την κίνηση νόμιμων εφαρμογών ή ακόμα και συγκεκριμένων τύπων malware, καθιστώντας την ανίχνευσή του βάσει γνωστών υπογραφών δικτύου εξαιρετικά δύσκολη.

Πέρα από την επικοινωνία, το Cobalt Strike προσφέρει μια εκτεταμένη σουίτα ενσωματωμένων τεχνικών για ενέργειες μετά την αρχική παραβίαση (post-exploitation). Περιλαμβάνει πληθώρα εντολών και modules για αναγνώριση του συστήματος και του δικτύου, κλιμάκωση δικαιωμάτων, πλευρική κίνηση σε άλλα συστήματα (χρησιμοποιώντας τεχνικές όπως PsExec, WMI, WinRM), συλλογή διαπιστευτηρίων (με ενσωμάτωση εργαλείων όπως το Mimikatz), καταγραφή πληκτρολογήσεων (keylogging), λήψη στιγμιοτύπων οθόνης και πολλά άλλα.

Η πλατφόρμα είναι επίσης ευέλικτη όσον αφορά την αρχική ανάπτυξη του Beacon στο σύστημα-στόχο. Υποστηρίζει διάφορες μεθόδους, όπως staged ή stageless payloads (δηλαδή, αν το πλήρες κακόβουλο φορτίο παραδίδεται αμέσως ή σε στάδια), εκτέλεση μέσω Powershell injection, τεχνικές όπως reflective DLL loading και process injection για την απόκρυψη της εκτέλεσης.

Επιπλέον, το Cobalt Strike είναι σχεδιασμένο με γνώμονα την ομαδική συνεργασία, ειδικά για επιχειρήσεις Red Team. Διαθέτει τη δυνατότητα «Team Server», η οποία επιτρέπει σε πολλούς χειριστές να συνδέονται ταυτόχρονα στον ίδιο κεντρικό C2 server και να διαχειρίζονται από κοινού τους ίδιους παραβιασμένους agents.

Λόγω ακριβώς αυτής της ισχύος και ευελιξίας, «κλεμμένες» ή παλαιότερες εκδόσεις του Cobalt Strike έχουν, δυστυχώς, υιοθετηθεί ευρέως και αποτελούν βασικό εργαλείο για πολλές προηγμένες ομάδες κυβερνοκατασκοπείας (APT) και συμμορίες ransomware, όπως οι περιβόητες Ryuk και Conti. Κατά συνέπεια, η ανίχνευση της δραστηριότητας που προέρχεται από το Cobalt Strike αποτελεί μια διαρκή και σημαντική πρόκληση για όσους ασχολούνται με την κυβερνοάμυνα.

2.5.2 Metasploit Framework / Meterpreter

Το **Metasploit Framework** [36] είναι ένα από τα πιο διάσημα και μακροβιότερα open-source πλαίσια για την ανάπτυξη, δοκιμή και χρήση exploits. Αν και ο κύριος σκοπός του είναι η εκμετάλλευση ευπαθειών

(exploitation), περιλαμβάνει ένα εξαιρετικά ισχυρό C2 payload γνωστό ως Meterpreter.

Ο Meterpreter, ένα κεντρικό στοιχείο του Metasploit Framework, διαθέτει μια σειρά από χαρακτηριστικά που τον καθιστούν ένα ισχυρό εργαλείο για ενέργειες μετά την αρχική παραβίαση. Ένα από τα πιο σημαντικά του γνωρίσματα είναι η σχεδίασή του να λειτουργεί εξ ολοκλήρου μέσα στη μνήμη του συστήματος-στόχου. Αποφεύγοντας την εγγραφή αρχείων στον σκληρό δίσκο, ο Meterpreter δυσκολεύει σημαντικά την ανίχνευσή του από παραδοσιακά προγράμματα antivirus που βασίζονται κυρίως στη σάρωση αρχείων.

Η πραγματική του ισχύς, ωστόσο, πηγάζει από την αρθρωτή του αρχιτεκτονική. Ο Meterpreter δεν περιέχει όλες τις δυνατότητές του εξ αρχής, αλλά φορτώνει δυναμικά επιπλέον λειτουργίες, γνωστές ως modules ή επεκτάσεις, μέσω του καναλιού επικοινωνίας C2, μόνο όταν ο χειριστής τις χρειάζεται. Αυτό κρατά το αρχικό μέγεθος του Meterpreter (το «αποτύπωμά» του στο σύστημα) μικρό και διακριτικό. Η βιβλιοθήκη των διαθέσιμων modules είναι τεράστια, καλύπτοντας σχεδόν κάθε πιθανή ενέργεια post-exploitation που μπορεί να φανταστεί κανείς. Για την επικοινωνία του με τον server, ο Meterpreter υποστηρίζει πολλαπλά κανάλια, συμπεριλαμβανομένων των TCP, HTTP και HTTPS. Όταν χρησιμοποιεί HTTPS, η επικοινωνία προστατεύεται με κρυπτογράφηση TLS, εξασφαλίζοντας ένα επίπεδο ασφάλειας και δυσκολεύοντας την παρακολούθηση. Ένα άλλο πλεονέκτημά του είναι η ανεξαρτησία του από την εκάστοτε πλατφόρμα. Υπάρχουν διαθέσιμες εκδόσεις του Meterpreter για τα πιο διαδεδομένα λειτουργικά συστήματα, όπως Windows, Linux και macOS, αλλά και για άλλες πλατφόρμες, καθιστώντας τον ένα ευέλικτο εργαλείο για διαφορετικά περιβάλλοντα. Η πλήρης και απρόσκοπτη ενσωμάτωσή του με το υπόλοιπο Metasploit Framework είναι επίσης καθοριστική. Αυτό επιτρέπει στους χρήστες να περνούν εύκολα από το στάδιο της αρχικής εκμετάλλευσης μιας ευπάθειας (exploit) στο στάδιο της μετά-εκμετάλλευσης, αποκτώντας απευθείας μια Meterpreter session στο παραβιασμένο σύστημα.

Στην πράξη, ο Meterpreter αποτελεί ένα θεμελιώδες εργαλείο για επαγγελματίες ελέγχου διείσδυσης (penetration testers) και ερευνητές ασφαλείας. Ωστόσο, λόγω της ανοιχτής φύσης του (open-source), χρησιμοποιείται αναπόφευκτα και από κακόβουλους παράγοντες. Παρ'όλα αυτά, η βασική, μη τροποποιημένη μορφή του Meterpreter μπορεί να είναι πιο εύκολα ανιχνεύσιμη σε σύγκριση με πιο εξειδικευμένα και παραμετροποιήσιμα εργαλεία όπως το Cobalt Strike, εκτός φυσικά αν ο επιτιθέμενος έχει προβεί σε σημαντικές τροποποιήσεις για να αυξήσει την απόκρυψή του [64], [65].

2.5.3 Sliver

Το **Sliver** [39] είναι ένα πιο σύγχρονο, open-source C2 framework γραμμένο στη γλώσσα Go, το οποίο κερδίζει συνεχώς δημοτικότητα τόσο στην κοινότητα των red teamers όσο και, αναπόφευκτα, στους κακόβουλους παράγοντες. Σχεδιάστηκε με γνώμονα την ευελιξία και την αποφυγή ανίχνευσης. Αυτό το C2 framework ξεχωρίζει για την ευελιξία και τις σύγχρονες δυνατότητές του. Ένα από τα βασικά του πλεονεκτήματα είναι ότι είναι γραμμένο στη γλώσσα προγραμματισμού Go, γεγονός που του επιτρέπει να μεταγλωττίζεται εύκολα και να λειτουργεί απρόσκοπτα σε διαφορετικά λειτουργικά συστήματα, όπως Windows, Linux και macOS, καθιστώντας το πραγματικά cross-platform.

Στον τομέα της επικοινωνίας, προσφέρει ποικιλία επιλογών, υποστηρίζοντας διάφορα πρωτόκολλα για τη διοίκηση και τον έλεγχο. Μεταξύ αυτών περιλαμβάνονται τα δημοφιλή HTTPS, το πιο ασφαλές mTLS

(Mutual TLS) που απαιτεί πιστοποιητικά και από τις δύο πλευρές, το WireGuard [66] για VPN-τύπου συνδέσεις, καθώς και το DNS για πιο διακριτική επικοινωνία. Επιπλέον, παρέχει ευελιξία στον τρόπο παράδοσης του agent στο θύμα, υποστηρίζοντας τόσο staged (σε στάδια) όσο και stageless (ολοκληρωμένα) payloads.

Η αρχιτεκτονική του είναι σχεδιασμένη με γνώμονα την ανθεκτικότητα. Υποστηρίζει τη σύνδεση πολλαπλών χειριστών (operators) ταυτόχρονα στον ίδιο server και ενσωματώνει μηχανισμούς ανακατεύθυνσης (redirectors), οι οποίοι βοηθούν στην απόκρυψη του πραγματικού C2 server και στη διατήρηση της επικοινωνίας ακόμα κι αν κάποιοι κόμβοι εντοπιστούν. Το framework διαθέτει επίσης ενσωματωμένες εντολές για την εκτέλεση συνηθισμένων εργασιών μετά την αρχική παραβίαση. Μια σημαντική δυνατότητα είναι η υποστήριξη για BOFs (Beacon Object Files), παρόμοια με αυτή του Cobalt Strike, που επιτρέπει τη δυναμική φόρτωση και εκτέλεση μικρών, μεταγλωττισμένων κομματιών κώδικα για εξειδικευμένες λειτουργίες.

Ένα ακόμα στοιχείο που δυσκολεύει την άμυνα είναι η δυνατότητα δυναμικής μεταγλώττισης. Κάθε φορά που δημιουργείται ένας νέος agent (implant), μπορεί να μεταγλωττιστεί με ελαφρώς διαφορετικές ρυθμίσεις, με αποτέλεσμα να αλλάζει το hash του αρχείου και να καθίσταται πιο δύσκολη η ανίχνευσή του από συστήματα που βασίζονται σε γνωστές υπογραφές (hashes). Δεδομένων αυτών των χαρακτηριστικών, δεν προκαλεί έκπληξη η αυξανόμενη υιοθέτησή του από ομάδες APT (Advanced Persistent Threats) και άλλες εγκληματικές οργανώσεις. Αυτό το καθιστά ένα σημαντικό C2 framework που οι ομάδες κυβερνοάμυνας πρέπει να παρακολουθούν στενά και να αναπτύσσουν στρατηγικές ανίχνευσης και αντιμετώπισης [67], [68].

2.5.4 Empire / Starkiller

Το **Empire** [38], μαζί με τον διάδοχό του που διαθέτει γραφικό περιβάλλον, το **Starkiller** [69], υπήρξε για καιρό ένα πολύ δημοφιλές C2 framework ανοιχτού κώδικα. Η κύρια του δύναμη και εστίαση ήταν η εκτεταμένη χρήση του Powershell για τους agents που στόχευαν συστήματα Windows, ενώ για περιβάλλοντα Linux και macOS βασιζόταν σε agents γραμμένους σε Python. Παρόλο που η αρχική ομάδα σταμάτησε την επίσημη ανάπτυξή του, η κοινότητα της κυβερνοασφάλειας ανέλαβε να το συντηρεί και να το εξελίσσει, με χαρακτηριστικό παράδειγμα το fork της BC-Security. Το δυνατότερο σημείο του Empire ήταν η πλήρης αξιοποίηση των δυνατοτήτων του Powershell. Επέτρεπε την εκτέλεση κώδικα απευθείας στη μνήμη, την αλληλεπίδραση με το Windows API και την υλοποίηση πολυάριθμων τεχνικών post-exploitation, όλα αυτά χωρίς να χρειάζεται να γράψει ούτε ένα δυαδικό αρχείο στον δίσκο, κάτι που το καθιστούσε πιο δύσκολο να εντοπιστεί από παλαιότερα συστήματα ασφαλείας.

Για την επικοινωνία μεταξύ του agent και του server, το Empire χρησιμοποιούσε κρυπτογραφημένα κανάλια, συνήθως βασισμένα στο πρωτόκολλο HTTP ή HTTPS, προσθέτοντας ένα επίπεδο προστασίας και δυσκολίας στην παρακολούθηση της κίνησης. Ακολουθώντας μια προσέγγιση παρόμοια με αυτή του Metasploit, το Empire είχε μια αρθρωτή δομή. Διέθετε ένα σύστημα από modules, τα οποία μπορούσαν να φορτωθούν και να εκτελεστούν για να πραγματοποιήσουν διάφορες εργασίες, από συλλογή πληροφοριών μέχρι κλιμάκωση δικαιωμάτων και πλευρική κίνηση. Ένα από τα στοιχεία που συνέβαλαν στη δημοτικότητά του ήταν και η σχετική ευκολία χρήσης του. Η κονσόλα διαχείρισής του ήταν φιλική προς

τον χρήστη, ενώ το Starkiller προσέφερε μια ακόμα πιο βολική εναλλακτική με το γραφικό του περιβάλλον.

Στην πράξη, το Empire αποτέλεσε αγαπημένο εργαλείο για πολλές ομάδες red team που ήθελαν να εστιάσουν και να δοκιμάσουν τεχνικές βασισμένες στο Powershell. Φυσικά, όπως συμβαίνει με τέτοια εργαλεία, υιοθετήθηκε και από κακόβουλους παράγοντες. Η χρήση του, ωστόσο, έχει μειωθεί σε σχέση με το παρελθόν, κυρίως λόγω της σημαντικής βελτίωσης των μηχανισμών καταγραφής (logging) και ανάλυσης του Powershell από τη Microsoft και της αυξημένης αποτελεσματικότητας των σύγχρονων λύσεων EDR (Endpoint Detection and Response) στον εντοπισμό τέτοιων δραστηριοτήτων [70], [71].

2.5.5 Havoc

Το **Havoc** [72] είναι ένα σχετικά καινούργιο όνομα στον χώρο των C2 frameworks, αλλά έχει καταφέρει να τραβήξει γρήγορα την προσοχή της κοινότητας κυβερνοασφάλειας, τόσο από την επιθετική όσο και από την αμυντική πλευρά. Πρόκειται για ένα σύγχρονο, ανοιχτού κώδικα πλαίσιο για ενέργειες μετάεκμετάλλευσης, γραμμένο κυρίως σε γλώσσες προγραμματισμού C++, Go και Python. Η φιλοσοφία του είναι έντονα αρθρωτή και επεκτάσιμη, ακολουθώντας τα βήματα γνωστών frameworks όπως το Cobalt Strike και το Sliver, με στόχο να προσφέρει ένα ευέλικτο και δύσκολο στην ανίχνευση εργαλείο.

Η αρχιτεκτονική του είναι ξεκάθαρα χωρισμένη σε διακριτά μέρη. Υπάρχει ο Teamserver, που είναι ο κεντρικός εγκέφαλος, ο C2 server δηλαδή. Έπειτα είναι ο Client, που αποτελεί τη διεπαφή για τον χρήστη (συχνά με γραφικό περιβάλλον) για να δίνει εντολές και να βλέπει αποτελέσματα. Τέλος, υπάρχουν τα Implants ή Agents, τα οποία στο Havoc ονομάζονται «Demons» και εγκαθίστανται στα παραβιασμένα συστήματα. Ο κύριος agent, ο «Demon», είναι γραμμένος σε C/C++, κάτι που του επιτρέπει να μεταγλωττιστεί και να τρέξει σε διάφορες πλατφόρμες, αν και η κύρια εστίαση είναι στα Windows. Η χρήση C/C++ δίνει τη δυνατότητα για εκτέλεση κώδικα σε πολύ χαμηλό επίπεδο, επιτρέποντας την απευθείας αλληλεπίδραση με το λειτουργικό σύστημα και τις κλήσεις του API, κάτι που μπορεί να είναι χρήσιμο για την υλοποίηση προηγμένων τεχνικών και την αποφυγή ανίχνευσης.

Όσον αφορά την επικοινωνία, το Havoc είναι ευέλικτο, υποστηρίζοντας διάφορους μηχανισμούς όπως HTTP/S και SMB Named Pipes. Παρέχει επίσης δυνατότητες για παραμετροποίηση αυτής της επικοινωνίας, επιτρέποντας στους χρήστες να αλλάζουν κάποια χαρακτηριστικά της για να την κάνουν να μοιάζει με νόμιμη κίνηση και να κρύβονται καλύτερα, αν και ίσως όχι στο ίδιο επίπεδο λεπτομέρειας με τα Malleable C2 Profiles του Cobalt Strike. Μια ενδιαφέρουσα δυνατότητα που ενσωματώνει είναι η υποστήριξη για «Sleep Obfuscation». Αυτό σημαίνει ότι προσπαθεί να κρύψει τον agent μέσα στη μνήμη του συστήματος κατά τις περιόδους που είναι αδρανής («κοιμάται»), κάνοντας πιο δύσκολη τη δουλειά των σαρωτών μνήμης που χρησιμοποιούν τα συστήματα EDR για να τον εντοπίσουν. Το Havoc διευκολύνει επίσης την εκτέλεση διαφόρων τύπων κώδικα post-exploitation. Επιτρέπει την εκτέλεση απλών εντολών shell, Powershell scripts, καθώς και .NET assemblies απευθείας μέσα στη μνήμη. Επιπλέον, υποστηρίζει τη χρήση BOFs (Beacon Object Files) και Reflective DLLs, προσφέροντας έτσι μεγάλη ευελιξία στους χειριστές για να τρέξουν εξειδικευμένα εργαλεία και κώδικα. Για τον χρήστη, ο client του Havoc προσφέρει μια σύγχρονη και εύχρηστη γραφική διεπαφή, κάνοντας τη διαχείριση των «Demons» και την εκτέλεση των εργασιών πιο απλή.

Επειδή είναι ένα σχετικά νέο framework, η καταγεγραμμένη χρήση του από κακόβουλους παράγοντες μπορεί να μην είναι τόσο εκτεταμένη όσο αυτή παλαιότερων εργαλείων όπως το Cobalt Strike, αλλά σίγουρα βρίσκεται σε άνοδο. Η ανοιχτή του φύση, η μοντέρνα αρχιτεκτονική του και οι δυνατότητες αποφυγής ανίχνευσης το καθιστούν ελκυστικό τόσο για τις ομάδες red team όσο και για τους πραγματικούς επιτιθέμενους. Θεωρείται, δικαίως, ένα από τα σημαντικά ανερχόμενα C2 frameworks που χρήζουν προσοχής [73].

Συμπερασματικά, η επισκόπηση αυτών των πλαισίων – από τα καθιερωμένα Cobalt Strike και Metasploit/Meterpreter, στα επικεντρωμένα στο Powershell όπως το Empire, και στα πιο σύγχρονα όπως το Sliver και το Havoc – καταδεικνύει την πολυπλοκότητα και την ισχύ των σύγχρονων εργαλείων C2. Χαρακτηριστικά όπως η ευέλικτη επικοινωνία, η αρθρωτή σχεδίαση, οι ενσωματωμένες τεχνικές post-exploitation και οι δυνατότητες απόκρυψης αποτελούν κοινό παρονομαστή. Η ύπαρξη τόσο εμπορικών όσο και ισχυρών open-source επιλογών σημαίνει ότι η πρόσβαση σε προηγμένες δυνατότητες C2 είναι ευρύτερη από ποτέ, υπογραμμίζοντας την ανάγκη για συνεχή επαγρύπνηση και προσαρμογή των αμυντικών στρατηγικών.

2.6 Μετά-Εκμετάλλευση: Το Εγχειρίδιο του Επιτιθέμενου

Αφού ένας επιτιθέμενος αποκτήσει αρχική πρόσβαση με κάποιον τρόπο (δεν θα συζητηθεί στην παρούσα εργασία διότι ξεφεύγει από την κεντρική θεματολογία της) και εγκαθιδρύσει ένα κανάλι διοίκησης και ελέγχου (C2), εισέρχεται στη φάση μετά-εκμετάλλευσης (post-exploitation). Αυτή η φάση περιλαμβάνει ένα ευρύ φάσμα ενεργειών που αποσκοπούν στην επίτευξη των τελικών στόχων της επίθεσης. Για την κατανόηση, την κατηγοριοποίηση και την ανάλυση αυτών των ενεργειών, η κοινότητα της κυβερνοασφάλειας βασίζεται εκτενώς στο πλαίσιο MITRE ATT&CK [10].

2.6.1 Εισαγωγή στο Μοντέλο MITRE ATT&CK για Επιχειρήσεις (Enterprise)

Το **MITRE ATT&CK** [10] (Adversarial Tactics, Techniques, and Common Knowledge) είναι μια παγκοσμίως προσβάσιμη βάση γνώσεων για τακτικές και τεχνικές που βασίζονται σε παρατηρήσεις πραγματικών επιθέσεων. Δημιουργήθηκε και συντηρείται από τον μη κερδοσκοπικό οργανισμό MITRE. Το πλαίσιο ATT&CK for Enterprise εστιάζει συγκεκριμένα στις ενέργειες που πραγματοποιούν οι επιτιθέμενοι εντός των εταιρικών δικτύων μετά την αρχική παραβίαση. Το ATT&CK οργανώνεται γύρω από τις εξής βασικές έννοιες:

- Τακτικές (Tactics): Αντιπροσωπεύουν τον τακτικό στόχο ή το «γιατί» πίσω από μια ενέργεια του επιτιθέμενου. Παραδείγματα τακτικών περιλαμβάνουν την Εκτέλεση (Execution), την Εδραίωση Παρουσίας (Persistence), την Κλιμάκωση Δικαιωμάτων (Privilege Escalation), την Αποφυγή Άμυνας (Defense Evasion), την Ανακάλυψη (Discovery), την Πλευρική Κίνηση (Lateral Movement), τη Συλλογή (Collection), τη Διοίκηση και Έλεγχο (Command and Control) και την Υποκλοπή (Exfiltration).
- Τεχνικές (Techniques): Περιγράφουν το «πώς» ένας επιτιθέμενος επιτυγχάνει έναν τακτικό στόχο.

Κάθε τακτική περιέχει ένα σύνολο τεχνικών. Για παράδειγμα, η τακτική «Execution» περιλαμβάνει τεχνικές όπως η «Command and Scripting Interpreter» (T1059) ή η «Scheduled Task/Job» (T1053).

- Υπο-τεχνικές (Sub-techniques): Ορισμένες τεχνικές αναλύονται περαιτέρω σε υπο-τεχνικές, οι οποίες περιγράφουν πιο συγκεκριμένους τρόπους υλοποίησης μιας τεχνικής. Για παράδειγμα, η τεχνική T1059 («Command and Scripting Interpreter») έχει υπο-τεχνικές όπως η T1059.001 («Powershell») και η T1059.003 («Windows Command Shell»).
- Διαδικασίες (Procedures): Περιγράφουν τις συγκεκριμένες υλοποιήσεις των τεχνικών/υπο-τεχνικών από συγκεκριμένες ομάδες απειλών (threat groups) ή κακόβουλα λογισμικά (malware).

Το πλαίσιο ATT&CK είναι ανεκτίμητο τόσο για τους επιτιθέμενους (red teams, που το χρησιμοποιούν για να σχεδιάσουν προσομοιώσεις απειλών) όσο και, κυρίως, για τους αμυνόμενους (blue teams, security analysts), οι οποίοι το χρησιμοποιούν για να κατανοήσουν τις μεθόδους των επιτιθέμενων, να αναπτύξουν κανόνες ανίχνευσης, να αξιολογήσουν την κάλυψη ασφαλείας τους και να ιεραρχήσουν τις αμυντικές τους προσπάθειες.

2.6.2 Βασικές Τακτικές και Τεχνικές Post-Exploitation

Το εκπαιδευτικό C2 framework που αναπτύχθηκε σε αυτή την εργασία, αν και βασικό, διευκολύνει την εκτέλεση τεχνικών που αντιστοιχούν σε διάφορες τακτικές του MITRE ATT&CK. Παρακάτω εξετάζονται συνοπτικά οι εν λόγω τεχνικές:

1. **Εκτέλεση (Execution - TA0002)**: Μετά την αρχική πρόσβαση και την ανακάλυψη του περιβάλλοντος, η τακτική της Εκτέλεσης είναι θεμελιώδης για την επίτευξη των στόχων του επιτιθέμενου [74]. Περιλαμβάνει όλες τις μεθόδους που επιτρέπουν την εκτέλεση κώδικα ή εντολών στο παραβιασμένο σύστημα υπό τον έλεγχο του επιτιθέμενου. Το C2 framework λειτουργεί ως ο μηχανισμός παράδοσης και ενεργοποίησης αυτών των εντολών. Μια κεντρική τεχνική που αξιοποιείται από το εκπαιδευτικό πλαίσιο της παρούσας εργασίας είναι:
    - T1059: Command and Scripting Interpreter: Αυτή η ευρεία τεχνική καλύπτει τη χρήση των ενσωματωμένων διερμηνέων εντολών και γλωσσών scripting του λειτουργικού συστήματος για την εκτέλεση κώδικα [75]. Είναι ιδιαίτερα δημοφιλής γιατί συχνά αποφεύγει την ανάγκη απόθεσης νέων εκτελέσιμων αρχείων (living off the land).
    - T1059.001: Powershell: Το C2 πλαίσιο που αναπτύχθηκε εστιάζει στην απομακρυσμένη εκτέλεση μέσω Powershell. Ο server αποστέλλει την εντολή Powershell στον agent, ο οποίος χρησιμοποιεί την εντολή Invoke-Expression (ή εναλλακτικά, θα μπορούσε να χρησιμοποιήσει το Powershell API) για να την εκτελέσει. Αυτό παρέχει τεράστια ευελιξία, καθώς το Powershell μπορεί να αλληλεπιδράσει με το WMI, το .NET Framework, τη Registry και πολλά άλλα στοιχεία του συστήματος [76]. Η δυνατότητα αυτή επιτρέπει την εκτέλεση σχεδόν οποιασδήποτε ενέργειας που θα μπορούσε να κάνει ένας χρήστης τοπικά.
    - T1059.003: Windows Command Shell: Παρόλο που το framework εστιάζει στο Powershell, είναι σημαντικό να σημειωθεί ότι μέσω του Powershell είναι δυνατή και η εκτέλεση εντολών

του κλασικού Command Prompt (cmd.exe). Έτσι, έμμεσα, το C2 μπορεί να αξιοποιήσει και αυτή την υπο-τεχνική, αν ο επιτιθέμενος στείλει μια εντολή που καλεί το cmd.exe [77].

Η δυνατότητα απομακρυσμένης εκτέλεσης μέσω C2 είναι αυτή που μετατρέπει ένα παραβιασμένο σύστημα από έναν παθητικό κόμβο σε ένα ενεργό εργαλείο στα χέρια του επιτιθέμενου.

2. **Ανακάλυψη (Discovery - TA0007)**: Μετά την εδραίωση της πρόσβασης και της επικοινωνίας C2, ένα από τα πρώτα και κρισιμότερα βήματα για έναν επιτιθέμενο είναι η τακτική της Ανακάλυψης [78]. Σκοπός αυτής της τακτικής είναι η συλλογή πληροφοριών για το εσωτερικό περιβάλλον του συστήματος και του δικτύου στο οποίο έχει αποκτήσει πρόσβαση ο επιτιθέμενος. Αυτή η «χαρτογράφηση» του εδάφους είναι θεμελιώδης για τον σχεδιασμό των επόμενων κινήσεων, όπως η πλευρική κίνηση, η κλιμάκωση δικαιωμάτων ή η στοχευμένη συλλογή δεδομένων. Το εκπαιδευτικό C2 πλαίσιο διευκολύνει την εκτέλεση διαφόρων τεχνικών ανακάλυψης, όπως:

   - T1082: System Information Discovery: Επιτρέπει τη συγκέντρωση βασικών αλλά και λεπτομερών χαρακτηριστικών του συστήματος-στόχου. Μέσω του C2, εντολές όπως systeminfo μπορούν να αποκαλύψουν την έκδοση του λειτουργικού συστήματος, την αρχιτεκτονική (x86/x64), το όνομα του υπολογιστή, τον κατασκευαστή, καθώς και τις εγκατεστημένες ενημερώσεις (patches), πληροφορίες κρίσιμες για τον εντοπισμό πιθανών ευπαθειών [79].
   - T1033: System Owner/User Discovery: Η γνώση του χρήστη που εκτελεί τον agent ή άλλων συνδεδεμένων χρηστών είναι συχνά απαραίτητη. Η εκτέλεση της εντολής whoami μέσω του C2 παρέχει άμεσα το όνομα του τρέχοντος χρήστη και το domain στο οποίο ανήκει, βοηθώντας στην κατανόηση του επιπέδου δικαιωμάτων και του πλαισίου λειτουργίας [80].
   - T1016: System Network Configuration Discovery: Η κατανόηση της θέσης του παραβιασμένου συστήματος στο δίκτυο είναι ζωτικής σημασίας. Τεχνικές όπως η εκτέλεση ipconfig /all ή Get-NetIPConfiguration μέσω του C2 αποκαλύπτουν τη διεύθυνση IP, τη μάσκα υποδικτύου, την προεπιλεγμένη πύλη (default gateway) και τους DNS servers, παρέχοντας στοιχεία για τη δομή του τοπικού δικτύου [81].
   - T1057: Process Discovery: Η ανάλυση των διεργασιών που εκτελούνται μπορεί να αποκαλύψει την παρουσία λογισμικού ασφαλείας (AV/EDR), κρίσιμων εταιρικών εφαρμογών ή ευκαιριών για άλλες τεχνικές (process injection). Η δυνατότητα εκτέλεσης tasklist ή Get-Process μέσω του C2 παρέχει αυτή την ορατότητα στον επιτιθέμενο [82].

   Αυτές οι τεχνικές ανακάλυψης, εκτελούμενες συνδυαστικά μέσω ενός C2 πλαισίου, δίνουν στον επιτιθέμενο μια ολοκληρωμένη εικόνα του άμεσου περιβάλλοντός του, επιτρέποντάς του να λαμβάνει τεκμηριωμένες αποφάσεις για τις επόμενες ενέργειές του εντός του παραβιασμένου δικτύου.

3. **Συλλογή (Collection - TA0009)**: Αφού ο επιτιθέμενος έχει κατανοήσει το περιβάλλον και μπορεί να εκτελεί εντολές, συχνά προχωρά στην τακτική της Συλλογής [83]. Στόχος είναι η συγκέντρωση συγκεκριμένων δεδομένων που έχουν αξία για τον επιτιθέμενο, είτε πρόκειται για διαπιστευτήρια, ευαίσθητα έγγραφα, είτε για πληροφορίες που θα διευκολύνουν περαιτέρω ενέργειες. Το C2 κανάλι είναι απαραίτητο τόσο για την καθοδήγηση της συλλογής όσο και για την μετέπειτα υποκλοπή των δεδομένων. Μια χαρακτηριστική τεχνική που θα μπορούσε να υποστηριχθεί από ένα C2 όπως το εκπαιδευτικό πλαίσιο της εργασίας είναι:

- T1113: Screen Capture: Η δυνατότητα λήψης στιγμιότυπων της οθόνης του παραβιασμένου συστήματος παρέχει στον επιτιθέμενο άμεση οπτική πρόσβαση στο τι βλέπει ο νόμιμος χρήστης τη δεδομένη στιγμή [84]. Αυτό μπορεί να αποκαλύψει ευαίσθητες πληροφορίες που εμφανίζονται σε εφαρμογές, ανοιχτά έγγραφα, ή ακόμα και κωδικούς πρόσβασης που πληκτρολογούνται αν η καταγραφή γίνει τη σωστή στιγμή. Η υλοποίηση αυτής της δυνατότητας στο C2 θα απαιτούσε την εκτέλεση κώδικα στον agent για τη λήψη της εικόνας, την κωδικοποίησή της και την αποστολή της πίσω στον server μέσω του C2 καναλιού.

4. **Διοίκηση και Έλεγχος (Command and Control - TA0011)**: Αυτή η τακτική είναι η ίδια η ουσία της λειτουργίας του C2 framework και περιλαμβάνει όλες τις τεχνικές που χρησιμοποιούνται για την εγκαθίδρυση και διατήρηση της επικοινωνίας μεταξύ του επιτιθέμενου και των παραβιασμένων συστημάτων [85]. Οι επιλογές που γίνονται εδώ επηρεάζουν άμεσα την ανθεκτικότητα και τη μυστικότητα της επιχείρησης. Το εκπαιδευτικό πλαίσιο της εργασίας χρησιμοποιεί τις ακόλουθες τεχνικές:

   - T1071: Application Layer Protocol: Το πλαίσιο χρησιμοποιεί πρωτόκολλα επιπέδου εφαρμογής για την επικοινωνία [86]. Συγκεκριμένα:
     - Ένα προσαρμοσμένο πρωτόκολλο βασισμένο σε κείμενο (text-based) με διαχωριστικά (|) μεταδίδεται πάνω από την κρυπτογραφημένη TCP/TLS σύνδεση μεταξύ agent και server για τις κύριες εντολές και τα αποτελέσματα. Αυτό θα μπορούσε να θεωρηθεί ως μια μορφή χρήσης μη τυποποιημένου πρωτοκόλλου εφαρμογής.
     - T1071.001: Web Protocols: Το τμήμα της διαχείρισης του C2 από τον χρήστη γίνεται μέσω μιας web διεπαφής που εξυπηρετείται από τον Flask server, χρησιμοποιώντας το πρότυπο HTTP για τις κλήσεις API μεταξύ του προγράμματος περιήγησης και του server [87].
   - T1573: Encrypted Channel: Η διασφάλιση της εμπιστευτικότητας της C2 επικοινωνίας είναι κρίσιμη. Το πλαίσιο της εργασίας το επιτυγχάνει μέσω κρυπτογράφησης σε επίπεδο εφαρμογής [88].
     - T1573.001: Symmetric Cryptography: Η επικοινωνία μεταξύ server και agent κρυπτογραφείται χρησιμοποιώντας τον συμμετρικό αλγόριθμο AES με ένα προκαθορισμένο κοινό κλειδί. Αν και δεν υπάρχει διαδικασία δυναμικής ανταλλαγής κλειδιών, με RSA ή άλλες μεθόδους όπως στο TLS, η ίδια η κρυπτογράφηση των δεδομένων εμπίπτει σε αυτή την τεχνική [89].
   - T1571: Non-Standard Port: Η χρήση της θύρας 9999 ή οποιασδήποτε άλλης μη καθιερωμένης θύρας για τυπικές υπηρεσίες όπως HTTP/HTTPS/DNS, για την επικοινωνία agent-server αντιστοιχεί σε αυτή την τεχνική [90]. Ενώ μπορεί να παρακάμψει απλά φίλτρα, η μη τυπική θύρα μπορεί επίσης να προκαλέσει υποψίες ή να μπλοκαριστεί από πιο αυστηρά firewalls.

   Η επιλογή ενός προσαρμοσμένου πρωτοκόλλου πάνω σε TLS σε μη τυπική θύρα προσφέρει μεν έλεγχο και κρυπτογράφηση, αλλά στερείται της δυνατότητας απόκρυψης που προσφέρουν τα τυποποιημένα πρωτόκολλα όπως το HTTPS σε τυπικές θύρες.

5. **Υποκλοπή (Exfiltration - TA0010)**: Η τελική φάση πολλών επιθέσεων περιλαμβάνει την εξαγωγή των δεδομένων που έχουν συλλεχθεί από το παραβιασμένο δίκτυο [91]. Η μέθοδος υποκλοπής πρέπει συχνά να είναι εξίσου συγκεκαλυμμένη με την ίδια την C2 επικοινωνία.

- T1041: Exfiltration Over C2 Channel: Αυτή είναι η πιο άμεση και κοινή μέθοδος υποκλοπής, ειδικά για μικρούς όγκους δεδομένων. Τα δεδομένα που συλλέγονται απλά αποστέλλονται πίσω στον C2 server μέσω του ίδιου κρυπτογραφημένου καναλιού που χρησιμοποιείται για τις εντολές [92]. Το εκπαιδευτικό πλαίσιο της εργασίας χρησιμοποιεί αποκλειστικά αυτή τη μέθοδο για την επιστροφή των αποτελεσμάτων των εντολών. Η αποτελεσματικότητά της εξαρτάται από το πόσο καλά κρυμμένο και μη ανιχνεύσιμο είναι το ίδιο το C2 κανάλι.

Η αντιστοίχιση των λειτουργιών του C2 framework με τις τακτικές και τεχνικές του MITRE ATT&CK παρέχει μια δομημένη και αναγνωρισμένη από τη βιομηχανία μέθοδο για την κατανόηση της σημασίας και του πιθανού αντίκτυπου κάθε υλοποιημένης δυνατότητας. Ακόμα και ένα βασικό C2 μπορεί να αγγίξει πολλαπλά σημεία της επιθετικής αλυσίδας.

2.7 Το Powershell ως Επιθετικό Εργαλείο

Η επιλογή της τεχνολογίας για την υλοποίηση του agent σε ένα C2 framework είναι κρίσιμης σημασίας, καθώς καθορίζει τις δυνατότητες, το αποτύπωμα στο σύστημα-στόχο και την ευκολία ανίχνευσης. Στην παρούσα εργασία, επιλέχθηκε η Powershell για την ανάπτυξη του agent που στοχεύει συστήματα Windows. Αυτή η επιλογή δεν είναι τυχαία. Το Powershell έχει εξελιχθεί από ένα απλό εργαλείο διαχείρισης συστημάτων σε μια πανίσχυρη πλατφόρμα αυτοματισμού και scripting, η οποία, ακριβώς λόγω των δυνατοτήτων της, έχει γίνει εξαιρετικά δημοφιλής τόσο στους διαχειριστές συστημάτων όσο και στην κοινότητα της επιθετικής ασφάλειας (offensive security). Η εμπέδωση των λόγων πίσω από αυτή τη δημοτικότητα είναι απαραίτητη.

2.7.1 Εγγενής Παρουσία και Αξιοπιστία στα Windows

Το βασικότερο πλεονέκτημα του Powershell είναι ότι αποτελεί αναπόσπαστο κομμάτι των σύγχρονων λειτουργικών συστημάτων Windows (από τα Windows 7/Server 2008 R2 και μετά). Είναι προεγκατεστημένο, ψηφιακά υπογεγραμμένο από τη Microsoft και θεωρείται αξιόπιστο συστατικό του συστήματος. Αφ' ενός, αυτό σημαίνει ότι δεν απαιτείται εγκατάσταση, μιας και ενας επιτιθέμενος δεν χρειάζεται να εγκαταστήσει επιπλέον λογισμικό ή βιβλιοθήκες και dlls για να εκτελέσει powershell scripts, μειώνοντας το ίχνος της επίθεσης και τις πιθανότητες ανίχνευσης κατά την αρχική εγκατάσταση [93].

Επιπροσθέτως, η χρήση του powershell εντάσσεται πλήρως στη φιλοσοφία του «living off the land» [45], κατά την οποία οι επιτιθέμενοι χρησιμοποιούν τα ήδη υπάρχοντα και κατα πάσα πιθανότητα ελεγμένα εργαλεία του λειτουργικού συστήματος, για να εκτελέσουν κακόβουλες ενέργειες. Αυτό κάνει την ανίχνευση που βασίζεται μονάχα στον εντοπισμό άγνωστων ή κακόβουλων εκτελέσιμων αρχείων (file-based detection) αναποτελεσματική.

2.7.2 Πρόσβαση στο .Net Framework και το Windows API

Η Powershell είναι χτισμένη πάνω στο .NET Framework και στο .NET Core/5+, στις νεότερες εκδόσεις. Αυτό της παρέχει άμεση πρόσβαση σε ένα τεράστιο σύνολο βιβλιοθηκών και κλάσεων που μπορούν να χρησιμοποιηθούν για την αλληλεπίδραση με σχεδόν κάθε πτυχή του λειτουργικού συστήματος. Μέσω του .NET, η Powershell μπορεί να κάνει κλήσεις σε functions του API των Windows, να αλληλεπιδράσει με το μητρώο (registry) των Windows, να διαχειριστεί διεργασίες και υπηρεσίες, να πραγματοποιήσει δικτυακές συνδέσεις (TCP, UDP, HTTP όπως θα παρουσιαστεί και στη συνέχεια), να χειριστεί αρχεία και φακέλους, να αλληλεπιδράσει με το Windows Management Instrumentation (WMI) για συλλογη πληροφοριών και διαχείριση, να εκτελέσει κώδικα .NET assembly απευθείας στη μνήμη και, τέλος, να υλοποιήσει κρυπτογραφικές λειτουργίες (System.Security.Cryptography) [94].

Αυτή η βαθιά ενσωμάτωση και η πρόσβαση σε ισχυρά APIs την καθιστούν ιδανική για την υλοποίηση πολύπλοκων λειτουργιών post-exploitation απευθείας μέσω scripting, χωρίς την ανάγκη μεταγλώττισης δυαδικού κώδικα (αν και μπορεί να φορτώσει και να εκτελέσει τέτοιο κώδικα).

2.7.3 «Fileless» Εκτέλεση και Λειτουργία Εντός Μνήμης (In-Memory Operation)

Ίσως το πιο ελκυστικό χαρακτηρηστικό της Powershell για επιθετικούς σκοπούς είναι η δυνατότητά της να εκτελεί κώδικα χωρίς απαραίτητα να τον αποθηκεύει μόνιμα στον δίσκο. Αυτό αναφέρεται συχνά ως «fileless» εκτέλεση κακόβουλου λογισμικού (αν και ο όρος μπορεί να είναι ελαφρώς παραπλανητικός, καθώς κάποια ίχνη πάντα μένουν). Τεχνικές όπως η λήψη και εκτέλεση εντολών από το διαδίκτυο (παράδειγμα: η εντολή IEX (New-Object Net.WebClient).DownloadString('http://evil.com/payload.ps1') «κατεβάζει» και εκτελεί ένα script απευθείας στη μνήμη), η εκτέλεση κωδικοποιημένων εντολών (π.χ. με την παράμετρο EncodedCommand επιτρέπεται η εκτέλεση κωδικοποιημένων scripts σε base64, αποφεύγοντας έτσι απλούς ελέγχους που βασίζονται σε συμβολοσειρές) αλλά και η φόρτωση και εκτέλεση .NET assembly κώδικα (dll) απευθείας από τη μνήμη του συστήματος χωρίς αποθήκευση στον σκληρό δίσκο, δυσκολεύουν σημαντικά την ανίχνευση απο παραδοσιακές antivirus λύσεις που απλώς σαρώνουν αρχεία στον δίσκο. Σε αργότερο σημείο της εργασίας, φαίνεται πρακτικά πως υιοθετείται αυτή η φιλοσοφία μέσα στο εκπαιδευτικό περιβάλλον διοίκησης και ελέγχου που αναπτύχθηκε [95], [96].

2.7.4 Προκλήσεις Ασφάλειας και Ανίχνευσης

Η ίδια η ευελιξία και η ισχύς που κάνουν την Powershell τόσο χρήσιμη, δημιούργησαν ταυτόχρονα την ανάγκη για την ανάπτυξη πιο ισχυρών μηχανισμών ασφαλείας και ανίχνευσης, τόσο από τη Microsoft όσο και από τις εταιρείες που παρέχουν λύσεις ασφαλείας. Οι ακόλουθοι μηχανισμοί προσπαθούν να περιορίσουν την κακόβουλη χρήση της Powershell:

Ένας από τους πιο βασικούς μηχανισμούς είναι η **Πολιτική Εκτέλεσης (Execution Policy)** [97]. Αυτή θέτει περιορισμούς στο ποια scripts επιτρέπεται να εκτελεστούν σε ένα σύστημα. Αν και αποτελεί έναν πρώτο φραγμό, δεν θεωρείται ισχυρό μέτρο ασφαλείας από μόνο του, καθώς υπάρχουν τρόποι παράκαμψής του – όπως ακριβώς γίνεται και κατά την εκτέλεση του agent που αναπτύχθηκε για την παρούσα

εργασία με την παράμετρο -ExecutionPolicy Bypass.

Στη συνέχεια, ένα πολύ πιο χρήσιμο εργαλείο για την ανίχνευση είναι η **Καταγραφή Μπλοκ Κώδικα (Script Block Logging)**. Αυτή η δυνατότητα επιτρέπει στο σύστημα να καταγράφει το ακριβές περιεχόμενο των scripts που εκτελούνται μέσω Powershell, ακόμη και αν αυτά είναι κωδικοποιημένα ή δημιουργούνται και εκτελούνται δυναμικά κατά τη διάρκεια της εκτέλεσης άλλου κώδικα. Αυτό παρέχει πολύτιμες πληροφορίες στους αμυνόμενους για το τι ακριβώς προσπάθησε να κάνει ένα script. Παρόμοια λειτουργεί και η Καταγραφή Modules (Module Logging), η οποία καταγράφει ποια συγκεκριμένα modules της Powershell φορτώνονται και χρησιμοποιούνται κατά τη διάρκεια μιας συνεδρίας. Αυτό μπορεί να αποκαλύψει τη χρήση επικίνδυνων ή ύποπτων modules [98]. Για μια πλήρη εικόνα, υπάρχει η δυνατότητα της Μεταγραφής (Transcription). Όταν είναι ενεργοποιημένη, καταγράφει τα πάντα: κάθε εντολή που πληκτρολογείται και κάθε αποτέλεσμα που εμφανίζεται σε μια συνεδρία Powershell, δημιουργώντας ένα πλήρες αρχείο καταγραφής της δραστηριότητας.

Τέλος, ένας από τους πιο σημαντικούς σύγχρονους μηχανισμούς είναι το **AMSI (Antimalware Scan Interface)** [99]. Πρόκειται για μια ειδική διεπαφή που επιτρέπει σε εφαρμογές, όπως η ίδια η Powershell, να επικοινωνούν με τα εγκατεστημένα προϊόντα antimalware. Πριν εκτελεστεί ένα script ή μια εντολή, η Powershell μπορεί να στείλει το περιεχόμενό της (ακόμα και αν είναι κωδικοποιημένο ή βρίσκεται μόνο στη μνήμη) στο antimalware για έλεγχο. Το AMSI είναι πολύ αποτελεσματικό στο να ανιχνεύει γνωστό κακόβουλο κώδικα Powershell, αν και, όπως συμβαίνει συχνά στον κόσμο της κυβερνοασφάλειας, έχουν αναπτυχθεί και εξελίσσονται συνεχώς τεχνικές για την παράκαμψή του (γνωστές ως AMSI bypass) [100], [101].

Ωστόσο, παρά αυτών των βελτιώσεων στην άμυνα, η Powershell παραμένει ένα εξαιρετικά δημοφιλές εργαλείο για post-exploitation λόγω της προεγκατεστημένης της παρουσίας, της ευελιξίας και συνάμα, της ισχύος της. Πολλά σύγχρονα C2 frameworks (όπως αυτά που αναφέρθηκαν στην ενότητα 2.5) προσφέρουν δυνατότητες εκτέλεσης Powershell ή ενσωματώνουν λειτουργίες που αρχικά αναπτύχθηκαν ως Powershell scripts. Η επιλογή της στην παρούσα εργασία αντικατοπτρίζει τη σημασία της στον πραγματικό κόσμο των επιθέσεων σε περιβάλλοντα Windows.

2.8 Ασφάλεια Επικοινωνίας Δικτύου

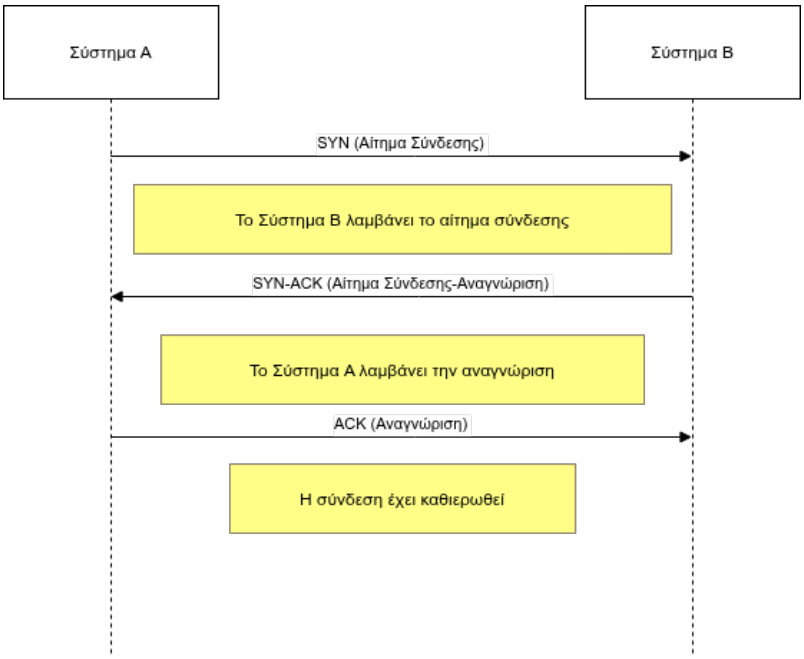
Η επικοινωνία μεταξύ του C2 server και των agents αποτελεί την καρδιά της υποδομής διοίκησης και ελέγχου. Η ασφάλεια αυτής της επικοινωνίας είναι κρίσιμης σημασίας για τον επιτιθέμενο, καθώς η αποκάλυψη ή η υποκλοπή της μπορεί να οδηγήσει στην ανίχνευση ολόκληρης της επιχείρησης, στην απώλεια ελέγχου των παραβιασμένων συστημάτων, ή ακόμα και στην αποκάλυψη της ταυτότητας του επιτιθέμενου. Για την κατανόηση των μηχανισμών ασφάλειας, είναι απαραίτητο να εξεταστούν οι βασικές αρχές των πρωτοκόλλων δικτύου που χρησιμοποιούνται.

2.8.1 Βασικές Αρχές TCP/IP

Η συντριπτική πλειοψηφία της επικοινωνίας στο διαδίκτυο, συμπεριλαμβανομένης και της C2 επικοινωνίας στην περίπτωση της συγκεκριμένης εργασίας, βασίζεται στη σουίτα πρωτοκόλλων TCP/IP. Τα δύο κύρια πρωτόκολλα που έχουν ενδιαφέρον εδώ είναι το Internet Protocol (IP) και το Transmission Control Protocol (TCP) [102].

Το IP, λειτουργεί στο τρίτο επίπεδο (Network Layer) του μοντέλου OSI και αντίστοιχα και του μοντέλου TCP/IP. Είναι υπεύθυνο για τη διευθυνσιοδότηση (χρήση διευθύνσεων IP) και τη δρομολόγηση των πακέτων δεδομένων από τον αποστολέα στον παραλήπτη μέσω του δικτύου. Το IP είναι ένα πρωτόκολλο «χωρίς σύνδεση» (connectionless) και «βέλτιστης προσπάθειας» (best-effort), που σημαίνει ότι δεν εγγυάται την παράδοση, τη σωστή σειρά ή την ακεραιότητα των πακέτων [102].

Το TCP [103], ή πρωτόκολλο ελέγχου μεταφοράς, είναι ένας από τους ακρογωνιαίους λίθους του διαδικτύου και λειτουργεί στο λεγόμενο επίπεδο μεταφοράς (transport layer) του μοντέλου δικτύωσης. Ο κύριος σκοπός του είναι να προσφέρει έναν αξιόπιστο τρόπο για να μεταφέρονται δεδομένα μεταξύ δύο συσκευών, πάνω από το αναξιόπιστο πρωτόκολλο IP. Αν το IP μπορεί να παρομοιαστεί σαν ένα ταχυδρομείο που απλά προσπαθεί να στείλει γράμματα, χωρίς να εγγυάται ότι θα φτάσουν ή με ποια σειρά, το TCP έρχεται από «πάνω» και βάζει τάξη σε αυτό το χάος. Αυτό γίνεται εφικτό ως εξής. Πρώτα απ' όλα, το TCP είναι connection-oriented, πράγμα το οποίο σημαίνει πως πριν σταλεί έστω και ένα byte δεδομένων, το TCP φροντίζει να δημιουργήσει μια σταθερή, λογική σύνδεση μεταξύ του αποστολέα και του παραλήπτη. Αυτό γίνεται μέσω μιας διαδικασίας που ονομάζεται «τριμερής χειραψία» (three-way handshake, βλ. Σχήμα 2.3).



Σχήμα 2.3: Τριμερής Χειραψία

Ο αποστολέας στέλνει ένα πακέτο SYN (Synchronize), ο παραλήπτης απαντά με ένα SYN/ACK (Synchronize/Acknowledge), και ο αποστολέας ολοκληρώνει τη σύνδεση με ένα ACK (Acknowledge). Μόνο τότε ξε-

κινά η κανονική ροή δεδομένων.

Αφού η σύνδεση εγκατασταθεί, το TCP εγγυάται την παράδοση των δεδομένων. Χρησιμοποιεί αριθμούς ακολουθίας (sequence numbers) για να βάλει τα δεδομένα στη σωστή σειρά και επιβεβαιώσεις (acknowledgements - ACKs). Κάθε φορά που ο παραλήπτης λαμβάνει ένα κομμάτι δεδομένων (που ονομάζεται τμήμα ή segment), στέλνει πίσω μια επιβεβαίωση. Αν ο αποστολέας δεν λάβει επιβεβαίωση για κάποιο τμήμα μέσα σε ένα λογικό χρονικό διάστημα, υποθέτει ότι χάθηκε και το ξαναστέλνει.

Εκτός από την εγγύηση παράδοσης, το TCP διαχειρίζεται και τον ρυθμό της επικοινωνίας. Ο έλεγχος ροής (flow control) διασφαλίζει ότι ο αποστολέας δεν θα «πνίξει» τον παραλήπτη στέλνοντας δεδομένα γρηγορότερα απ' ό,τι μπορεί να τα επεξεργαστεί. Παράλληλα, ο έλεγχος συμφόρησης (congestion control) προσπαθεί να αποτρέψει την υπερφόρτωση του ίδιου του δικτύου, προσαρμόζοντας δυναμικά τον ρυθμό αποστολής ανάλογα με τις συνθήκες του δικτύου.

Τέλος, το TCP αναλαμβάνει να διασπάσει τη συνεχή ροή δεδομένων που στέλνει μια εφαρμογή σε μικρότερα, διαχειρίσιμα κομμάτια ή τμήματα (segments) κατά την αποστολή, και στη συνέχεια να τα επανασυναρμολογήσει στη σωστή σειρά στην πλευρά του παραλήπτη, παραδίδοντας τα στην αντίστοιχη εφαρμογή σαν μια ενιαία ροή δεδομένων, ακριβώς όπως στάλθηκαν.

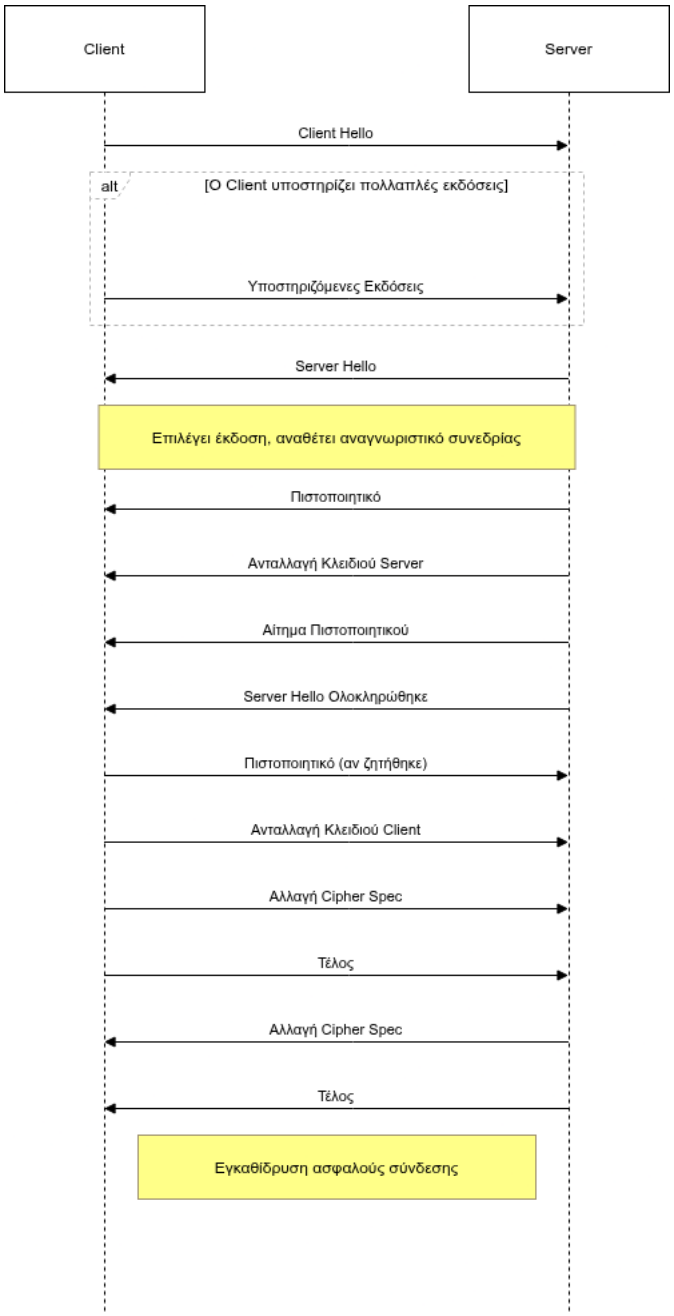
Η χρήση του TCP για τη μεταφορά δεδομένων του C2 πλαισίου εξασφαλίζει ότι οι εντολές και τα αποτελέσματα θα παραδοθούν αξιόπιστα. Ωστόσο, το TCP από μόνο του δεν παρέχει καμία κρυπτογράφηση, αφήνοντας τα δεδομένα εκτεθειμένα σε υποκλοπή.

2.8.2 Ανάγκη και Μέθοδοι Κρυπτογράφησης και Ασφαλούς Επικοινωνίας C2

Για την αντιμετώπιση της έλλειψης ασφάλειας του απλού TCP και την προστασία της εμπιστευτικότητας των δεδομένων που ανταλλάσσονται σε μια υποδομή C2, είναι επιτακτική η χρήση μηχανισμών που παρέχουν ασφάλεια. Οι προσεγγίσεις ποικίλλουν, από την αξιοποίηση καθιερωμένων πρωτοκόλλων ασφαλούς μεταφοράς έως την εφαρμογή κρυπτογράφησης σε επίπεδο εφαρμογής ή τη χρήση εναλλακτικών πρωτοκόλλων επικοινωνίας.

Transport Layer Security (TLS)

Το **Transport Layer Security (TLS)**, διάδοχος του Secure Sockets Layer (SSL), είναι το κυρίαρχο πρωτόκολλο για την παροχή ασφάλειας στις επικοινωνίες μέσω δικτύου [104], [105]. Λειτουργεί μεταξύ του επιπέδου μεταφοράς (TCP) και του επιπέδου εφαρμογής, δημιουργώντας ένα ασφαλές κανάλι πάνω από μια υπάρχουσα TCP σύνδεση. Το TLS παρέχει τρεις θεμελιώδεις εγγυήσεις ασφαλείας: εμπιστευτικότητα (μέσω συμμετρικής κρυπτογράφησης όπως AES μετά από ασφαλή ανταλλαγή κλειδιών κατά τη χειραψία TLS), ακεραιότητα δεδομένων (μέσω MACs) και αυθεντικοποίηση (κυρίως του server μέσω ψηφιακών πιστοποιητικών X.509). Η διαδικασία εγκαθίδρυσης μιας TLS σύνδεσης ξεκινά με τη χειραψία TLS (TLS Handshake), όπου διαπραγματεύονται οι παράμετροι ασφαλείας και ανταλλάσσονται τα κλειδιά συνόδου (Σχήμα 2.4). Αποτελεί το βιομηχανικό πρότυπο για ασφαλή επικοινωνία στο διαδίκτυο και χρησιμοποιείται ευρέως από C2 frameworks που επιδιώκουν να μιμηθούν νόμιμη κίνηση HTTPS.



Σχήμα 2.4: Απλοποιημένη Χειραψία TLS

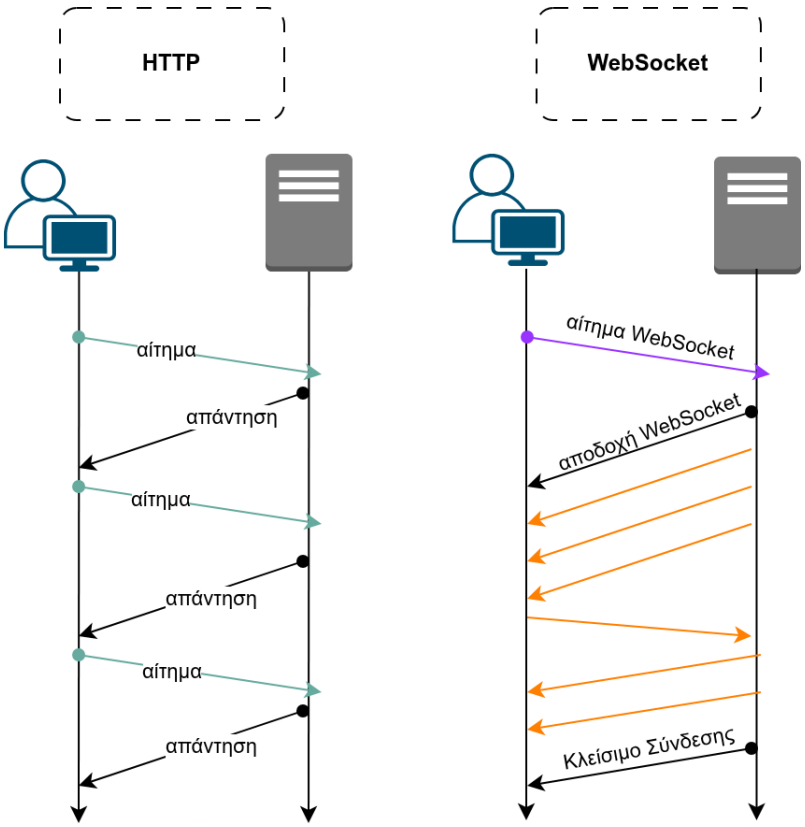
Κρυπτογράφηση σε Επίπεδο Εφαρμογής με Συμμετρικούς Αλγορίθμους

Μια εναλλακτική προσέγγιση για την εξασφάλιση της εμπιστευτικότητας είναι η κρυπτογράφηση των δεδομένων απευθείας από την εφαρμογή, πριν την αποστολή τους μέσω μιας απλής TCP σύνδεσης. Αυτό συνήθως περιλαμβάνει τη χρήση ενός συμμετρικού αλγορίθμου κρυπτογράφησης, όπως ο **AES (Advanced Encryption Standard)**, με ένα **προκαθορισμένο κοινό μυστικό κλειδί (Pre-Shared Key - PSK)** και, συχνά, έναν **Αρχικοποίησης (Initialization Vector - IV)**. Ο αποστολέας κρυπτογραφεί το μήνυμα και ο παραλήπτης το αποκρυπτογραφεί χρησιμοποιώντας το ίδιο κλειδί/IV. Αυτή η μέθοδος, η οποία υιοθετήθηκε στην παρούσα εργασία, προσφέρει απλούστερη υλοποίηση και έλεγχο της διαδικα-

σίας κρυπτογράφησης από την εφαρμογή. Ωστόσο, η ασφάλειά της εξαρτάται από τη μυστικότητα του PSK και στερείται των ενσωματωμένων μηχανισμών αυθεντικοποίησης και διαχείρισης κλειδιών του TLS.

WebSockets για Αμφίδρομη Επικοινωνία

Τα **WebSockets** [106] παρέχουν ένα διαφορετικό μοντέλο επικοινωνίας σε σχέση με το παραδοσιακό HTTP request-response. Πρόκειται για ένα πρωτόκολλο που επιτρέπει την πλήρως αμφίδρομη (full-duplex) επικοινωνία μεταξύ ενός client (π.χ. ενός agent) και ενός server πάνω από μία μόνο, μακρόβια TCP σύνδεση. Η αρχική σύνδεση ενός WebSocket ξεκινά ως ένα τυπικό HTTP αίτημα με ειδικές κεφαλίδες (`Upgrade: websocket`, `Connection: Upgrade`), το οποίο, αν γίνει αποδεκτό από τον server, «αναβαθμίζει» την HTTP σύνδεση σε μια μόνιμη WebSocket σύνδεση.



Σχήμα 2.5: Σύγκριση Επικοινωνίας HTTP με WebSockets

Όσον αφορά την ασφάλεια, τα WebSockets μπορούν να λειτουργήσουν πάνω από TLS, οπότε ονομάζονται **Secure WebSockets (WSS)**. Σε αυτή την περίπτωση, η αρχική HTTP αναβάθμιση γίνεται μέσω HTTPS, και όλη η επακόλουθη WebSocket επικοινωνία κρυπτογραφείται με τις ίδιες εγγυήσεις που παρέχει το TLS (εμπιστευτικότητα, ακεραιότητα, αυθεντικοποίηση server).

Τα WebSockets είναι ελκυστικά για C2 επικοινωνία για διάφορους λόγους:

- **Χαμηλή Καθυστέρηση (Low Latency):** Η μόνιμη σύνδεση επιτρέπει την άμεση αποστολή εντο-

λών από τον server στον agent χωρίς την ανάγκη ο agent να κάνει συνεχώς polling (beaconing).

- **Αμφίδρομη Ροή:** Τόσο ο server όσο και ο agent μπορούν να στείλουν δεδομένα ανά πάσα στιγμή.

- **Παράκαμψη Firewalls:** Η αρχική σύνδεση μέσω HTTP/HTTPS (πόρτες 80/443) διευκολύνει την παράκαμψη πολλών firewalls.

- **Απόκρυψη:** Η κίνηση WSS, όντας κρυπτογραφημένη με TLS, μπορεί να είναι δύσκολο να διακριθεί από άλλη νόμιμη κίνηση HTTPS WebSockets που χρησιμοποιείται από σύγχρονες web εφαρμογές.

Ωστόσο, η υλοποίηση ενός WebSocket server και client μπορεί να είναι πιο πολύπλοκη από την απλή TCP επικοινωνία. Επιπλέον, αν και η κίνηση μπορεί να είναι κρυπτογραφημένη, η ίδια η εγκαθίδρυση μιας μακρόβιας σύνδεσης WebSocket μπορεί να ανιχνευθεί από προηγμένα συστήματα παρακολούθησης δικτύου. Αρκετά σύγχρονα C2 frameworks υποστηρίζουν τα WebSockets ως ένα από τα κανάλια επικοινωνίας τους [107].

2.8.3 Η Σημασία της Κρυπτογραφημένης και Ασφαλούς Επικοινωνίας για το C2

Ανεξάρτητα από την επιλεγμένη μέθοδο (TLS, AES σε επίπεδο εφαρμογής, ή WSS), η προστασία της C2 επικοινωνίας είναι θεμελιώδης για την επιτυχία και τη μυστικότητα μιας επιθετικής επιχείρησης. Η κρυπτογράφηση διασφαλίζει ότι οι εντολές που αποστέλλονται στους παραβιασμένους agents, καθώς και τα ευαίσθητα δεδομένα που επιστρέφονται, παραμένουν εμπιστευτικά και μη αναγνώσιμα από τυχόν παθητικούς υποκλοπείς.

Η κρυπτογραφημένη κίνηση είναι επίσης εγγενώς πιο δύσκολο να αναλυθεί από συστήματα ανίχνευσης εισβολών (IDS) και άλλα εργαλεία ασφαλείας που βασίζονται στην επιθεώρηση του περιεχομένου των πακέτων. Αν και προηγμένες τεχνικές ανάλυσης μεταδεδομένων ή η ανάλυση συμπεριφοράς της κρυπτογραφημένης ροής μπορούν ακόμα να παρέχουν ενδείξεις για κακόβουλη δραστηριότητα, η κρυπτογράφηση αποτελεί ένα σημαντικό πρώτο επίπεδο απόκρυψης.

Στο πλαίσιο της παρούσας εκπαιδευτικής εργασίας, η επιλογή της κρυπτογράφησης AES σε επίπεδο εφαρμογής, παρότι εισάγει τους προαναφερθέντες περιορισμούς σε σύγκριση με μια πλήρη υλοποίηση TLS ή WSS, επιτρέπει την επίτευξη του βασικού στόχου της εμπιστευτικότητας των δεδομένων και την επίδειξη ενός κρυπτογραφημένου καναλιού C2. Η επιλογή αυτή υπαγορεύτηκε από την ανάγκη για απλούστερη υλοποίηση και έλεγχο, καθώς και από τις τεχνικές δυσκολίες που παρουσιάστηκαν κατά την αρχική προσπάθεια ενσωμάτωσης του TLS, καθιστώντας την προσέγγιση AES μια πιο εφικτή λύση για την ολοκλήρωση των εκπαιδευτικών στόχων της εργασίας. Η κατανόηση εναλλακτικών όπως το TLS και τα WebSockets, ωστόσο, είναι σημαντική για την πληρέστερη εικόνα των δυνατοτήτων και των προκλήσεων στην ασφάλεια των C2 επικοινωνιών.

2.9 Βασικές Αρχές Ανίχνευσης και Αποφυγής C2

Η διαρκής «μάχη» μεταξύ επιτιθέμενων και αμυνόμενων στον κυβερνοχώρο είναι ιδιαίτερα εμφανής στον τομέα της διοίκησης και ελέγχου. Ενώ οι επιτιθέμενοι εξελίσσουν συνεχώς τις τεχνικές τους για να διατηρούν κρυφά τα C2 κανάλια τους, οι αμυνόμενοι αναπτύσσουν όλο και πιο εξελιγμένες μεθόδους για την ανίχνευσή τους. Η κατανόηση των βασικών αρχών πίσω από την ανίχνευση και τις αντίστοιχες τεχνικές αποφυγής (evasion) είναι κρίσιμη για την πλήρη εκτίμηση της λειτουργίας και των προκλήσεων που σχετίζονται με τα C2 frameworks, συμπεριλαμβανομένων και των περιορισμών του εκπαιδευτικού πλαισίου που αναπτύχθηκε.

2.9.1 Μέθοδοι Ανίχνευσης C2

Η ανίχνευση της C2 δραστηριότητας μπορεί να πραγματοποιηθεί σε διάφορα επίπεδα, από το δίκτυο έως το μεμονωμένο τελικό σημείο (endpoint). Οι κυριότερες κατηγορίες μεθόδων ανίχνευσης περιλαμβάνουν:

Ανίχνευση Βάσει Δικτύου (Network-Based Detection)

Εστιάζοντας στο επίπεδο του δικτύου, οι κυριότερες κατηγορίες μεθόδων ανίχνευσης μπορούν να ταξινομηθούν ως εξής: Πρωτίστως, η ανίχνευση βάσει υπογραφών (Signature-Based Detection) παραμένει μια θεμελιώδης προσέγγιση. Συστήματα Ανίχνευσης/Αποτροπής Εισβολών (IDS/IPS) και Τείχη Προστασίας Επόμενης Γενιάς (NGFW) συχνά ενσωματώνουν βάσεις δεδομένων με υπογραφές που αντιστοιχούν σε γνωστά μοτίβα δικτυακής κίνησης που σχετίζονται με συγκεκριμένα C2 frameworks ή οικογένειες κακόβουλου λογισμικού. Αυτές οι υπογραφές μπορεί να αφορούν χαρακτηριστικά όπως συγκεκριμένες διαδρομές URL (URL paths), αναγνωριστικά πράκτορα χρήστη (User-Agent strings), ή συγκεκριμένες ακολουθίες bytes εντός των πακέτων. Παρότι η μέθοδος αυτή είναι αποτελεσματική έναντι γνωστών απειλών, η αποτελεσματικότητά της μειώνεται σημαντικά όταν οι επιτιθέμενοι τροποποιούν τα πρωτόκολλα επικοινωνίας ή χρησιμοποιούν κρυπτογράφηση, καθιστώντας τις υπογραφές άχρηστες.

Μια δεύτερη σημαντική τεχνική είναι η ανάλυση της περιοδικής επικοινωνίας (Beaconing Analysis). Η χαρακτηριστική περιοδική, συχνά ρυθμική ή προβλέψιμη, μετάδοση σημάτων (beacons) από έναν μολυσμένο agent προς έναν εξωτερικό εξυπηρετητή C2 συνιστά ισχυρή ένδειξη κακόβουλης δραστηριότητας. Η ανάλυση παραμέτρων όπως η συχνότητα των beacons, η διάρκεια κάθε σύνδεσης, ο όγκος των ανταλλασσόμενων δεδομένων ανά beacon, και η παρουσία ή απουσία τυχαίας απόκλισης στα μεσοδιαστήματα (jitter), μπορεί να αποκαλύψει πρότυπα που αποκλίνουν σημαντικά από τη φυσιολογική δικτυακή συμπεριφορά.

Τρίτον, η ανάλυση της κίνησης DNS (DNS Analysis) είναι κρίσιμη, δεδομένης της συχνής κατάχρησης του πρωτοκόλλου DNS για σκοπούς C2. Η παρακολούθηση της κίνησης DNS μπορεί να αποκαλύψει ανωμαλίες όπως ασυνήθιστα υψηλό όγκο DNS ερωτημάτων από συγκεκριμένους hosts, αιτήματα για μη τυπικούς τύπους εγγραφών (ίσως TXT records που χρησιμοποιούνται για τη μεταφορά δεδομένων), τη χρήση εξαιρετικά μεγάλων ή κωδικοποιημένων υποτομέων (subdomains) για την ενθυλάκωση δεδομέ-

νων, ή την επικοινωνία με domains που έχουν συσχετιστεί στο παρελθόν με κακόβουλη δραστηριότητα ή έχουν δημιουργηθεί πολύ πρόσφατα (domain reputation/age analysis).

Επιπρόσθετα, η ανάλυση της κρυπτογραφημένης κίνησης TLS/SSL (TLS/SSL Analysis) μπορεί να προσφέρει πολύτιμες ενδείξεις, ακόμη και όταν το περιεχόμενο της επικοινωνίας είναι κρυφό. Τα μεταδεδομένα που ανταλλάσσονται κατά τη φάση της χειραψίας TLS μπορούν να αποκαλύψουν ύποπτα χαρακτηριστικά. Η εξέταση του ψηφιακού πιστοποιητικού του εξυπηρετητή (εάν είναι αυτο-υπογεγραμμένο, η ημερομηνία έκδοσης και λήξης του, ο εκδότης του), η διαπραγματευθείσα έκδοση του πρωτοκόλλου TLS, και οι επιλεγμένες σουίτες κρυπτογράφησης (cipher suites) μπορούν να συνθέσουν ένα μοναδικό «δακτυλικό αποτύπωμα» (fingerprint) της σύνδεσης. Τεχνικές όπως η δημιουργία κατακερματισμών JA3/JA3S [108], [109], που βασίζονται στις παραμέτρους των μηνυμάτων Client Hello και Server Hello αντίστοιχα, επιτρέπουν τη σύγκριση αυτών των αποτυπωμάτων με γνωστά αποτυπώματα που σχετίζονται με συγκεκριμένα C2 frameworks.

Η ανάλυση της γεωγραφικής θέσης και της φήμης των διευθύνσεων IP και των domains (Geographical Location and IP/Domain Reputation) αποτελεί επίσης σημαντικό εργαλείο. Η επικοινωνία ενός εσωτερικού host με διευθύνσεις IP ή ονόματα χώρου που εδρεύουν σε γεωγραφικές περιοχές ασυνήθιστες για τη συνήθη δραστηριότητα του οργανισμού, ή που είναι καταχωρημένα σε λίστες κακής φήμης (γνωστό ότι φιλοξενούν κακόβουλο περιεχόμενο ή χρησιμοποιούνται για επιθέσεις), αποτελεί ισχυρή ένδειξη πιθανής C2 επικοινωνίας.

Τέλος, η ανάλυση του όγκου των μεταφερόμενων δεδομένων (Data Volume Analysis) μπορεί να αποκαλύψει ανωμαλίες. Η παρατήρηση ασυνήθιστα μεγάλων όγκων δεδομένων που διακινούνται μέσω πρωτοκόλλων που τυπικά χρησιμοποιούνται για μικρές ανταλλαγές, ή αντιστρόφως, η σταθερή και περιοδική μεταφορά πολύ μικρών πακέτων δεδομένων, μπορεί να υποδηλώνει την ύπαρξη ενός κρυφού καναλιού C2 [110], [111].

Ανίχνευση Βάσει Τελικού Σημείου (Endpoint-Based Detection)

Παράλληλα με την ανάλυση της δικτυακής κίνησης, η εξέταση της δραστηριότητας απευθείας στο τελικό σημείο (endpoint) αποτελεί κρίσιμο πυλώνα για την ανίχνευση των υποδομών C2. Οι κυριότερες τεχνικές που εφαρμόζονται σε αυτό το επίπεδο περιλαμβάνουν τις παρακάτω: Η ανίχνευση βάσει υπογραφών αρχείων και hash αποτελεί την παραδοσιακή προσέγγιση των λύσεων Antivirus (AV). Αυτή η μέθοδος βασίζεται στην ταυτοποίηση γνωστών C2 agents μέσω της σύγκρισης των αρχείων που βρίσκονται αποθηκευμένα στο σύστημα αρχείων με μια βάση δεδομένων γνωστών υπογραφών ή τιμών κατακερματισμού (hashes). Εντούτοις, η αποτελεσματικότητα αυτής της προσέγγισης περιορίζεται σημαντικά από τη χρήση τεχνικών «fileless» (όπου ο κακόβουλος κώδικας εκτελείται απευθείας στη μνήμη χωρίς να εγγράφεται στον δίσκο) ή από την εφαρμογή μεθόδων πολυμορφισμού (polymorphism) και μεταμορφισμού (metamorphism) που μεταβάλλουν τον κώδικα του agent, καθιστώντας τις στατικές υπογραφές αναποτελεσματικές.

Μια πιο σύγχρονη και δυναμική προσέγγιση είναι η ανάλυση συμπεριφοράς (Behavioral Analysis), η οποία υλοποιείται κυρίως από τα συστήματα Ανίχνευσης και Απόκρισης Τελικού Σημείου (Endpoint Detection and Response - EDR). Τα συστήματα EDR παρακολουθούν συνεχώς τις ενέργειες και τις αλ-

ληλεπιδράσεις των διεργασιών που εκτελούνται στο endpoint. Συμπεριφορές που θεωρούνται ύποπτες και ενδέχεται να υποδηλώνουν την παρουσία ενός C2 agent περιλαμβάνουν:

- Τη δημιουργία ασυνήθιστων εξερχόμενων δικτυακών συνδέσεων από μια διεργασία (σύνδεση σε άγνωστη ή κακόφημη διεύθυνση IP, χρήση μη τυπικών θυρών επικοινωνίας, εμφάνιση σταθερού ρυθμού περιοδικής επικοινωνίας - beaconing).
- Την εκτέλεση εντολών Powershell που περιέχουν κωδικοποιημένα ορίσματα ή τη δυναμική λήψη και εκτέλεση κώδικα από το διαδίκτυο μέσω Powershell.
- Την εφαρμογή τεχνικών εισαγωγής κώδικα σε άλλες, συχνά νόμιμες, διεργασίες (process injection) με σκοπό την απόκρυψη ή την κλιμάκωση δικαιωμάτων.
- Την αλληλεπίδραση με ευαίσθητα αρχεία του συστήματος ή με κλειδιά και τιμές του μητρώου των Windows (Registry) που σχετίζονται άμεσα με γνωστούς μηχανισμούς εδραίωσης παρουσίας (persistence).
- Τη χρήση συγκεκριμένων κλήσεων του προγραμματιστικού περιβάλλοντος (API calls) που συχνά αξιοποιούνται από κακόβουλο λογισμικό για την υλοποίηση επιθετικών τεχνικών.

Η σάρωση της μνήμης (Memory Scanning) αποτελεί μια επιπλέον κρίσιμη δυνατότητα των σύγχρονων λύσεων ασφαλείας. Επιτρέπει την ανίχνευση κακόβουλου κώδικα, τμημάτων κώδικα (shellcode) ή γνωστών δεικτών παραβίασης (Indicators of Compromise - IoCs) απευθείας εντός του χώρου διευθύνσεων των εκτελούμενων διεργασιών. Αυτή η τεχνική είναι ιδιαίτερα σημαντική για τον εντοπισμό «fileless» απειλών και την αναγνώριση τεχνικών απόκρυψης, όπως η φόρτωση βιβλιοθηκών DLL απευθείας στη μνήμη (reflective DLL loading). Ο έλεγχος των μηχανισμών εδραίωσης παρουσίας (Persistence Mechanism Monitoring) είναι επίσης θεμελιώδης. Τα συστήματα EDR συχνά παρακολουθούν ενεργά τις κοινές τοποθεσίες και μεθόδους που χρησιμοποιεί το κακόβουλο λογισμικό για να διασφαλίσει την εκ νέου εκτέλεσή του μετά από επανεκκίνηση του συστήματος. Αυτό περιλαμβάνει την παρακολούθηση αλλαγών σε κλειδιά εκκίνησης στο μητρώο (πχ. Run keys), τη δημιουργία ύποπτων προγραμματισμένων εργασιών (Scheduled Tasks), και τη χρήση συνδρομών συμβάντων WMI (WMI event subscriptions) για την εκκίνηση κακόβουλου κώδικα [112].

Τέλος, η ενσωμάτωση με το AMSI (AMSI Integration), όπως αναλύθηκε και σε προηγούμενη ενότητα, παρέχει ένα ισχυρό επίπεδο άμυνας. Η διεπαφή AMSI επιτρέπει στα εγκατεστημένα προϊόντα ασφαλείας (π.χ. AV, EDR) να επιθεωρούν το περιεχόμενο των scripts (όπως Powershell, VBScript, JScript) και άλλων δυναμικών κωδίκων πριν από την εκτέλεσή τους, ακόμη και αν αυτά δεν εγγράφονται ποτέ στον δίσκο και εκτελούνται αποκλειστικά εντός της μνήμης [99].

2.9.2 Τεχνικές Αποφυγής C2 (Evasion Techniques)

Για την αντιμετώπιση των μηχανισμών ανίχνευσης και τη διασφάλιση της λειτουργικής συνέχειας των υποδομών διοίκησης και ελέγχου, οι επιτιθέμενοι εφαρμόζουν ένα ευρύ φάσμα τεχνικών αποφυγής, οι οποίες στοχεύουν τόσο στην απόκρυψη της δικτυακής επικοινωνίας όσο και στην παρεμπόδιση της ανίχνευσης σε επίπεδο τελικού σημείου [113].

Στο επίπεδο του δικτύου, η θεμελιώδης προσέγγιση είναι η χρήση ισχυρής κρυπτογράφησης, κυρίως μέσω TLS, για την προστασία του περιεχομένου της C2 επικοινωνίας από επιθεώρηση. Επιπλέον, καταβάλλεται σημαντική προσπάθεια για την απόκρυψη της κακόβουλης κίνησης εντός της νόμιμης δικτυακής δραστηριότητας. Αυτό περιλαμβάνει τη χρήση κοινών πρωτοκόλλων και θυρών (HTTP/S, DNS), καθώς και την αξιοποίηση προσαρμόσιμων προφίλ C2 (όπως τα Malleable C2 Profiles) για την προσομοίωση γνωστών εφαρμογών. Για την περαιτέρω δυσχέρανση της ανάλυσης δικτυακής συμπεριφοράς, οι επιτιθέμενοι εισάγουν τυχαία απόκλιση (jitter) στα μεσοδιαστήματα επικοινωνίας, χρησιμοποιούν μεταβλητά μεγέθη δεδομένων, εφαρμόζουν εκτεταμένες περιόδους αδράνειας (sleep) και ενίοτε περιορίζουν τη δραστηριότητα C2 εντός των τυπικών ωρών εργασίας.

Στο επίπεδο του τελικού σημείου, οι τεχνικές αποφυγής εστιάζουν στην ελαχιστοποίηση των ανιχνεύσιμων ιχνών. Η «fileless» εκτέλεση, που αποφεύγει την εγγραφή αρχείων στον δίσκο, είναι κεντρική στρατηγική. Συμπληρώνεται από τεχνικές απόκρυψης ή κρυπτογράφησης του κακόβουλου κώδικα εντός της μνήμης (memory obfuscation/encryption) και την εισαγωγή κώδικα σε νόμιμες διεργασίες (process injection) για την κάλυψη της δραστηριότητας. Παράλληλα, αναπτύσσονται συνεχώς μέθοδοι για την παράκαμψη αμυντικών μηχανισμών όπως το AMSI (Antimalware Scan Interface) και οι μηχανισμοί καταγραφής του λειτουργικού συστήματος (logging bypass). Η αξιοποίηση ενσωματωμένων εργαλείων του συστήματος με απρόβλεπτους τρόπους (LOLBAS - Living Off The Land Binaries And Scripts) συνεισφέρει επίσης στην αποφυγή ανίχνευσης βάσει αρχείων [114].

Τέλος, για τη διασφάλιση της ανθεκτικότητας της ίδιας της υποδομής C2, οι επιτιθέμενοι υιοθετούν στρατηγικές πλεονασμού και απόκρυψης. Αυτές περιλαμβάνουν τη χρήση πολλαπλών εφεδρικών C2 εξυπηρετητών και domains, τη δυναμική παραγωγή ονομάτων domain μέσω Αλγορίθμων Παραγωγής Domain (DGAs) για την αποφυγή στατικών μαύρων λιστών, και τη δρομολόγηση της επικοινωνίας μέσω ενδιάμεσων ανακατευθυντών (redirectors) ή proxies για την απόκρυψη της πραγματικής πηγής της C2 υποδομής [115].

Ξεκάθαρα, η ανίχνευση και η αποφυγή C2 αποτελούν ένα δυναμικό πεδίο. Το εκπαιδευτικό C2 framework που αναπτύχθηκε, χρησιμοποιώντας κρυπτογράφηση δεδομένων με AES μεν, αλλά σε μη τυπική TCP θύρα, και χωρίς καμία από τις παραπάνω τεχνικές αποφυγής, θα ήταν εξαιρετικά ευάλωτο σε ανίχνευση σε ένα πραγματικό, παρακολουθούμενο περιβάλλον. Η κατανόηση αυτών των βασικών αρχών ανίχνευσης και αποφυγής είναι απαραίτητη για την πλήρη αξιολόγηση της αποτελεσματικότητας (ή της έλλειψής της) οποιουδήποτε C2 framework.

2.9.3 Επίλογος

Στο παρόν κεφάλαιο, πραγματοποιήθηκε μια εκτενής επισκόπηση του θεωρητικού υποβάθρου που διέπει τις υποδομές διοίκησης και ελέγχου και τις τεχνικές που εφαρμόζονται κατά τη φάση της μετά-εκμετάλλευσης. Η ανάλυση εκκίνησε με την ιστορική εξέλιξη των συστημάτων C2, παρακολουθώντας τη διαδρομή τους από τις πρώιμες μορφές απομακρυσμένης πρόσβασης και τα δίκτυα botnet βασισμένα στο πρωτόκολλο IRC, έως τη σταδιακή μετάβαση στην αξιοποίηση πρωτοκόλλων του Παγκόσμιου Ιστού και την αυξανόμενη εξειδίκευση που χαρακτηρίζει τα σύγχρονα πλαίσια, είτε εμπορικής είτε ανοιχτού κώδικα προέλευσης. Η εξέταση σημαντικών κυβερνο-συμβάντων, όπως τα Stuxnet, SolarWinds και

WannaCry, υπηρέτησε στην ανάδειξη του κρίσιμου επιχειρησιακού ρόλου και της μορφολογικής ποικιλομορφίας των C2 στην πρακτική εφαρμογή.

Κατόπιν, η μελέτη εμβάθυνε στις θεμελιώδεις αρχές λειτουργίας, τον εγγενή σκοπό και τον τυπικό κύκλο ζωής ενός C2 συστήματος. Παράλληλα, εξετάστηκαν οι επικρατούσες αρχιτεκτονικές (κεντροποιημένη, ομότιμη - P2P, υβριδική) και τα συχνότερα χρησιμοποιούμενα κανάλια επικοινωνίας (HTTP/S, DNS, SMB, ICMP), πραγματοποιώντας μια συγκριτική αξιολόγηση των πλεονεκτημάτων και μειονεκτημάτων τους, καθώς και της συνάφειάς τους με την προσέγγιση TCP/TLS που υιοθετήθηκε στο πλαίσιο της παρούσας εργασίας. Η επισκόπηση αξιοσημείωτων C2 frameworks, συμπεριλαμβανομένων των Cobalt Strike, Metasploit/Meterpreter, Empire, Sliver και Havoc, παρείχε μια ενδεικτική εικόνα των σύγχρονων δυνατοτήτων και της τεχνολογικής πολυπλοκότητας των εν λόγω εργαλείων.

Επιπλέον, το κεφάλαιο επιχείρησε τη σύνδεση των λειτουργικών δυνατοτήτων των C2 με το ευρύτερο πλαίσιο της μετά-εκμετάλλευσης, αξιοποιώντας το πλαίσιο MITRE ATT&CK για την ταξινόμηση των τακτικών και τεχνικών που διευκολύνονται ή υλοποιούνται μέσω μιας C2 υποδομής, όπως η Εκτέλεση (Execution), η Ανακάλυψη (Discovery), η Συλλογή (Collection) και η Υποκλοπή (Exfiltration). Αναλύθηκε περαιτέρω ο ιδιαίτερος ρόλος του Powershell ως ισχυρού επιθετικού διαύλου σε περιβάλλοντα Windows, ενώ ανασκοπήθηκαν οι θεμελιώδεις αρχές της ασφάλειας δικτύων βάσει της σουίτας πρωτοκόλλων TCP/IP, και συζητήθηκαν μέθοδοι για την επίτευξη εμπιστευτικότητας στην επικοινωνία, όπως το πρωτόκολλο TLS, τα Websockets και η κρυπτογράφηση σε επίπεδο εφαρμογής. Ολοκληρώνοντας τη θεωρητική ανάλυση, εξετάστηκαν οι βασικές αρχές ανίχνευσης της C2 δραστηριότητας, τόσο σε επίπεδο δικτύου όσο και σε επίπεδο τελικού σημείου, καθώς και οι αντίστοιχες τεχνικές αποφυγής που χρησιμοποιούνται από τους επιτιθέμενους δρώντες.

Έχοντας πλέον εδραιώσει αυτή τη στέρεη θεωρητική βάση, η εργασία προχωρά στο επόμενο κεφάλαιο, το οποίο θα επικεντρωθεί στον λεπτομερή σχεδιασμό και τη μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του εκπαιδευτικού C2 framework που αποτελεί τον πυρήνα της παρούσας πτυχιακής εργασίας. Στο πλαίσιο αυτό, θα εξεταστούν οι στόχοι σχεδιασμού, η αρχιτεκτονική που υιοθετήθηκε, η αιτιολόγηση της τεχνολογικής στοίβας που επιλέχθηκε, το σχεδιασμένο πρωτόκολλο επικοινωνίας και ο επιμέρους σχεδιασμός των συστατικών στοιχείων του πλαισίου.

Κεφάλαιο 3

Εκπαιδευτικό Πλαίσιο C2: Σχεδιασμός και Μεθοδολογία

Κατόπιν της εκτενούς ανάλυσης του θεωρητικού υποβάθρου που αφορά τις υποδομές διοίκησης και ελέγχου, τις τεχνικές μετά-εκμετάλλευσης και τις συναφείς τεχνολογίες, όπως αυτή παρουσιάστηκε στο προηγούμενο κεφάλαιο, η παρούσα ενότητα σηματοδοτεί τη μετάβαση από τη θεωρητική διερεύνηση στην πρακτική υλοποίηση. Το παρόν κεφάλαιο επικεντρώνεται στην λεπτομερή παρουσίαση του σχεδιασμού και της μεθοδολογίας που υιοθετήθηκε για την ανάπτυξη του εκπαιδευτικού C2 πλαισίου, μέρος του συνολικού περιβάλλοντος διοίκησης και ελέγχου, το οποίο συνιστά το κύριο πρακτικό αντικείμενο της παρούσας πτυχιακής εργασίας.

Στο κεφάλαιο αυτό, θα εξεταστούν ενδελεχώς οι θεμελιώδεις αρχιτεκτονικές επιλογές που διέπουν τη δομή του συστήματος, η τεχνολογική στοίβα που αξιοποιήθηκε για την υλοποίηση, το ειδικά σχεδιασμένο πρωτόκολλο επικοινωνίας που ορίστηκε για την αλληλεπίδραση των συστατικών, καθώς και ο επιμέρους σχεδιασμός των βασικών δομικών ενοτήτων του συστήματος: του κεντρικού εξυπηρετητή (Server), του πράκτορα (Agent) που εγκαθίσταται στο σύστημα-στόχο, και της διαδικτυακής διεπαφής χρήστη (Web UI) για τον χειρισμό. Η ενδελεχής κατανόηση των εν λόγω σχεδιαστικών αποφάσεων και της λογικής που τις υπαγόρευσε κρίνεται απαραίτητη. Παρέχει το αναγκαίο υπόβαθρο για την κριτική αξιολόγηση της λειτουργικότητας, των παρεχόμενων δυνατοτήτων, αλλά και των εγγενών περιορισμών του αναπτυχθέντος πλαισίου, η πρακτική επίδειξη του οποίου θα αποτελέσει αντικείμενο της μελέτης περίπτωσης στο επόμενο κεφάλαιο.

3.1 Στόχοι Σχεδιασμού

Ο σχεδιασμός του C2 framework που αναπτύχθηκε δεν αποσκοπούσε στη δημιουργία ενός εργαλείου ανταγωνιστικού προς τα εξελιγμένα εμπορικά ή open-source πλαίσια που αναλύθηκαν στο Κεφάλαιο 2, ούτε φυσικά στην παροχή ενός εργαλείου για κακόβουλη χρήση. Αντιθέτως, οι σχεδιαστικές αποφάσεις προέκυψαν κυρίως από τους εκπαιδευτικούς και επιδεικτικούς σκοπούς που τέθηκαν στην Εισαγωγή (Ενότητες 1.6 και 1.7). Κύριο μέλημα ήταν η δημιουργία ενός συστήματος που να είναι:

- Κατανοητό: Η αρχιτεκτονική και ο κώδικας (ιδίως του server και του agent) έπρεπε να είναι όσο το δυνατόν πιο σαφείς και κατανοητοί, ώστε να διευκολύνεται η μελέτη και η εκμάθηση των βασικών μηχανισμών C2.
- Λειτουργικό: Το πλαίσιο έπρεπε να υλοποιεί τις θεμελιώδεις λειτουργίες ενός C2 συστήματος, επιτρέποντας την πρακτική επίδειξη των βασικών εννοιών.
- Εστιασμένο: Η πολυπλοκότητα έπρεπε να διατηρηθεί σε διαχειρίσιμα επίπεδα για το πλαίσιο μιας διπλωματικής εργασίας, εστιάζοντας στις κεντρικές λειτουργίες και αποφεύγοντας προηγμένα χαρακτηριστικά που θα αύξαναν σημαντικά τον χρόνο ανάπτυξης και την πολυπλοκότητα του κώδικα, αν και προστέθηκαν ορισμένες προχωρημένες λειτουργίες για πληρέστερη επίδειξη.

Με βάση τα παραπάνω, οι συγκεκριμένοι στόχοι σχεδιασμού για το εκπαιδευτικό C2 framework ήταν οι εξής:

1. Υποδομή Server-Agent: Να υλοποιηθεί μια κεντροποιημένη αρχιτεκτονική με έναν κεντρικό server που ακροάζεται για συνδέσεις και πολλαπλούς agents (συμβατούς με Windows) που συνδέονται περιοδικά σε αυτόν.
2. Ασφαλής Επικοινωνία: Να υλοποιηθεί κρυπτογράφηση της επικοινωνίας μεταξύ server και agent χρησιμοποιώντας τον συμμετρικό αλγόριθμο AES (Advanced Encryption Standard) σε κατάσταση λειτουργίας CBC (Cipher Block Chaining) με ένα προκαθορισμένο, κοινό μυστικό κλειδί (Pre-Shared Key - PSK) και έναν σταθερό Αρχικοποίησης (Initialization Vector - IV). Στόχος είναι η παροχή εμπιστευτικότητας στα ανταλλασσόμενα δεδομένα σε επίπεδο εφαρμογής. Η επιλογή αυτή έγινε για λόγους απλοποίησης της υλοποίησης, ελέγχου της διαδικασίας κρυπτογράφησης και αντιμετώπισης τεχνικών προκλήσεων που προέκυψαν με το TLS στο πλαίσιο της παρούσας εργασίας.
3. Βασική Διαχείριση Agents: Ο server να μπορεί να αναγνωρίζει τους agents που συνδέονται, να παρακολουθεί την κατάστασή τους, όπως για παράδειγμα την τελευταία επικοινωνία, τις βασικές πληροφορίες συστήματος όπως όνομα χρήστη, hostname και λειτουργικό σύστημα και να τους παρουσιάζει στον χρήστη μέσω της διεπαφής.
4. Απομακρυσμένη Εκτέλεση Εντολών: Να παρέχεται η δυνατότητα στον χρήστη να στέλνει αυθαίρετες εντολές Powershell προς εκτέλεση σε έναν επιλεγμένο agent. Η επιλογή του Powershell έγινε λόγω της ισχύος και της εγγενούς παρουσίας του στα Windows, όπως αναλύθηκε στην Ενότητα 2.7.
5. Ανάκτηση Αποτελεσμάτων: Ο agent να μπορεί να στέλνει την έξοδο των εκτελεσμένων εντολών πίσω στον server, και ο server να αποθηκεύει και να παρουσιάζει αυτά τα αποτελέσματα στον χρήστη.
6. Προχωρημένες Λειτουργίες Επίδειξης:
   - Να υλοποιηθεί η δυνατότητα λήψης στιγμιότυπου οθόνης (screenshot) από τον agent.
   - Να παρέχεται λειτουργικότητα περιήγησης στο σύστημα αρχείων του agent.

- Να υποστηρίζεται η μεταφόρτωση αρχείων από τον χειριστή στον agent και η λήψη αρχείων από τον agent στον server.

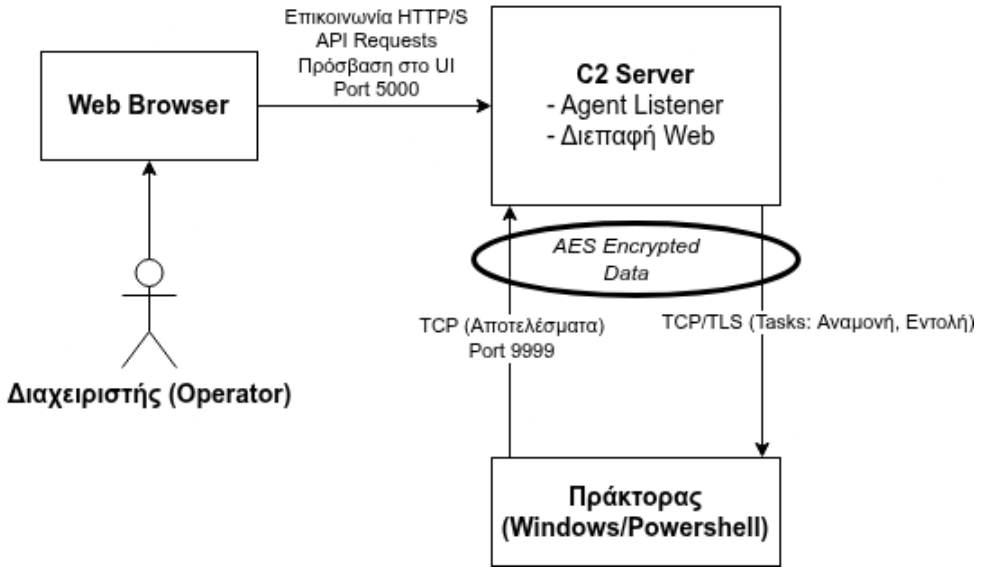
7. Λειτουργική Διεπαφή Χρήστη: Να δημιουργηθεί μια πρακτική και παρουσιάσιμη διεπαφή ιστού που να επιτρέπει στον χρήστη να βλέπει τους ενεργούς agents, να αναθέτει εργασίες, να βλέπει τα αποτελέσματα και να χρησιμοποιεί τις προχωρημένες λειτουργίες με εύκολο τρόπο.

Για την επίτευξη αυτών των στόχων, έγινε συνειδητή επιλογή να παραλειφθούν ορισμένα πολύ προηγμένα χαρακτηριστικά που συναντώνται σε άλλα C2 frameworks, όπως: περίπλοκοι μηχανισμοί αποφυγής ανίχνευσης πέραν της κρυπτογράφησης σε επίπεδο εφαρμογής, πλήρεις μηχανισμοί εδραίωσης επίμονης παρουσίας , υποστήριξη για πολλαπλά πρωτόκολλα ή κανάλια επικοινωνίας πέραν του custom TCP με AES κρυπτογράφηση, και εξαντλητικός χειρισμός όλων των πιθανών σφαλμάτων δικτύου ή συστήματος σε επαγγελματικό, production επίπεδο.

Η εστίαση παρέμεινε στην υλοποίηση των παραπάνω λειτουργιών με τρόπο σαφή και λειτουργικό, ώστε να επιτευχθούν οι εκπαιδευτικοί και επιδεικτικοί σκοποί της εργασίας. Οι επόμενες ενότητες αυτού του κεφαλαίου θα αναλύσουν την αρχιτεκτονική, τις τεχνολογίες και το πρωτόκολλο που επιλέχθηκαν για να υλοποιηθούν αυτοί οι στόχοι σχεδιασμού.

3.2 Αρχιτεκτονική Συστήματος

Η συνολική αρχιτεκτονική του εκπαιδευτικού C2 framework που αναπτύχθηκε ακολουθεί ένα κεντροποιημένο μοντέλο Server-Agent και απεικονίζεται σχηματικά στο Σχήμα 3.1. Η αρχιτεκτονική αυτή σχεδιάστηκε με γνώμονα την απλότητα και την σαφήνεια, επιτρέποντας την εστίαση στις θεμελιώδεις λειτουργίες ενός συστήματος διοίκησης και ελέγχου. Παρακάτω, παρατίθεται ένα διάγραμμα υψηλού επιπέδου στο οποίο φαίνονται ξεκάθαρα τα κύρια μέρη του πλαισίου και ταυτόχρονα, οι βασικές ροές επικοινωνίας μεταξύ τους:



Σχήμα 3.1: Διάγραμμα Αρχιτεκτονικής Υψηλού Επιπέδου του Εκπαιδευτικού C2 Framework

Αναλυτικότερα, το σύστημα απαρτίζεται από τα εξής κύρια συστατικά στοιχεία:

- C2 Server: Αποτελεί την κεντρική οντότητα του πλαισίου, υλοποιημένη σε γλώσσα προγραμματισμού Python. Ο Server φιλοξενεί τη λογική διαχείρισης των agents, την αποθήκευση των εντολών και των αποτελεσμάτων τους, συμπεριλαμβανομένων των δεδομένων από screenshots και αρχεία, και την παροχή διεπαφών επικοινωνίας. Αναλύεται σε δύο κύρια υποσυστήματα που λειτουργούν ταυτόχρονα χάρη στη χρήση νημάτων (threading):
  - Agent Listener (TCP με Κρυπτογράφηση AES): Αυτό το υποσύστημα είναι υπεύθυνο για την ακρόαση σε μια συγκεκριμένη θύρα δικτύου (στην περίπτωση της παρούσας εργασίας είναι η TCP 9999) για εισερχόμενες συνδέσεις από τους agents. Αποδέχεται τις απλές TCP συνδέσεις και στη συνέχεια, για κάθε μήνυμα που ανταλλάσσεται, διαχειρίζεται την αποκρυπτογράφηση των εισερχόμενων δεδομένων και την κρυπτογράφηση των εξερχόμενων δεδομένων χρησιμοποιώντας τον αλγόριθμο AES και το προκαθορισμένο κλειδί/IV, ακολουθώντας το προσαρμοσμένο πρωτόκολλο επικοινωνίας επιπέδου εφαρμογής.
  - Web Server (Flask API/UI): Αυτό το υποσύστημα, υλοποιημένο με τη βιβλιοθήκη Flask της Python, παρέχει τη διεπαφή για τον χειριστή του C2. Εξυπηρετεί τη γραφική διεπαφή χρήστη (Web UI) ως ιστοσελίδα (HTML/CSS/JavaScript) προσβάσιμη μέσω ενός web browser. Παράλληλα, εκθέτει ένα προγραμματιστικό API τύπου RESTful, το οποίο χρησιμοποιείται από το Web UI για την ανάκτηση πληροφοριών (λίστα ενεργών agents, κατάσταση εντολών, αποτελέσματα, δεδομένα screenshots/ αρχείων) και την υποβολή αιτημάτων (ανάθεση νέας εντολής, αίτημα για screenshot, έναρξη μεταφόρτωσης/ λήψης αρχείου).
- Agent (Windows - Powershell): Είναι το λογισμικό που εγκαθίσταται και εκτελείται στο παραβιασμένο σύστημα-στόχο, το οποίο στην παρούσα υλοποίηση είναι σύστημα Windows. Ο agent είναι γραμμένος σε Powershell για να αξιοποιεί τις εγγενείς δυνατότητες του λειτουργικού συστήματος. Ο κύριος ρόλος του είναι να συνδέεται περιοδικά με τον Agent Listener του server μέσω απλής TCP σύνδεσης. Κάθε μήνυμα που αποστέλλεται από ή προς τον agent κρυπτογραφείται/αποκρυπτογραφείται σε επίπεδο εφαρμογής χρησιμοποιώντας τον αλγόριθμο AES και το κοινό μυστικό κλειδί/IV. Σε κάθε κύκλο επικοινωνίας, ο agent λαμβάνει τις αποκρυπτογραφημένες εντολές από τον server, τις εκτελεί, και αποστέλλει πίσω τα κρυπτογραφημένα αποτελέσματα ή τυχόν σφάλματα.
- Χρήστης/Operator: Αναφέρεται στον άνθρωπο που χειρίζεται και ελέγχει το C2 framework.
- Web Browser: Η τυπική εφαρμογή περιήγησης ιστού που χρησιμοποιεί ο Χρήστης/ Operator για να αποκτήσει πρόσβαση και να αλληλεπιδράσει με το Web UI που παρέχεται από τον Flask Web Server.

Οι κύριες ροές επικοινωνίας, όπως φαίνονται και στο Σχήμα 3.1, είναι δύο:

- Agent - Server (Listener): Η κρίσιμη επικοινωνία διοίκησης και ελέγχου. Πραγματοποιείται μέσω TCP στην πόρτα 9999. Τα δεδομένα της εφαρμογής (εντολές, αποτελέσματα, συμπεριλαμβανομένων δεδομένων Base64 για screenshots και αρχεία) κρυπτογραφούνται σε επίπεδο εφαρμογής με

AES από τον αποστολέα και αποκρυπτογραφούνται από τον παραλήπτη, παρέχοντας εμπιστευτικότητα στο περιεχόμενο που μεταφέρεται πάνω από την TCP σύνδεση.

- Χρήστης/Browser - Server (Flask): Η επικοινωνία για τη διαχείριση του C2 η οποία πραγματοποιείται μέσω του πρωτοκόλλου HTTP στην πόρτα 5000. Ο χρήστης ζητά την ιστοσελίδα του UI και στέλνει αιτήματα στο API, ενώ ο server απαντά με το περιεχόμενο της ιστοσελίδας και τις απαντήσεις του API σε μορφή JSON. Αυτή η επικοινωνία, στην τρέχουσα υλοποίηση, δεν είναι κρυπτογραφημένη με TLS (αν και θα μπορούσε να προστεθεί μελλοντικά κάποια κρυπτογράφηση), καθώς θεωρείται ότι ο χρήστης αλληλεπιδρά με τον server σε ένα ελεγχόμενο περιβάλλον. Εξάλλου, το κύριο μέλημα της εργασίας αυτής δεν είναι η κρυπτογραφία.

Συνολικά, η αρχιτεκτονική που υιοθετήθηκε είναι κεντροποιημένη, απλή στην κατανόηση, και διαχωρίζει σαφώς τις αρμοδιότητες μεταξύ των βασικών συστατικών, επιτρέποντας την υλοποίηση των θεμελιωδών και ορισμένων προχωρημένων λειτουργιών ενός C2 framework για εκπαιδευτικούς σκοπούς.

3.3 Αιτιολόγηση Τεχνολογικής Στοίβας

Η επιλογή των κατάλληλων τεχνολογιών είναι θεμελιώδης για την επιτυχή υλοποίηση οποιουδήποτε έργου λογισμικού, πόσο μάλλον ενός συστήματος όπως ένα C2 framework, όπου παράγοντες όπως η ταχύτητα ανάπτυξης, η συμβατότητα με το περιβάλλον-στόχο, η διαθεσιμότητα βιβλιοθηκών και η ευκολία διαχείρισης παίζουν σημαντικό ρόλο. Για την ανάπτυξη του εκπαιδευτικού C2 πλαισίου της παρούσας εργασίας, επιλέχθηκε μια συγκεκριμένη τεχνολογική στοίβα, με κάθε στοιχείο να εξυπηρετεί έναν διακριτό σκοπό, όπως αναλύεται παρακάτω.

3.3.1 Python (Server)

Η Python επιλέχθηκε ως η κύρια γλώσσα προγραμματισμού για την υλοποίηση του C2 Server για τους ακόλουθους λόγους. Αρχικά, επιλέχθηκε λόγω της ταχύτητας ανάπτυξης που προσφέρει αφού η Python είναι γνωστή για τη σαφή και συνοπτική της σύνταξη, η οποία επιτρέπει τη γρήγορη ανάπτυξη πρωτοτύπων και λειτουργικών συστημάτων. Αυτό ήταν κρίσιμο δεδομένου του περιορισμένου χρόνου στο πλαίσιο μιας πτυχιακής εργασίας. Επιπλέον επιλέχθηκε λόγω του πλούσιου οικοσυστήματος βιβλιοθηκών της. Η Python διαθέτει ένα τεράστιο οικοσύστημα από βιβλιοθήκες τρίτων (third-party libraries) μέσω του διαχειριστή πακέτων pip. Για την παρούσα εργασία, μερικές από τις βιβλιοθήκες που αξιοποιήθηκαν είναι οι εξής:

- `socket`: Για τη βασική δικτυακή επικοινωνία TCP.
- `PyCryptodome`: Για την υλοποίηση της συμμετρικής κρυπτογράφησης AES.
- `threading`: Για τη διαχείριση ταυτόχρονων συνδέσεων από agents και τη λειτουργία του listener παράλληλα με τον web server.
- `Flask`: Για τη δημιουργία του Web API και του UI.

- `struct`: Για τη συσκευασία/αποσυσκευασία του προθέματος μήκους στην επικοινωνία.
- `json`, `base64`, `uuid`, `pathlib`, `datetime`: Για γενικές λειτουργίες χειρισμού δεδομένων.

Επίσης, η Python επιλέχθηκε επειδή είναι σχετικά εύκολη στην εκμάθηση και στη χρήση της. Η σχετικά απλή σύνταξη της Python την καθιστά προσιτή, διευκολύνοντας την κατανόηση του κώδικα του server. Ακόμα, ένα πολύ σημαντικό χαρακτηριστικό που κατέχει η Python είναι οι cross-platform δυνατότητες της. Αν και ο agent που αναπτύχθηκε στοχεύει τα Windows, ο Python server μπορεί να εκτελεστεί σε διάφορα λειτουργικά συστήματα (Windows, Linux, macOS) με ελάχιστες ή καθόλου τροποποιήσεις, προσθέτοντας έτσι επιπλέον ευελιξία. Τέλος, είναι γνωστό πως η Python είναι κατάλληλη για δικτυακές εφαρμογές και επιλέχθηκε με αυτό το γεγονός ως γνώμονα. Η Python χρησιμοποιείται ευρέως για την ανάπτυξη δικτυακών εφαρμογών και διαθέτει εξαιρετική υποστήριξη για socket programming και web frameworks.

3.3.2 Flask (Web UI/API)

Για την υλοποίηση της διεπαφής ιστού και του προγραμματιστικού API του C2 server, επιλέχθηκε η βιβλιοθήκη Flask. Καταρχάς, η Flask προσφέρει απλότητα και μινιμαλισμό. Η Flask είναι ένα «microframework», που σημαίνει ότι παρέχει τα βασικά εργαλεία για τη δημιουργία web εφαρμογών (routing, request handling, templating) χωρίς να επιβάλλει μια αυστηρή δομή ή να περιλαμβάνει πολλές προεγκατεστημένες εξαρτήσεις. Το γεγονός αυτό την καθιστά ιδανική για μικρότερα έργα και γρήγορη ανάπτυξη ενός λειτουργικού API και UI. Φυσικά, η Flask προσφέρει ευελιξία, διότι παρόλο που είναι microframework, είναι εξαιρετικά ευέλικτη και επεκτάσιμη μέσω διαφόρων επεκτάσεων (Flask extensions) αν χρειαστεί περαιτέρω λειτουργικότητα. Ωστόσο, η Flask έχει το θετικό ότι ενσωματώνεται απρόσκοπτα με τον υπόλοιπο κώδικα Python του server, επιτρέποντας στις Flask routes να αλληλεπιδρούν εύκολα με τις δομές δεδομένων του C2 (για παράδειγμα τα λεξικά `clients`, `command_queues`, `command_status`, με τη χρήση των κατάλληλων κλειδωμάτων `threading.Lock`). Έπειτα, η ενσωματωμένη υποστήριξη για τη μηχανή προτύπων Jinja2 διευκολύνει τη δημιουργία δυναμικών HTML σελίδων για το Web UI. Τέλος, είναι πολύ εύκολο να οριστούν REST API endpoints που επιστρέφουν δεδομένα σε μορφή JSON, κάτι που είναι ιδανικό για την επικοινωνία με το JavaScript του frontend.

3.3.3 PowerShell (Πράκτορας Windows)

Η επιλογή της Powershell για την υλοποίηση του agent που εκτελείται στο σύστημα-στόχο (Windows) βασίστηκε στα πλεονεκτήματα που αναλύθηκαν εκτενώς στην Ενότητα 2.7 και αξίζει να αναφερθούν ξανά. Η Powershell είναι προεγκατεστημένη στα περισσότερα σύγχρονα συστήματα Windows και έτσι αποφεύγεται η ανάγκη απόθεσης εξωτερικών εκτελέσιμων αρχείων. Ταυτόχρονα έχει ισχυρές δυνατότητες scripting εφόσον παρέχει πλήρη πρόσβαση στο .NET Framework και το Windows API, επιτρέποντας την υλοποίηση σύνθετων λειτουργιών όπως η δικτυακή επικοινωνία, η αλληλεπίδραση με το σύστημα αρχείων, η διαχείριση διεργασιών και η εκτέλεση εντολών. Στη συνέχεια, η Powershell, μέσω των κλάσεων .NET όπως `System.Net.Sockets.TcpClient` για την απλή TCP επικοινωνία και `System.Security.Cryptography.AesManaged` για την κρυπτογράφηση AES, επιτρέπει την

υλοποίηση της κρυπτογραφημένης δικτυακής επικοινωνίας απευθείας μέσα από το script. Σημαντική είναι και η δυνατότητα εκτέλεσης στη μνήμη, αφού η Powershell υποστηρίζει τεχνικές που επιτρέπουν την εκτέλεση κώδικα με μειωμένο αποτύπωμα στον δίσκο του συστήματος το οποίο τη φιλοξενεί. Τέλος, για κάποιον με βασικές γνώσεις scripting, η Powershell προσφέρει ένα ισχυρό περιβάλλον για την αυτοματοποίηση εργασιών, συμπεριλαμβανομένων αυτών που απαιτούνται για έναν C2 agent.

3.3.4 Κρυπτογράφηση AES με Προκαθορισμένο Κλειδί

Για την εξασφάλιση της εμπιστευτικότητας των δεδομένων που ανταλλάσσονται μεταξύ του C2 server και των agents, επιλέχθηκε η υλοποίηση συμμετρικής κρυπτογράφησης σε επίπεδο εφαρμογής, χρησιμοποιώντας τον αλγόριθμο AES (Advanced Encryption Standard) σε κατάσταση λειτουργίας CBC (Cipher Block Chaining). Η προσέγγιση αυτή βασίζεται σε ένα προκαθορισμένο, κοινό μυστικό κλειδί (Pre-Shared Key - PSK) και έναν σταθερό Αρχικοποίησης (Initialization Vector - IV), τα οποία είναι ενσωματωμένα (hardcoded) τόσο στον κώδικα του server όσο και του agent.

Οι κύριοι λόγοι για αυτή την επιλογή ήταν:

- **Εστίαση στην Επίδειξη της Κρυπτογράφησης:** Η υλοποίηση επέτρεψε την εστίαση στην ίδια τη διαδικασία κρυπτογράφησης και αποκρυπτογράφησης των δεδομένων της εφαρμογής, χωρίς την πολυπλοκότητα της διαχείρισης πιστοποιητικών και του πρωτοκόλλου TLS, τα οποία συχνά παρουσιάζουν δυσκολίες υλοποίησης και αποσφαλμάτωσης σε εκπαιδευτικά πλαίσια με περιορισμένο χρόνο, όπως διαπιστώθηκε και κατά την αρχική προσπάθεια υλοποίησης.
- **Απλότητα Υλοποίησης και Ελέγχου:** Η χρήση βιβλιοθηκών όπως η `PyCryptodome` στην Python και οι ενσωματωμένες κλάσεις `System.Security.Cryptography` του .NET Framework στο Powershell (συγκεκριμένα η `AesManaged`), καθιστούν την εφαρμογή του AES σχετικά ευθύγραμμη, επιτρέποντας τον άμεσο έλεγχο της διαδικασίας από τον κώδικα της εφαρμογής.
- **Επάρκεια για Εκπαιδευτική Επίδειξη:** Παρόλο που η χρήση ενός σταθερού, προκαθορισμένου κλειδιού και IV εισάγει σημαντικούς περιορισμούς ασφαλείας σε πραγματικές συνθήκες, για τους σκοπούς της παρούσας εργασίας (επίδειξη ενός κρυπτογραφημένου καναλιού C2), η προσέγγιση αυτή κρίθηκε επαρκής για να καταστήσει τα μεταδιδόμενα δεδομένα μη αναγνώσιμα από παθητικούς παρατηρητές δικτύου.

Είναι σημαντικό να τονιστεί ότι αυτή η μέθοδος κρυπτογράφησης, αν και παρέχει εμπιστευτικότητα, εισάγει συγκεκριμένους περιορισμούς ασφαλείας σε σύγκριση με πιο ολοκληρωμένα πρωτόκολλα όπως το TLS, όπως η εξάρτηση από ένα προκαθορισμένο κλειδί και η απουσία ενσωματωμένων μηχανισμών αυθεντικοποίησης. Οι περιορισμοί αυτοί αναλύονται διεξοδικά στην Ενότητα 5.3.

3.4 Πρωτόκολλο Επικοινωνίας

Για την επιτυχή ανταλλαγή πληροφοριών μεταξύ των συστατικών ενός κατανεμημένου συστήματος, όπως το C2 framework, απαιτείται ο ορισμός ενός σαφούς πρωτοκόλλου επικοινωνίας. Το πρωτόκολλο

καθορίζει τους κανόνες, τη μορφή των μηνυμάτων και τη σειρά των αλληλεπιδράσεων. Στην περίπτωση του εκπαιδευτικού C2 που αναπτύχθηκε, το πρωτόκολλο επικοινωνίας ορίστηκε λαμβάνοντας υπόψη τους στόχους της λειτουργικότητας, της ανθεκτικότητας και της χρήσης κρυπτογραφημένου καναλιού, υιοθετώντας μια πιο στιβαρή προσέγγιση σε σχέση με απλά text-based πρωτόκολλα. Η επικοινωνία διακρίνεται σε δύο κύρια επίπεδα: το επίπεδο μεταφοράς και το επίπεδο εφαρμογής.

3.4.1 Επίπεδο Μεταφοράς: TCP

Η βάση της επικοινωνίας μεταξύ του C2 Server και του Powershell Agent είναι το πρωτόκολλο TCP (Transmission Control Protocol), επιλεγμένο για την αξιοπιστία που παρέχει στην παράδοση των δεδομένων (εγγυημένη παράδοση, σωστή σειρά). Σε αντίθεση με προσεγγίσεις που χρησιμοποιούν TLS σε αυτό το επίπεδο για την κρυπτογράφηση, η παρούσα υλοποίηση βασίζεται σε απλές TCP συνδέσεις. Η εμπιστευτικότητα των δεδομένων επιτυγχάνεται μέσω κρυπτογράφησης AES που εφαρμόζεται στο επίπεδο εφαρμογής, πριν τα δεδομένα παραδοθούν στο TCP για μετάδοση, όπως περιγράφεται στην Ενότητα 3.3.4.

3.4.2 Πρωτόκολλο Επιπέδου Εφαρμογής

Πάνω από την απλή TCP σύνδεση, ένα πρωτόκολλο επιπέδου εφαρμογής χρησιμοποιείται για τη δόμηση των μηνυμάτων. Αντί για απλούς διαχωριστές κειμένου, το πρωτόκολλο αυτό υιοθετεί την τεχνική του προθέματος μήκους (length prefixing) για τον σαφή ορισμό των ορίων κάθε κρυπτογραφημένου μηνύματος. Το πραγματικό περιεχόμενο της εφαρμογής (εντολές, απαντήσεις) κρυπτογραφείται πρώτα με AES και στη συνέχεια αυτό το κρυπτογραφημένο σύνολο δεδομένων ή αλλιώς, «ωφέλιμο φορτίο» (payload) αποστέλλεται.

Μορφή Μηνύματος (Δικτυακό Επίπεδο)

Κάθε μήνυμα που ανταλλάσσεται μεταξύ Server και Agent πάνω από την TCP σύνδεση έχει την ακόλουθη δομή:

1. **Πρόθεμα Μήκους (Length Prefix):** Ένας ακέραιος 4ων bytes (32-bit unsigned integer) που αποστέλλεται σε μορφή big-endian. Αυτός ο αριθμός υποδεικνύει το ακριβές μήκος σε bytes του *κρυπτογραφημένου* σώματος μηνύματος που ακολουθεί.
2. **Κρυπτογραφημένο Σώμα Μηνύματος (Encrypted Message Body):** Τα κρυπτογραφημένα με AES bytes του αρχικού μηνύματος της εφαρμογής, με μήκος ακριβώς όσο ορίζει το πρόθεμα μήκους.

Αυτή η δομή διασφαλίζει ότι ο παραλήπτης μπορεί να διαβάσει πρώτα το μήκος του επερχόμενου κρυπτογραφημένου μπλοκ και στη συνέχεια να διαβάσει ακριβώς τόσα bytes όσα απαιτούνται για να λάβει ολόκληρο το κρυπτογραφημένο μήνυμα, προτού το αποκρυπτογραφήσει.

Περιεχόμενο Αποκρυπτογραφημένων Μηνυμάτων (Επίπεδο Εφαρμογής)

Μετά την αποκρυπτογράφηση του «Κρυπτογραφημένου Σώματος Μηνύματος» από τον παραλήπτη, το αποκρυπτογραφημένο string που προκύπτει, ακολουθεί μια συγκεκριμένη μορφοποίηση για την αναγνώριση εντολών και απαντήσεων:

Μηνύματα από Server προς Agent (Εντολές):

Η μορφή είναι: `CID:<command_uuid>:CMD:<command_string>` όπου:

- `CID:`: Σταθερό πρόθεμα.
- `<command_uuid>`: Ένα μοναδικό αναγνωριστικό (UUID v4) που παράγεται από τον server για κάθε νέα εντολή.
- `:CMD:`: Σταθερό πρόθεμα που υποδηλώνει ότι ακολουθεί η εντολή.
- `<command_string>`: Η πραγματική εντολή προς εκτέλεση από τον agent. Περιλαμβάνει την ίδια την εντολή (για παράδειγμα, `whoami`, `screenshot`, `browse "C:\Users"`, `download "C:\file.txt"`, `upload "C:\dest.dat" <base64_data>`). Για την εντολή upload, τα `<base64_data>` είναι μέρος του `<command_string>`.

Μηνύματα από Agent προς Server (Απαντήσεις):

Η μορφή είναι: `CID:<command_uuid>:<TYPE>:<data>`

- `CID:`: Σταθερό πρόθεμα.
- `<command_uuid>`: Το UUID της εντολής στην οποία απαντά ο agent.
- `:<TYPE>:`: Υποδεικνύει τον τύπο της απάντησης:
  - `:RESP:`: Η εντολή εκτελέστηκε επιτυχώς και ακολουθούν τα δεδομένα αποτελέσματος.
  - `:ERR:`: Παρουσιάστηκε σφάλμα κατά την εκτέλεση της εντολής και ακολουθεί το μήνυμα σφάλματος.
- `:<data>`: Τα δεδομένα της απάντησης, δηλαδή η έξοδος της εντολής ή το μήνυμα σφάλματος, κωδικοποιημένα ως UTF-8 string ή Base64 string.

Χαρακτηριστικά Πρωτοκόλλου

- **Στιβαρότητα:** Η χρήση προθέματος μήκους για τα κρυπτογραφημένα δεδομένα καθιστά το πρωτόκολλο ανθεκτικό σε ζητήματα framing.
- **Παρακολούθηση Κατάστασης:** Η χρήση Command IDs (UUIDs) στο αποκρυπτογραφημένο μήνυμα επιτρέπει στον server να συσχετίζει με αξιοπιστία τις εντολές με τις απαντήσεις τους.

- **Αίτηση/Απάντηση:** Η υλοποίηση ακολουθεί ένα μοντέλο όπου ο server στέλνει μια εντολή και ο agent απαντά σε αυτήν. Δεν υπάρχει ενεργό beaconing από τον agent προς τον server όταν δεν υπάρχει εντολή.

- **Επεκτασιμότητα:** Η δομή `TYPE:data` στο αποκρυπτογραφημένο μήνυμα επιτρέπει την εύκολη προσθήκη νέων τύπων μηνυμάτων.

Αυτό το πρωτόκολλο επικοινωνίας, συνδυάζοντας το πρόθεμα μήκους για τα κρυπτογραφημένα δεδομένα και μια σαφή δομή για τα αποκρυπτογραφημένα μηνύματα εφαρμογής, παρέχει μια αξιόπιστη βάση για την ανταλλαγή πληροφοριών.

3.5 Σχεδιασμός Στοιχείου Server

Ο C2 Server αποτελεί την κεντρική μονάδα ελέγχου και διαχείρισης του εκπαιδευτικού πλαισίου. Υλοποιημένος σε Python, συνδυάζει τη λειτουργικότητα ακρόασης για συνδέσεις από agents με την παροχή μιας web διεπαφής για τον χειριστή. Ο σχεδιασμός του περιλαμβάνει διακριτά λογικά μέρη, τα οποία αναλύονται στις παρακάτω υποενότητες.

3.5.1 Νήμα Ακρόασης (Listener Thread)

Για την ταυτόχρονη εξυπηρέτηση των agents και της web διεπαφής, η λειτουργία ακρόασης για συνδέσεις agents εκτελείται σε ένα αυτόνομο νήμα (thread), ανεξάρτητο από το κύριο νήμα που εκτελεί την εφαρμογή Flask. Κατά την εκκίνηση του server, δημιουργείται και ξεκινά αυτό το νήμα, το οποίο αναλαμβάνει τις εξής εργασίες:

1. Δημιουργία και Δέσμευση Socket: Χρησιμοποιώντας τη βιβλιοθήκη `socket`, δημιουργείται ένα TCP socket (τύπου `AF_INET`, `SOCK_STREAM`). Αυτό το socket ρυθμίζεται κατάλληλα (π.χ. με `SO_REUSEADDR` για άμεση επαναχρησιμοποίηση της διεύθυνσης) και δεσμεύεται (bind) στη διεύθυνση IP του server (συνήθως `0.0.0.0` για αποδοχή συνδέσεων από οποιαδήποτε διεπαφή) και στην προκαθορισμένη θύρα ακρόασης (στην υλοποίηση, η θύρα 9999).

2. Έναρξη Ακρόασης: Το socket τίθεται σε κατάσταση ακρόασης (`listen()`), έτοιμο να δεχτεί εισερχόμενες συνδέσεις TCP από τους agents, με έναν καθορισμένο μέγιστο αριθμό συνδέσεων στην ουρά αναμονής.

3. Βρόχος Αποδοχής Συνδέσεων: Το νήμα εισέρχεται σε έναν βρόχο που ελέγχεται από ένα `shutdown_event`, όπου η εκτέλεση μπλοκάρει περιοδικά (λόγω του timeout στο socket) στη μέθοδο `accept()`, περιμένοντας μια νέα σύνδεση.

4. Δημιουργία Νήματος Handler: Μόλις ένας agent συνδεθεί και η κλήση `accept()` επιστρέψει ένα νέο socket για τη συγκεκριμένη σύνδεση και τη διεύθυνση του agent, το νήμα του listener

δημιουργεί αμέσως ένα νέο νήμα εκτέλεσης. Σε αυτό το νέο νήμα ανατίθεται η εκτέλεση της συνάρτησης `handle_client` (που περιγράφεται παρακάτω), περνώντας της ως ορίσματα το νέο (απλό TCP) socket και τη διεύθυνση του agent.

5. Συνέχιση Ακρόασης: Αφού δημιουργήσει το νήμα για τον agent handler, το νήμα του listener επιστρέφει αμέσως στον βρόχο και στην αναμονή της επόμενης σύνδεσης.

Αυτή η πολυνηματική (multi-threaded) προσέγγιση επιτρέπει στον server να παραμένει αποκριτικός και να διαχειρίζεται ταυτόχρονα πολλαπλούς agents χωρίς να μπλοκάρει η κύρια λειτουργία ακρόασης.

3.5.2 Διαχειριστής Πράκτορα (Agent Handler)

Η συνάρτηση `handle_client` είναι υπεύθυνη για την πλήρη διαχείριση της επικοινωνίας με έναν μεμονωμένο agent, από τη στιγμή που η σύνδεσή του γίνεται αποδεκτή από τον listener μέχρι τον τερματισμό της. Κάθε εκτέλεση της `handle_client` γίνεται στο δικό της νήμα. Η λογική της περιλαμβάνει τα εξής βήματα:

1. Αρχικοποίηση Σύνδεσης: Το απλό TCP socket που παρέχεται από τον listener ρυθμίζεται με ένα χρονικό όριο για τις δικτυακές λειτουργίες. Καταγράφεται η σύνδεση του νέου client στις καθολικές δομές δεδομένων και προετοιμάζονται οι αρχικές εντολές συλλογής πληροφοριών (`whoami`, `hostname`, πληροφορίες έκδοσης λειτουργικού συστήματος) για τον νέο agent.

2. Κύριος Βρόχος Επικοινωνίας: Η συνάρτηση εισέρχεται σε έναν βρόχο που εκτελείται όσο η σύνδεση είναι ενεργή και δεν έχει σηματοδοτηθεί ο τερματισμός του server.

   - Έλεγχος και Αποστολή Εντολής: Ελέγχεται η ουρά εντολών για τον συγκεκριμένο agent. Αν υπάρχει διαθέσιμη εντολή, μορφοποιείται με το Command ID και αποστέλλεται στον agent, αφού πρώτα κρυπτογραφηθεί με AES.
   - Λήψη και Αποκρυπτογράφηση Απάντησης: Ο handler προσπαθεί να λάβει δεδομένα από τον agent. Τα ληφθέντα bytes που αναμένονται να είναι κρυπτογραφημένα, αποκρυπτογραφούνται με AES και η λήψη γίνεται με χρονικό όριο για την αποφυγή ατέρμονης αναμονής.
   - Ανάλυση και Επεξεργασία Απάντησης: Το αποκρυπτογραφημένο μήνυμα αναλύεται με βάση τη μορφή `CID:<uuid>:<TYPE>:<data>`. Ανάλογα με τον τύπο (`RESP` ή `ERR`), ενημερώνεται η κατάσταση της αντίστοιχης εντολής στο `command_status` και αποθηκεύεται το αποτέλεσμα ή το σφάλμα. Ειδικός χειρισμός γίνεται για τις απαντήσεις των αρχικών εντολών (για την ενημέρωση των `client_details`) και για τις απαντήσεις ειδικών εντολών όπως screenshot, download όπου αποθηκεύονται και τα σχετικά αρχεία στον server).

3. Τερματισμός Σύνδεσης: Ο βρόχος τερματίζεται αν η σύνδεση διακοπεί, αν παρουσιαστεί μη ανακτήσιμο σφάλμα, ή αν ο server σηματοδοτήσει τερματισμό. Στον `finally` block, το socket κλείνει και οι πόροι που σχετίζονται με τον client αποδεσμεύονται από τις καθολικές δομές. Οι εντολές που εκκρεμούσαν για τον client αυτόν μαρκάρονται ως αποτυχημένες.

4. Χειρισμός Σφαλμάτων: Σε όλη τη διάρκεια εκτέλεσης, υπάρχουν `try...except` blocks για τον εντοπισμό και τη διαχείριση πιθανών σφαλμάτων δικτύου, κρυπτογράφησης/αποκρυπτογράφησης, ή λογικής.

3.5.3 Διαχείριση Δεδομένων

Ο C2 server χρειάζεται να διατηρεί την κατάσταση σχετικά με τους συνδεδεμένους agents, τις εντολές που τους έχουν ανατεθεί και τα αποτελέσματα που έχουν επιστρέψει. Στην παρούσα υλοποίηση, αυτή η διαχείριση κατάστασης γίνεται χρησιμοποιώντας δομές δεδομένων της Python που αποθηκεύονται στη μνήμη:

- `clients` (Λεξικό - Dictionary): Αποθηκεύει τα ενεργά αντικείμενα socket για κάθε συνδεδεμένο client. Το κλειδί είναι το `client_id` (ένας ακέραιος που αυξάνεται) και η τιμή είναι το αντικείμενο `socket` της σύνδεσης.
- `client_ips` (Λεξικό - Dictionary): Συσχετίζει το `client_id` με τη διεύθυνση IP του αντίστοιχου client. Κλειδί: `client_id`, Τιμή: συμβολοσειρά IP.
- `client_last_seen` (Λεξικό - Dictionary): Καταγράφει τη χρονοσφραγίδα (timestamp) της τελευταίας φοράς που λήφθηκαν δεδομένα από κάθε client. Κλειδί: `client_id`, Τιμή: timestamp (float).
- `client_details` (Λεξικό - Dictionary): Αποθηκεύει βασικές πληροφορίες για κάθε client, όπως λειτουργικό σύστημα, hostname και όνομα χρήστη, οι οποίες συλλέγονται μέσω των αρχικών εντολών. Κλειδί: `client_id`, Τιμή: λεξικό με τις λεπτομέρειες.
- `command_queues` (Λεξικό - Dictionary): Διατηρεί μια ουρά αναμονής εντολών για κάθε ενεργό client. Το κλειδί είναι το `client_id` και η τιμή είναι ένα αντικείμενο `queue.Queue()`, όπου τοποθετούνται οι εντολές που προορίζονται για τον συγκεκριμένο client.
- `command_status` (Λεξικό - Dictionary): Η κεντρική δομή για την παρακολούθηση της κατάστασης όλων των εντολών που έχουν εκδοθεί. Το κλειδί είναι το μοναδικό `command_id` (UUID string) της εντολής. Η τιμή είναι ένα άλλο λεξικό που περιέχει λεπτομέρειες για την εντολή, όπως:
  - `status`: Η τρέχουσα κατάσταση ('queued', 'sent', 'completed', 'error').
  - `timestamp`: Η χρονοσφραγίδα της τελευταίας αλλαγής κατάστασης.
  - `client_id`: Το ID του client στον οποίο στάλθηκε η εντολή.
  - `command`: Η αρχική συμβολοσειρά της εντολής.
  - `response`: Τα δεδομένα της απάντησης από τον agent (αν `status='completed'`).
  - `error`: Το μήνυμα σφάλματος από τον agent (αν `status='error'`).
  - `data_type`: Ένας δείκτης για τον τύπο των δεδομένων στην απάντηση («text», «screenshot _base64_png», «browse_json», «file_download_complete», «upload_confirmation», «upload _command», «initial_discovery»).

– `server_filepath` (προαιρετικά): Η διαδρομή στο filesystem του server όπου αποθηκεύτηκε ένα αρχείο που κατέβηκε από τον agent ή ένα στιγμιότυπο οθόνης.

Καθώς πολλαπλά νήματα (agent handlers και Flask request handlers) μπορεί να προσπαθήσουν να διαβάσουν ή να γράψουν σε αυτές τις δομές ταυτόχρονα, η πρόσβαση προστατεύεται με τη χρήση δύο αντικειμένων `threading.Lock`: το `clients_lock` για τις δομές που σχετίζονται άμεσα με τους clients και τις ουρές εντολών τους, και το `status_lock` για το λεξικό `command_status` και το `client_details`. Οποιαδήποτε λειτουργία τροποποιεί ή διαβάζει από αυτές τις δομές γίνεται εντός ενός `with <lock_name>:` block, διασφαλίζοντας την αμοιβαία αποκλειστικότητα (mutual exclusion) και αποτρέποντας καταστάσεις ανταγωνισμού (race conditions). Αυτή η προσέγγιση διαχείρισης δεδομένων στη μνήμη, αν και απλή, έχει το μειονέκτημα ότι όλη η κατάσταση χάνεται αν ο server τερματιστεί.

3.5.4 Endpoints του Flask API

Η διεπαφή διαχείρισης του C2 παρέχεται μέσω μιας web εφαρμογής που υλοποιήθηκε με το Flask. Το Flask εκθέτει συγκεκριμένα HTTP endpoints (διαδρομές URL) που επιτρέπουν την αλληλεπίδραση με το C2 σύστημα:

- `/` (GET): Εξυπηρετεί την κύρια σελίδα HTML της διεπαφής χρήστη (`index.html`).
- `/clients` (GET): Επιστρέφει μια λίστα των συνδεδεμένων agents και των λεπτομερειών τους σε μορφή JSON.
- `/send_command` (POST): Δέχεται δεδομένα JSON (`client_id`, `command`) για την ανάθεση νέας εντολής σε έναν agent.
- `/command_status/<command_id>` (GET): Επιστρέφει την κατάσταση και το αποτέλεσμα (αν υπάρχει) μιας συγκεκριμένης εντολής σε μορφή JSON.
- `/download_file/<command_id>` (GET): Επιτρέπει τη λήψη ενός αρχείου που έχει μεταφερθεί από τον agent στον server, συσχετισμένου με μια εντολή `download`.
- `/upload_to_agent` (POST): Δέχεται ένα αρχείο και πληροφορίες (`client_id`, `target_path`) για τη μεταφόρτωση ενός αρχείου από τον χειριστή στον επιλεγμένο agent.
- `/api/screenshot_data/<command_id>` (GET): Επιστρέφει τα δεδομένα Base64 ενός στιγμιότυπου οθόνης.

Αυτά τα endpoints επιτρέπουν στη Javascript του frontend να ανακτά δυναμικά πληροφορίες και να υποβάλλει εντολές.

3.6 Σχεδιασμός Στοιχείου Agent

Ο Agent (Πράκτορας) αποτελεί το στοιχείο του C2 framework που εκτελείται στο παραβιασμένο σύστημα-στόχο. Στην παρούσα υλοποίηση, ο agent αναπτύχθηκε ως ένα script Powershell, ειδικά σχεδιασμένο για

λειτουργικά συστήματα Windows. Ο σχεδιασμός του επικεντρώθηκε στην υλοποίηση της λογικής επικοινωνίας με τον C2 Server, τη λήψη και εκτέλεση εντολών, και την επιστροφή των αποτελεσμάτων, με τα δεδομένα της επικοινωνίας να κρυπτογραφούνται με AES.

3.6.1 Κύριος Βρόχος και Σύνδεση

Η καρδιά του agent είναι ένας ατέρμων βρόχος (`while ($true)`), ο οποίος διαχειρίζεται τη συνεχή προσπάθεια σύνδεσης και την επικοινωνία με τον C2 server.

1. **Προσπάθεια Σύνδεσης TCP:** Σε κάθε επανάληψη του εξωτερικού βρόχου, ο agent επιχειρεί να εγκαθιδρύσει μια απλή TCP σύνδεση με τον C2 server στην προκαθορισμένη διεύθυνση IP και θύρα. Χρησιμοποιείται το `System.Net.Sockets.TcpClient` και η μέθοδος `ConnectAsync()` με χρονικό όριο. Γίνονται πολλαπλές προσπάθειες σύνδεσης αν η αρχική αποτύχει.

2. **Λήψη NetworkStream:** Μετά την επιτυχή TCP σύνδεση, λαμβάνεται η ροή δεδομένων δικτύου (`NetworkStream`) από το `TcpClient`. Σε αυτή τη ροή ορίζεται ένα χρονικό όριο ανάγνωσης (`ReadTimeout`) για τις επόμενες λειτουργίες.

3. **Εσωτερικός Βρόχος Επικοινωνίας:** Μόλις η σύνδεση είναι ενεργή, ο agent εισέρχεται σε έναν εσωτερικό βρόχο (`while ($tcpClient.Connected)`) όπου:

   - Λαμβάνει και Αποκρυπτογραφεί Εντολή: Καλεί τη συνάρτηση `Receive-TcpMessage`, η οποία διαβάζει το πρόθεμα μήκους, λαμβάνει τα κρυπτογραφημένα bytes και τα αποκρυπτογραφεί χρησιμοποιώντας AES για να πάρει την εντολή από τον server.
   - Επεξεργάζεται την Εντολή: Αναλύει την αποκρυπτογραφημένη εντολή στην μορφή `CID:uuid:CMD:command_string` και την εκτελεί.
   - Κρυπτογραφεί και Στέλνει Απάντηση: Το αποτέλεσμα ή το σφάλμα της εντολής μορφοποιείται σε μορφή `CID:uuid:RESP/ ERR:data`, κρυπτογραφείται με AES, και αποστέλλεται πίσω στον server μέσω της συνάρτησης `Send-TcpMessage`.
   - Ο εσωτερικός βρόχος συνεχίζεται, περιμένοντας την επόμενη εντολή. Αν η σύνδεση διακοπεί ή παρουσιαστεί μη ανακτήσιμο σφάλμα, ο εσωτερικός βρόχος τερματίζεται.

4. **Καθαρισμός και Επανάληψη:** Αν ο εσωτερικός βρόχος τερματιστεί, γίνεται καθαρισμός της σύνδεσης (κλείσιμο του stream και του TcpClient). Ο εξωτερικός βρόχος εισάγει μια περίοδο αδράνειας (`$ReconnectLoopDelaySeconds`) και στη συνέχεια επιχειρεί ξανά να συνδεθεί στον server.

Αυτή η δομή επιτρέπει στον agent να προσπαθεί συνεχώς να διατηρεί επικοινωνία με τον server και να επεξεργάζεται τις εντολές που λαμβάνει. Δεν υπάρχει πλέον η έννοια του «beacon» μηνύματος που στέλνεται προληπτικά από τον agent. Εδώ, ο agent απλά περιμένει εντολές και απαντά σε αυτές.

3.6.2 Ανάλυση και Εκτέλεση Εντολών

Αφού ο agent λάβει και αποκρυπτογραφήσει ένα μήνυμα εντολής από τον server, προχωρά στην ανάλυση και εκτέλεσή της. Η λογική αυτή είναι ενσωματωμένη στον εσωτερικό βρόχο επικοινωνίας:

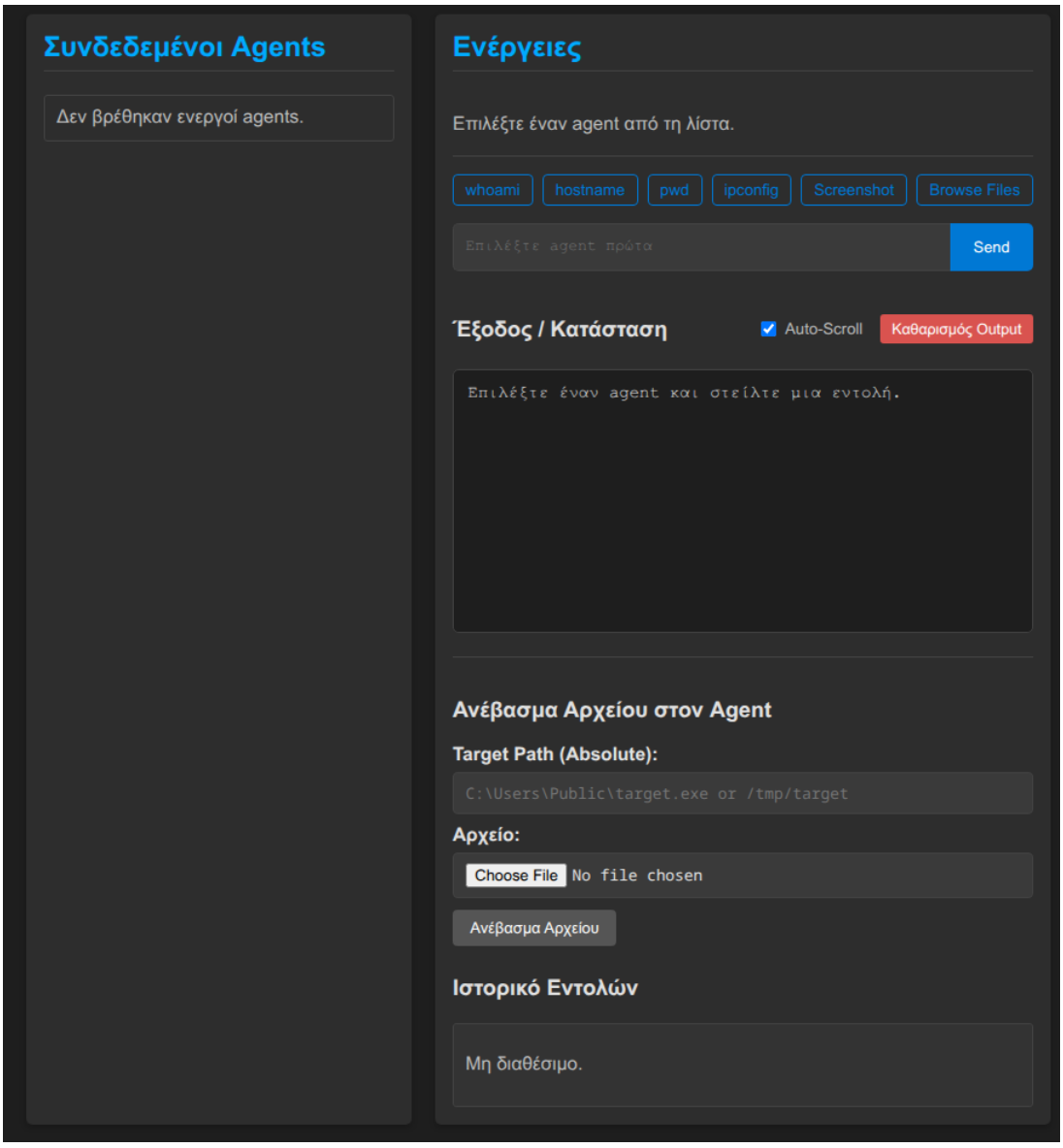
1. Ανάλυση Μορφοποίησης: Το αποκρυπτογραφημένο μήνυμα αναλύεται για να εξαχθεί το Command ID (`$commandId`) και η πραγματική εντολή (`$commandText`).
2. Διαχωρισμός Ενέργειας και Ορισμάτων: Η `$commandText` διαχωρίζεται στην κύρια ενέργεια (`$firstWord`, για παράδειγμα, `whoami`, `screenshot`, `browse`) και στα υπόλοιπα ορίσματά της (αν υπάρχουν, για παράδειγμα η διαδρομή για το `browse` ή `download`).
3. Επιλεκτική Εκτέλεση:
   - **Ειδικές Εντολές:** Ελέγχεται αν το `$firstWord` αντιστοιχεί σε μια από τις προκαθορισμένες ειδικές ενέργειες (`screenshot`, `browse`, `download`, `upload`). Αν ναι, καλείται η αντίστοιχη, ειδικά γραμμένη συνάρτηση Powershell (`Get-ScreenshotAsBase64`, `Get-DirectoryListingAsJson`, `Get-FileAsBase64`, `Save-FileFromBase64`). Αυτές οι συναρτήσεις χειρίζονται τη συγκεκριμένη λειτουργικότητα και επιστρέφουν το αποτέλεσμα ή διαχειρίζονται τυχόν σφάλματα.
   - **Γενικές Εντολές (Fallback σε Invoke-Expression):** Αν το `$firstWord` δεν ταιριάζει με καμία ειδική εντολή, τότε ολόκληρη η `$commandText` εκτελείται χρησιμοποιώντας το `& $scriptBlock` (όπου το `$scriptBlock` δημιουργείται από την `$commandText`). Η έξοδος (stdout και stderr) συλλέγεται.
4. Χειρισμός Σφαλμάτων Εκτέλεσης: Για κάθε τύπο εκτέλεσης (ειδική ή γενική), τυχόν εξαιρέσεις (exceptions) ή μη μηδενικοί κωδικοί εξόδου (`$LASTEXITCODE` για γενικές εντολές) καταγράφονται ως σφάλμα.
5. Προετοιμασία και Αποστολή Απάντησης: Το αποτέλεσμα της εκτέλεσης (το `$output`) ή το μήνυμα σφάλματος (`$executionError`) χρησιμοποιείται για τη διαμόρφωση του μηνύματος απάντησης με τη σωστή μορφή (`CID:<uuid>:RESP:<data>` ή `CID:<uuid>:ERR:<error_message>`). Αυτό το μήνυμα απάντησης στη συνέχεια κρυπτογραφείται με AES και αποστέλλεται πίσω στον server μέσω της συνάρτησης `Send-TcpMessage`.

Αυτή η προσέγγιση, με τον αρχικό έλεγχο για ειδικές εντολές και την εκτέλεση γενικών εντολών ως fallback, παρέχει μια ισορροπία μεταξύ της παροχής εξειδικευμένης λειτουργικότητας και της ευελιξίας για την εκτέλεση αυθαίρετων εντολών Powershell.

3.7 Σχεδιασμός Διεπαφής Ιστού (Web Interface)

Για την παροχή μιας εύχρηστης και οπτικά κατανοητής μεθόδου διαχείρισης του C2 framework από τον χειριστή, αναπτύχθηκε μια διεπαφή ιστού. Η διεπαφή αυτή εξυπηρετείται από τον ενσωματωμένο Flask web server και είναι προσβάσιμη μέσω ενός τυπικού προγράμματος περιήγησης ιστού. Ο σχεδιασμός

της εστιάστηκε στην παροχή των βασικών λειτουργιών που απαιτούνται για την παρακολούθηση και τον έλεγχο των agents, καθώς και στην υποστήριξη των νέων δυνατοτήτων όπως η λήψη στιγμιότυπων οθόνης, η περιήγηση αρχείων και η μεταφόρτωση/ λήψη αρχείων. Μια συνολική εικόνα της διεπαφής παρουσιάζεται στο Σχήμα 3.2.



Σχήμα 3.2: Η γραφική διεπαφή ιστού.

Η δομή της διεπαφής ορίστηκε σε ένα αρχείο HTML (`index.html`), το οποίο χρησιμοποιεί βασικά στοιχεία HTML για την παρουσίαση των πληροφοριών. Η σελίδα χωρίζεται λογικά σε κύριες περιοχές:

1. **Λίστα Συνδεδεμένων Agents:** Μια περιοχή όπου εμφανίζεται ένας πίνακας με τους agents που έχουν ενεργή σύνδεση ή έχουν επικοινωνήσει πρόσφατα με τον server. Ο πίνακας περιλαμβάνει στήλες για το μοναδικό αναγνωριστικό του agent (`client_id`), τη διεύθυνση IP του, βασικές πληροφορίες για το λειτουργικό του σύστημα, το όνομα χρήστη και το hostname (όταν αυτές οι πληροφορίες είναι διαθέσιμες από τις αρχικές εντολές), την ώρα της τελευταίας επικοινωνίας

(`last_seen`) καθώς και την κατάστασή του (connected, unresponsive, stale).

2. **Περιοχή Αλληλεπίδρασης Εντολών:** Όταν ένας agent επιλέγεται από τη λίστα, ενεργοποιείται αυτή η περιοχή. Περιλαμβάνει προκαθορισμένα κουμπιά για συχνές και χρήσιμες εντολές (`whoami`, `screenshot`, `browse files`) και μια φόρμα όπου ο χρήστης μπορεί να εισάγει μια αυθαίρετη εντολή Powershell για τον επιλεγμένο agent.

3. **Περιοχή Εμφάνισης Αποτελεσμάτων/Κατάστασης Εντολών:** Μια κεντρική περιοχή όπου εμφανίζονται τα αποτελέσματα των εντολών που έχουν επιστραφεί από τον επιλεγμένο agent, η κατάσταση των εντολών που εκκρεμούν, καθώς και τα ληφθέντα στιγμιότυπα οθόνης (ως εικόνες).

4. **Περιοχή Περιηγητή Αρχείων (File Browser):** Μια δυναμικά εμφανιζόμενη περιοχή που επιτρέπει την πλοήγηση στο σύστημα αρχείων του επιλεγμένου agent, την εμφάνιση των περιεχομένων των φακέλων (αρχεία και υποφάκελοι με τις ιδιότητές τους) και την έναρξη λήψης αρχείων ή την πλοήγηση σε υποφακέλους.

5. **Περιοχή Μεταφόρτωσης Αρχείου (File Upload):** Μια φόρμα που επιτρέπει στον χειριστή να επιλέξει ένα τοπικό αρχείο και να καθορίσει την πλήρη διαδρομή προορισμού στον agent για τη μεταφόρτωση.

6. **Ιστορικό Εντολών:** Μια περιοχή που εμφανίζει τις πρόσφατες εντολές που έχουν σταλεί, επιτρέποντας την εύκολη επαναχρησιμοποίησή τους.

Η εμφάνιση της διεπαφής μορφοποιείται με χρήση κανόνων CSS, οι οποίοι τοποθετούνται σε ένα ξεχωριστό αρχείο CSS (`style.css`) στον φάκελο `static`. Στόχος της μορφοποίησης ήταν η δημιουργία μιας καθαρής, ευανάγνωστης και λειτουργικής διεπαφής.

Η δυναμική λειτουργικότητα της διεπαφής υλοποιείται με χρήση Javascript κώδικα που εκτελείται στον browser του χρήστη. Αυτός ο κώδικας:

- Επικοινωνεί με το Flask API: Χρησιμοποιεί την ενσωματωμένη συνάρτηση `fetch` για να κάνει ασύγχρονες HTTP κλήσεις στα API endpoints που παρέχει ο Flask server (`/clients`, `/send_command`, `/command_status/ <command_id>`, `/download_file/<command_id>`, `/upload _to_agent`).

- Περιοδική Ανανέωση (Polling): Χρησιμοποιεί τη συνάρτηση `setInterval` για να καλεί περιοδικά το endpoint `/clients`. Για τις εντολές, χρησιμοποιεί polling στο `/command_status/ <command_id>` μέχρι η εντολή να ολοκληρωθεί ή να αποτύχει, ανακτώντας και εμφανίζοντας τα αποτελέσματα.

- Ενημέρωση DOM: Με βάση τα δεδομένα JSON που λαμβάνει, ο Javascript κώδικας ενημερώνει δυναμικά το περιεχόμενο της ιστοσελίδας, τροποποιώντας τον πίνακα των agents, εμφανίζοντας τα νέα αποτελέσματα, τα screenshots, τον περιεχόμενο του file browser, κλπ.

- Χειρισμός Γεγονότων: Διαχειρίζεται γεγονότα όπως το κλικ σε έναν agent, το κλικ στα κουμπιά εντολών, τις ενέργειες στον file browser και τη φόρμα upload.

Συνολικά, ο σχεδιασμός της διεπαφής ιστού στοχεύει στην παροχή μιας απλής αλλά αποτελεσματικής μεθόδου για την οπτικοποίηση της κατάστασης του C2 συστήματος και την αλληλεπίδραση του χειριστή.

3.8 Προκλήσεις Υλοποίησης

Κατά τη διάρκεια της ανάπτυξης του εκπαιδευτικού C2 framework, ανεφύησαν ορισμένες τεχνικές προκλήσεις. Η αντιμετώπισή τους αποτέλεσε σημαντικό μέρος της μαθησιακής διαδικασίας.

Μια αρχική προσέγγιση για την ασφάλεια της επικοινωνίας περιλάμβανε την υλοποίηση του πρωτοκόλλου TLS. Ωστόσο, προέκυψαν σημαντικές δυσκολίες στη σταθερή και αξιόπιστη λειτουργία του TLS handshake και της ανταλλαγής δεδομένων μεταξύ του Python server και του Powershell agent. Παρά τις πολλαπλές προσπάθειες αποσφαλμάτωσης, παρατηρούνταν προβλήματα με τον συγχρονισμό των μηνυμάτων, την ερμηνεία των TLS records από την πλευρά του agent, και την τμηματική λήψη δεδομένων από τον server, οδηγώντας σε συχνές αποσυνδέσεις και αδυναμία ολοκλήρωσης της επικοινωνίας. Δεδομένου του εκπαιδευτικού χαρακτήρα της εργασίας και του περιορισμένου χρόνου, αποφασίστηκε η μετάβαση σε μια απλούστερη προσέγγιση κρυπτογράφησης σε επίπεδο εφαρμογής με χρήση AES και προκαθορισμένου κλειδιού, η οποία, αν και με διαφορετικούς περιορισμούς ασφαλείας, επέτρεψε την επίτευξη του στόχου της εμπιστευτικής επικοινωνίας.

Κατά την υλοποίηση της κρυπτογράφησης AES, μια πρόκληση ήταν η σωστή διαχείριση και μετατροπή των bytes του κλειδιού και του IV μεταξύ Python και Powershell, καθώς και η διασφάλιση της συμβατότητας του padding (PKCS7) μεταξύ της βιβλιοθήκης `PyCryptodome` και των κλάσεων .NET.

Μια άλλη σημαντική πρόκληση αφορούσε την ορθή ανάλυση (parsing) των εντολών και των ορισμάτων τους στον Powershell agent, ειδικά για εντολές που περιείχαν διαδρομές αρχείων με κενά διαστήματα, όπως στην περίπτωση της εντολής `upload`. Η αρχική χρήση της μεθόδου `-split` με περιορισμένο αριθμό τμημάτων αποδείχθηκε ανεπαρκής, οδηγώντας σε λανθασμένη εξαγωγή της διαδρομής και των δεδομένων Base64. Η υιοθέτηση μιας πιο στιβαρής προσέγγισης με χρήση regular expressions για την ανάλυση της εντολής `upload` έλυσε αυτό το πρόβλημα.

Επιπλέον, η επέκταση συμβολοσειρών (string interpolation) στο Powershell, ειδικά όταν μεταβλητές ακολουθούνται από τον χαρακτήρα `:`, απαιτούσε τη χρήση αγκυλών (`${variableName}`) για τη σωστή αναγνώριση της μεταβλητής. Η παράλειψη αυτής της λεπτομέρειας οδηγούσε στη δημιουργία λανθασμένα μορφοποιημένων μηνυμάτων απάντησης από τον agent, προκαλώντας προβλήματα στην ανάλυσή τους από τον server.

Τέλος, ο χειρισμός της λήψης δεδομένων στον server απαιτούσε προσεκτική υλοποίηση του πρωτοκόλλου με πρόθεμα μήκους (length prefixing) για να διασφαλιστεί ότι ολόκληρο το (κρυπτογραφημένο) μήνυμα διαβάζεται σωστά πριν την αποκρυπτογράφηση, αποφεύγοντας προβλήματα με μερικώς ληφθέντα δεδομένα ή λανθασμένη ερμηνεία των ορίων των μηνυμάτων.

Γενικότερα, η μεγαλύτερη πρόκληση αποδείχθηκε η εκμάθηση και η προσπάθεια υλοποίησης τόσο πολλών διαφορετικών αντικειμένων, με το καθένα να έχει το δικό του βάθος και τη δική του δυσκολία. Από τον σχεδιασμό της εφαρμογής ιστού, τον κώδικα Python και τις ξεχωριστές βιβλιοθήκες του, την ποι-

κιλία μεθόδων δικτυακής επικοινωνίας, τις μεθόδους κρυπτογράφησης, έως και την οικειοποίηση του οικοσυστήματος των Windows και την επαφή με την Powershell, οι προκλήσεις σίγουρα ήταν πολλές και η συνολική εμπειρία αρκετά έντονη. Τα πλαίσια C2 είναι εξαιρετικά περίπλοκα και απαιτούν τον συνδυασμό πολλών διαφορετικών θεμελιωδών γνώσεων ανάπτυξης λογισμικού, δικτύων και κυβερνοασφάλειας, ακόμη και για την ανάπτυξη μιας σχετικά απλής υλοποίησης.

Η αντιμετώπιση αυτών των προκλήσεων υπογράμμισε τη σημασία της λεπτομερούς αποσφαλμάτωσης, της κατανόησης των ιδιαιτεροτήτων κάθε γλώσσας προγραμματισμού και των βιβλιοθηκών τους, καθώς και της προσεκτικής σχεδίασης του πρωτοκόλλου επικοινωνίας.

3.9 Ανακεφαλαίωση Κεφαλαίου 3

Το παρόν κεφάλαιο παρουσίασε λεπτομερώς τον σχεδιασμό και τη μεθοδολογία πίσω από την ανάπτυξη του εκπαιδευτικού C2 framework. Ξεκινώντας από τους στόχους σχεδιασμού που έδιναν έμφαση στην εκπαιδευτική αξία και τη βασική λειτουργικότητα, περιγράφηκε η κεντροποιημένη αρχιτεκτονική Server-Agent. Αιτιολογήθηκε η επιλογή της τεχνολογικής στοίβας, συμπεριλαμβανομένων των Python/Flask για τον server, Powershell για τον agent, και της κρυπτογράφησης AES σε επίπεδο εφαρμογής για την ασφαλή επικοινωνία. Αναλύθηκε το πρωτόκολλο επικοινωνίας που βασίζεται σε πρόθεμα μήκους για τα κρυπτογραφημένα δεδομένα και μια δομημένη μορφή για τα αποκρυπτογραφημένα μηνύματα εφαρμογής. Τέλος, παρουσιάστηκε ο σχεδιασμός των επιμέρους στοιχείων – του πολυνηματικού server με τον Agent Listener και το Flask API/UI, του agent με τον κύριο βρόχο σύνδεσης/επικοινωνίας και τη λογική εκτέλεσης εντολών, και της διεπαφής ιστού – καθώς και ορισμένες από τις υλοποιητικές προκλήσεις που αντιμετωπίστηκαν και οι λύσεις που υιοθετήθηκαν. Έχοντας περιγράψει το «πώς» κατασκευάστηκε το πλαίσιο, το επόμενο κεφάλαιο θα επικεντρωθεί στο «τι μπορεί να κάνει», παρουσιάζοντας τη μελέτη περίπτωσης και την επίδειξη των δυνατοτήτων του.

Κεφάλαιο 4

Μελέτη Περίπτωσης: Επίδειξη Δυνατοτήτων Μετά-Εκμετάλλευσης

Έχοντας αναλύσει το θεωρητικό υπόβαθρο στο κεφάλαιο 2 και τον σχεδιασμό του εκπαιδευτικού C2 framework στο κεφάλαιο 3, το παρόν κεφάλαιο επικεντρώνεται στην πρακτική επίδειξη της λειτουργικότητας και των δυνατοτήτων του. Μέσω μιας σειράς ελεγχόμενων σεναρίων, θα παρουσιαστεί βήμα προς βήμα πώς το πλαίσιο που δημιουργήθηκε μπορεί να χρησιμοποιηθεί για την εκτέλεση βασικών τεχνικών μετά-εκμετάλλευσης σε ένα προσομοιωμένο σύστημα-στόχο. Στόχος αυτής της μελέτης περίπτωσης είναι η οπτικοποίηση των κινδύνων που περιγράφηκαν θεωρητικά και η σύνδεση των λειτουργιών του C2 framework με τις τακτικές και τεχνικές του πλαισίου MITRE ATT&CK. Όλες οι ενέργειες πραγματοποιήθηκαν σε ένα απομονωμένο εργαστηριακό περιβάλλον για εκπαιδευτικούς και ερευνητικούς σκοπούς.

4.1 Ρύθμιση Περιβάλλοντος Δοκιμών

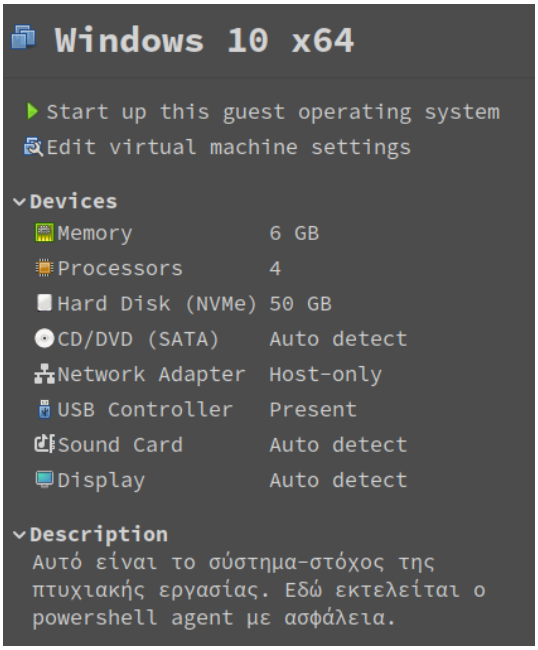
Για την ασφαλή και ελεγχόμενη διεξαγωγή των δοκιμών και της μελέτης περίπτωσης του εκπαιδευτικού C2 framework, δημιουργήθηκε ένα απομονωμένο εργαστηριακό περιβάλλον με χρήση τεχνολογιών εικονικοποίησης (virtualization). Η προσέγγιση αυτή διασφάλισε ότι η όλη δραστηριότητα περιορίστηκε εντός του εργαστηρίου, χωρίς να επηρεαστούν πραγματικά συστήματα ή να εκτεθεί δυνητικά κακόβουλη κίνηση στο διαδίκτυο, σύμφωνα με τις αρχές της ηθικής χρήσης και της εκπαιδευτικής φύσης της εργασίας.

Το περιβάλλον δοκιμών απαρτιζόταν από τα ακόλουθα στοιχεία:

1. **Μηχάνημα Φιλοξενίας (Host Machine)**: Ένας φυσικός υπολογιστής που εκτελούσε το λειτουργικό σύστημα Arch Linux. Αυτός ο υπολογιστής φιλοξενούσε το λογισμικό εικονικοποίησης και λειτουργούσε ως το περιβάλλον εκτέλεσης για τον C2 Server.
    - *Λογισμικό Εικονικοποίησης*: Χρησιμοποιήθηκε το VMware Workstation Pro. Εναλλακτικά, θα μπορούσαν να χρησιμοποιηθούν λύσεις όπως Oracle VirtualBox, QEMU/KVM, ή Hyper-V (σε Windows host).

2. **Εικονική Μηχανή - Στόχος (Target Virtual Machine - VM)**: Μέσα στο VMware Workstation, δημιουργήθηκε μια εικονική μηχανή στην οποία εγκαταστάθηκε το λειτουργικό σύστημα Windows 10 Enterprise LTSC. Αυτή η VM προσομοίωνε το σύστημα-στόχο.

    - *Ρυθμίσεις VM*: Στην εικονική μηχανή διατέθηκαν επαρκείς πόροι (6 GB Μνήμης RAM, 4 vCPUs, 50 GB εικονικός δίσκος).
    - *Αμυντικοί Μηχανισμοί*: Για τη διευκόλυνση των δοκιμών σε αυτό το εκπαιδευτικό πλαίσιο, απενεργοποιήθηκε το Windows Firewall για να μην προκαλέσει προβλήματα στην εξερχόμενη επικοινωνία προς τον C2 server εντός του απομονωμένου δικτύου. Ωστόσο, το Windows Defender, η ενσωματωμένη λύση antivirus των Windows αφέθηκε ενεργοποιημένη καθώς δεν θεώρησε τον agent κάποιου είδους απειλή.



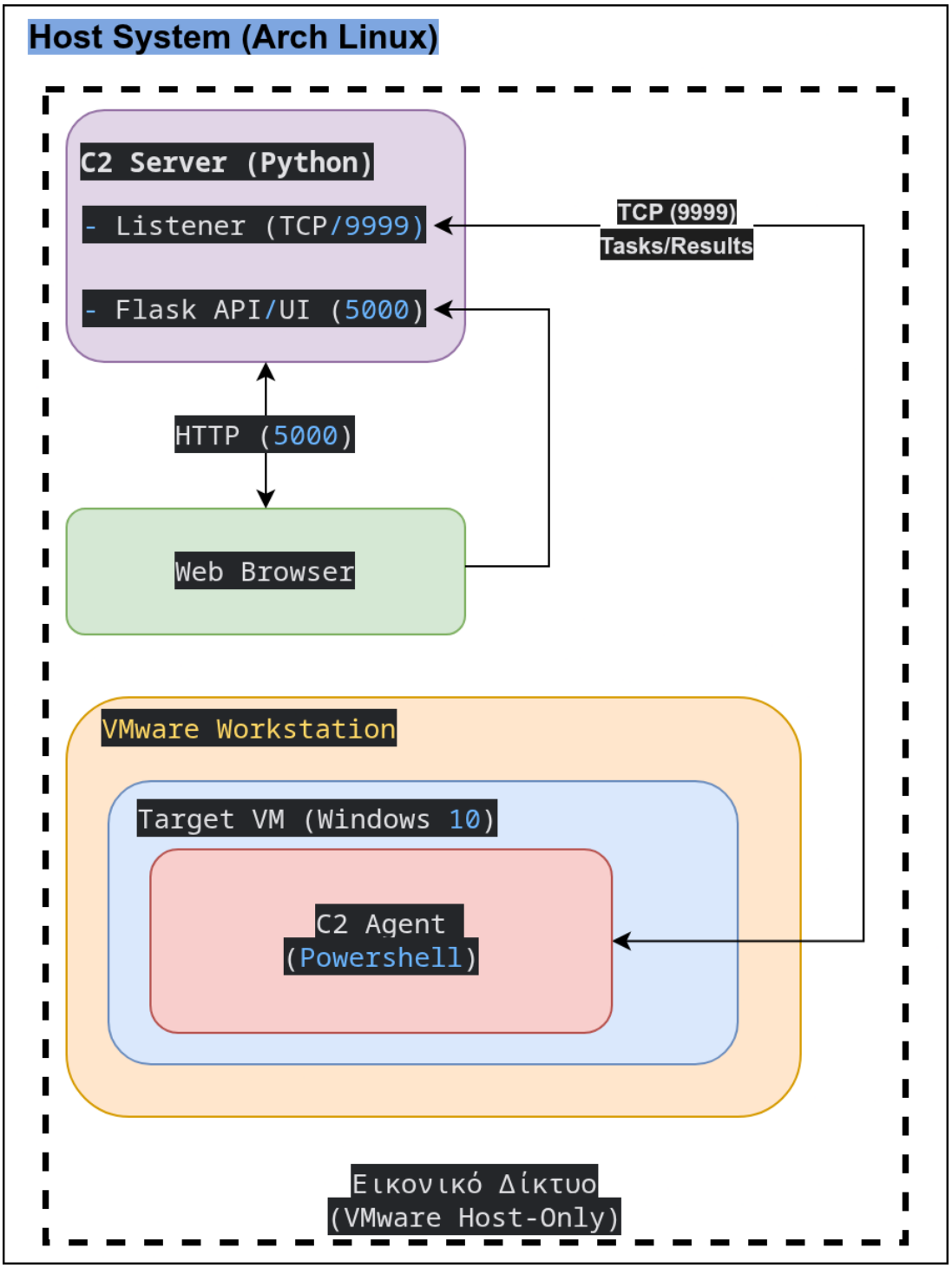
Σχήμα 4.1: Ρυθμίσεις Εικονικής Μηχανής

3. **Εικονικό Δίκτυο (Virtual Network)**: Η δικτύωση μεταξύ του Host Machine και της Target VM διαμορφώθηκε μέσω των δυνατοτήτων εικονικής δικτύωσης του VMware Workstation, χρησιμοποιώντας τη ρύθμιση δικτύου *Host-Only*. Αυτό εξασφάλισε ένα ιδιωτικό δίκτυο, απομονώνοντας την C2 επικοινωνία. Ο C2 server στον host ήταν προσβάσιμος από την VM μέσω της IP διεύθυνσης του host σε αυτό το host-only δίκτυο.

4. **Εκτέλεση Συστατικών C2:**

    - **C2 Server**: Εκτελούνταν απευθείας στο Host Machine (Arch Linux).
    - **C2 Agent**: Το script `agent.ps1` εκτελούνταν εντός της εικονικής μηχανής Windows 10, ρυθμισμένο να συνδέεται στον C2 Server.
    - **Web UI**: Η πρόσβαση στο UI γινόταν από έναν browser στο Host Machine.

Ένα σχηματικό διάγραμμα του περιβάλλοντος δοκιμών παρουσιάζεται στο Σχήμα 4.2.



Σχήμα 4.2: Διάγραμμα Περιβάλλοντος Δοκιμών

Αυτή η ρύθμιση παρείχε ένα ρεαλιστικό, αν και απλοποιημένο, περιβάλλον για την επίδειξη της λειτουργίας του C2 framework.

4.2 Αναλυτική Περιγραφή Σεναρίων

Σε αυτή την ενότητα, θα παρουσιαστούν συγκεκριμένα σενάρια χρήσης του εκπαιδευτικού C2 framework για την εκτέλεση βασικών ενεργειών μετά-εκμετάλλευσης στο σύστημα-στόχο. Κάθε σενάριο θα περιγράφει τα βήματα, τις εντολές, τα αποτελέσματα, και την αντιστοίχισή τους στο MITRE ATT&CK.

4.2.1 Αρχική Σύνδεση και Επίγνωση Κατάστασης

Στόχος: Να επιδειχθεί η αρχική σύνδεση του agent με τον C2 server, η εμφάνισή του στη διεπαφή διαχείρισης, και η αυτόματη συλλογή βασικών πληροφοριών συστήματος μέσω προκαθορισμένων εντολών, ενέργειες που αντιστοιχούν στην τακτική Ανακάλυψης (Discovery - TA0007) του ATT&CK.

Βήματα Εκτέλεσης:

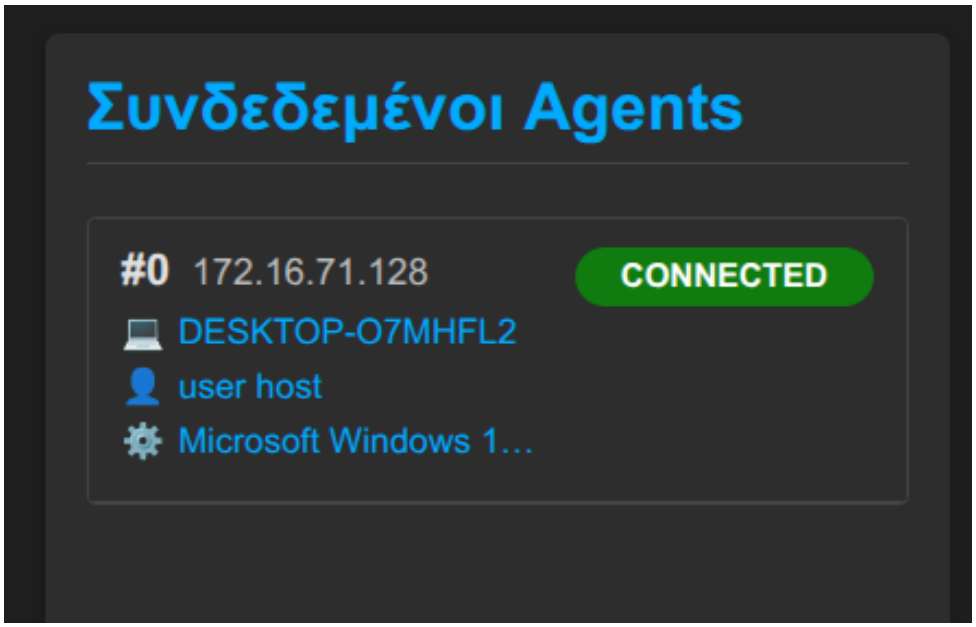
1. **Εκκίνηση Server και Agent:** Ο C2 server (`c2_server.py`) τίθεται σε λειτουργία στο host machine. Στην εικονική μηχανή-στόχο (Windows 10), εκτελείται το Powershell script του agent (`agent.ps1`) με τις κατάλληλες παραμέτρους για τη διεύθυνση IP και τη θύρα του server.

2. **Σύνδεση Agent και Αρχικές Εντολές:** Ο agent συνδέεται στον C2 server μέσω TCP. Ο server αναγνωρίζει τη νέα σύνδεση, εκχωρεί ένα `client_id` και θέτει αυτόματα στην ουρά για τον agent μια σειρά από αρχικές εντολές.

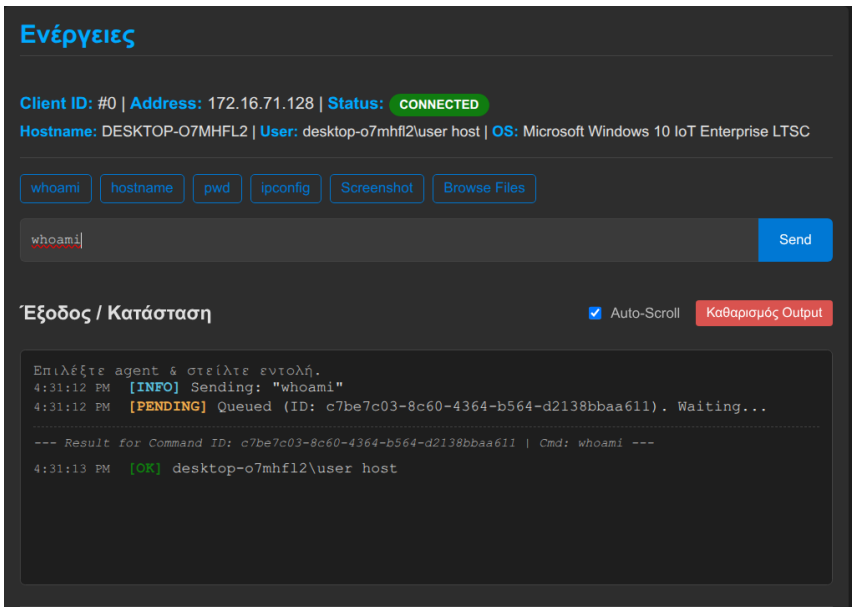
3. **Εμφάνιση στο Web UI:** Στη διεπαφή ιστού, ο νέος agent εμφανίζεται στη λίστα «Active Agents». Αρχικά, οι πληροφορίες συστήματος (OS, User, Hostname) ενδέχεται να είναι κενές.

4. **Επεξεργασία Αρχικών Εντολών:** Ο server αποστέλλει διαδοχικά τις αρχικές εντολές (κρυπτογραφημένες με AES) στον agent. Ο agent τις εκτελεί και επιστρέφει τα αποτελέσματα (επίσης κρυπτογραφημένα). Ο server αποκρυπτογραφεί τις απαντήσεις και ενημερώνει τις λεπτομέρειες του agent (`client_details`) και την κατάσταση των εντολών (`command_status`).

5. **Ενημέρωση Web UI:** Μέσω polling, η διεπαφή ιστού ανανεώνεται και εμφανίζει τις πληροφορίες που συλλέχθηκαν για τον agent (Σχήμα 4.3). Τα αποτελέσματα των επιμέρους αρχικών εντολών είναι επίσης ορατά στην περιοχή αποτελεσμάτων (Σχήμα 4.4).



Σχήμα 4.3: Οι λεπτομέρειες του agent (OS, User, Hostname) ενημερώνονται στο Web UI μετά την εκτέλεση των αρχικών εντολών.



Σχήμα 4.4: Εμφάνιση του αποτελέσματος της εντολής *whoami* στην περιοχή αποτελεσμάτων του Web UI.

Αυτό το σενάριο επιδεικνύει την επιτυχή εγκαθίδρυση της αμφίδρομης, κρυπτογραφημένης (με AES) C2 επικοινωνίας και την αυτοματοποιημένη συλλογή αρχικών πληροφοριών. Ο χειριστής αποκτά άμεσα βασική επίγνωση για το παραβιασμένο σύστημα. Η λήψη του ονόματος χρήστη (T1033: System Owner/User Discovery), του ονόματος του υπολογιστή και της έκδοσης του λειτουργικού συστήματος (T1082: System Information Discovery) αποτελούν θεμελιώδη βήματα για την κατανόηση του στόχου και τον σχεδιασμό περαιτέρω ενεργειών.

4.2.2 Απομακρυσμένη Εκτέλεση Εντολών και Αλληλεπίδραση με το Σύστημα

Στόχος: Να επιδειχθεί η θεμελιώδης δυνατότητα του C2 framework να εκτελεί αυθαίρετες εντολές Powershell στο σύστημα-στόχο, προσομοιώνοντας ενέργειες που θα μπορούσε να κάνει ένας επιτιθέμενος για να αλληλεπιδράσει με το σύστημα, να τροποποιήσει αρχεία, να εκκινήσει διεργασίες ή να συλλέξει περαιτέρω πληροφορίες. Αυτό αντιστοιχεί κυρίως στην τακτική Εκτέλεσης (Execution - TA0002) και δευτερευόντως σε άλλες τακτικές ανάλογα με τη φύση της εντολής.

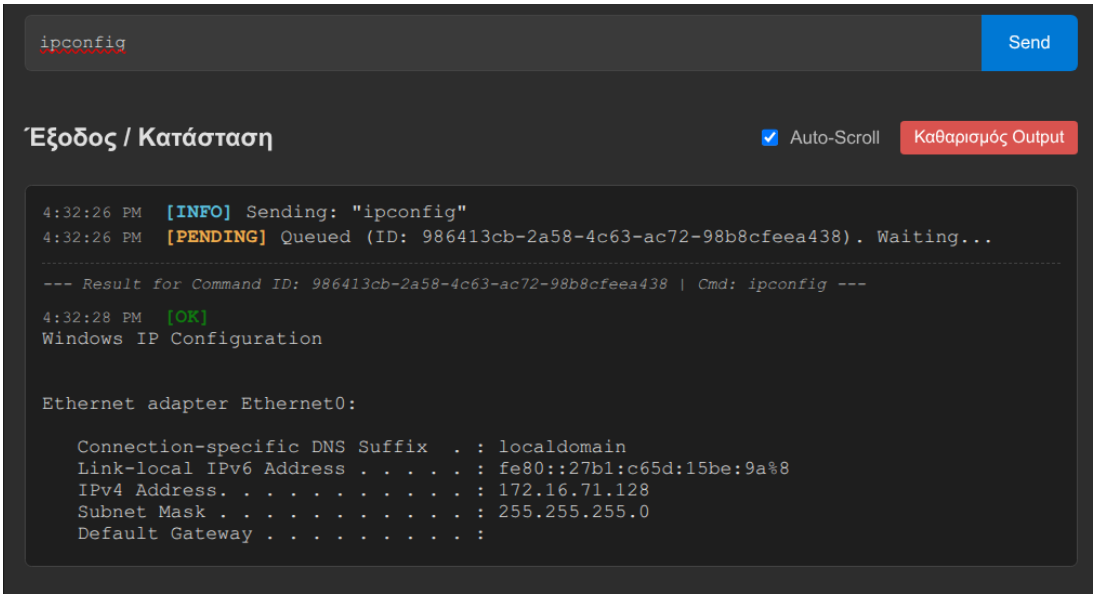
Βήματα Εκτέλεσης:

1. **Επιλογή Agent:** Ο χειριστής επιλέγει τον ενεργό agent από το Web UI, όπως φαίνεται στο Σχήμα 4.3 του προηγούμενου σεναρίου.

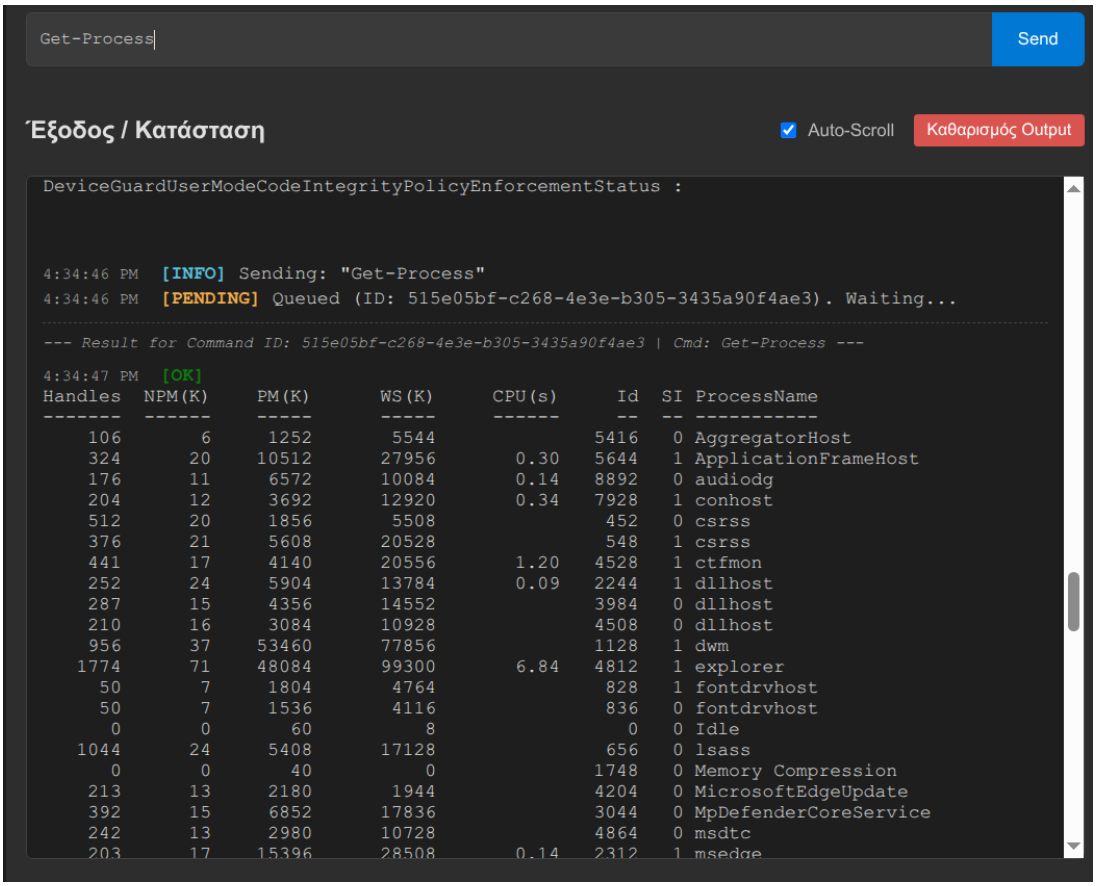
2. **Αποστολή Εντολών Ανακάλυψης:** Ο χειριστής χρησιμοποιεί τη φόρμα ανάθεσης εργασίας για να στείλει διαδοχικά εντολές με σκοπό τη συλλογή πληροφοριών:

   - Αποστέλλεται η εντολή `ipconfig`. Το Web UI, μετά τη λήψη της απάντησης από τον agent, εμφανίζει την έξοδο με τις ρυθμίσεις δικτύου του συστήματος-στόχου (Σχήμα 4.5). Αυτή η ενέργεια αντιστοιχεί στην τεχνική T1016 (System Network Configuration Discovery) του MITRE ATT&CK.

   - Ακολούθως, αποστέλλεται η εντολή `Get-Process` (ή ο alias `ps`). Το Web UI εμφανίζει μια λίστα των διεργασιών που εκτελούνται στην εικονική μηχανή (Σχήμα 4.6), παρέχοντας πληροφορίες που εντάσσονται στην τεχνική T1057 (Process Discovery).



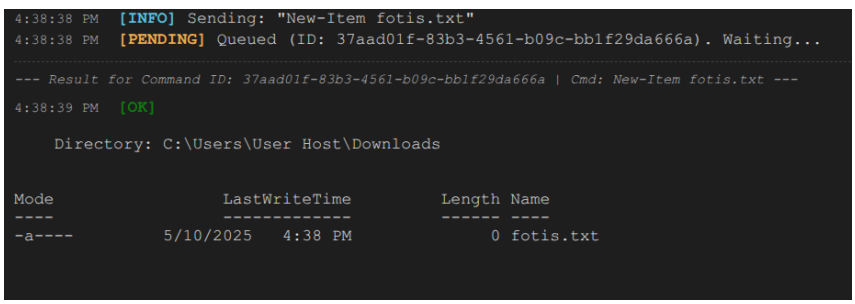
Σχήμα 4.5: Αποτέλεσμα της εντολής *ipconfig*.



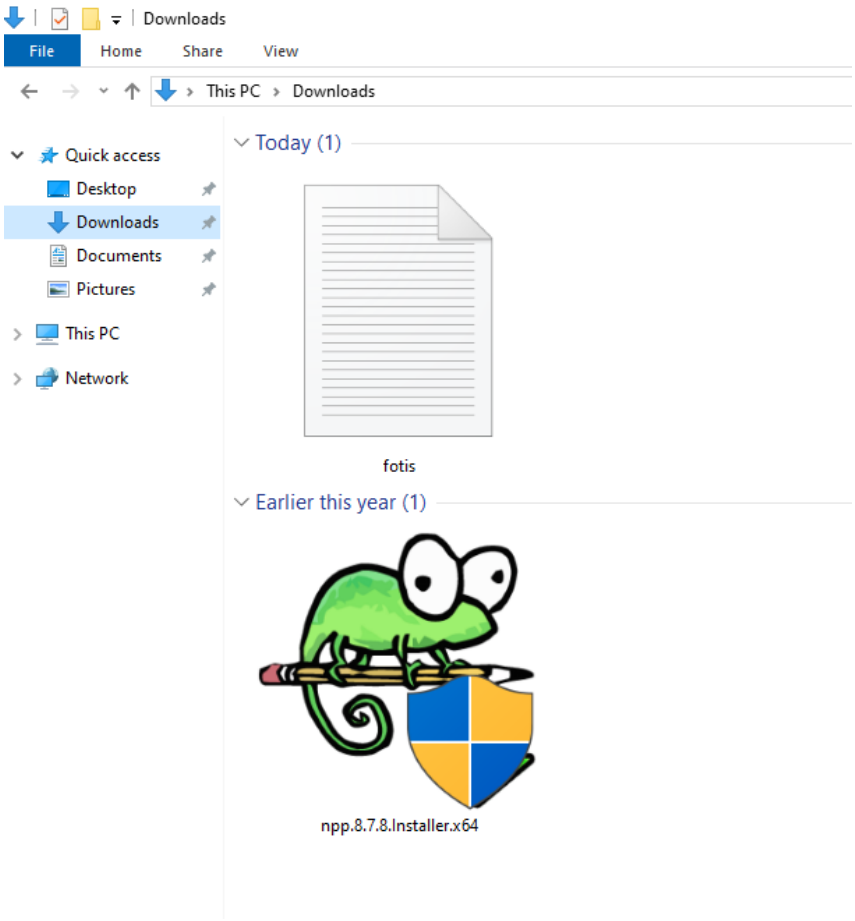
Σχήμα 4.6: Μερική λίστα διεργασιών από την εντολή *Get-Process*.

3. **Αποστολή Εντολών Τροποποίησης Συστήματος Αρχείων:** Για την επίδειξη της δυνατότητας αλληλεπίδρασης με το σύστημα αρχείων:

   - Ο χειριστής μεταφέρεται (με την εντολή *cd*) στον φάκελο *Downloads* του στόχου και έπειτα, στέλνει την εντολή: `New-Item fotis.txt` (Σχήμα 4.7). Εναλλακτικά, θα μπορούσε να ορίσει κατευθείαν το πλήρες μονοπάτι στη *New-Item* χωρίς να κάνει *cd*. Η επιτυχής εκτέλεση επιβεβαιώνεται με την παρατήρηση της δημιουργίας του αρχείου `fotis.txt` στον καθορισμένο φάκελο της εικονικής μηχανής-στόχου (Σχήμα 4.8). Αυτή η ενέργεια αξιοποιεί την τεχνική T1059.001 (Powershell) για την εκτέλεση.

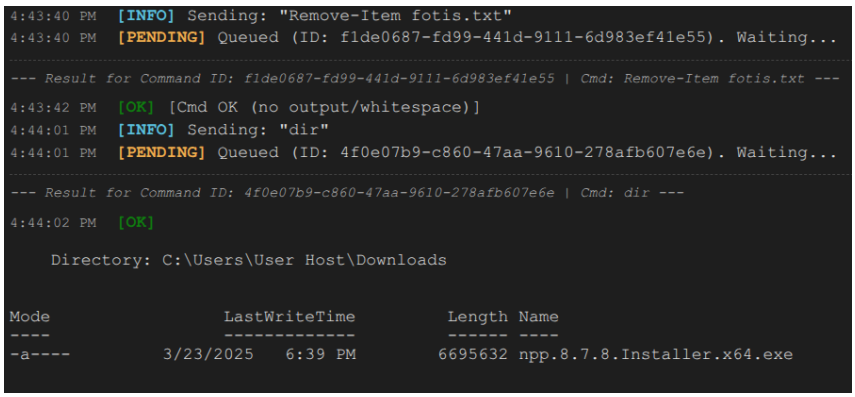


Σχήμα 4.7: Χρήση της *New-Item* για τη δημιουργία αρχείου στο σύστημα.

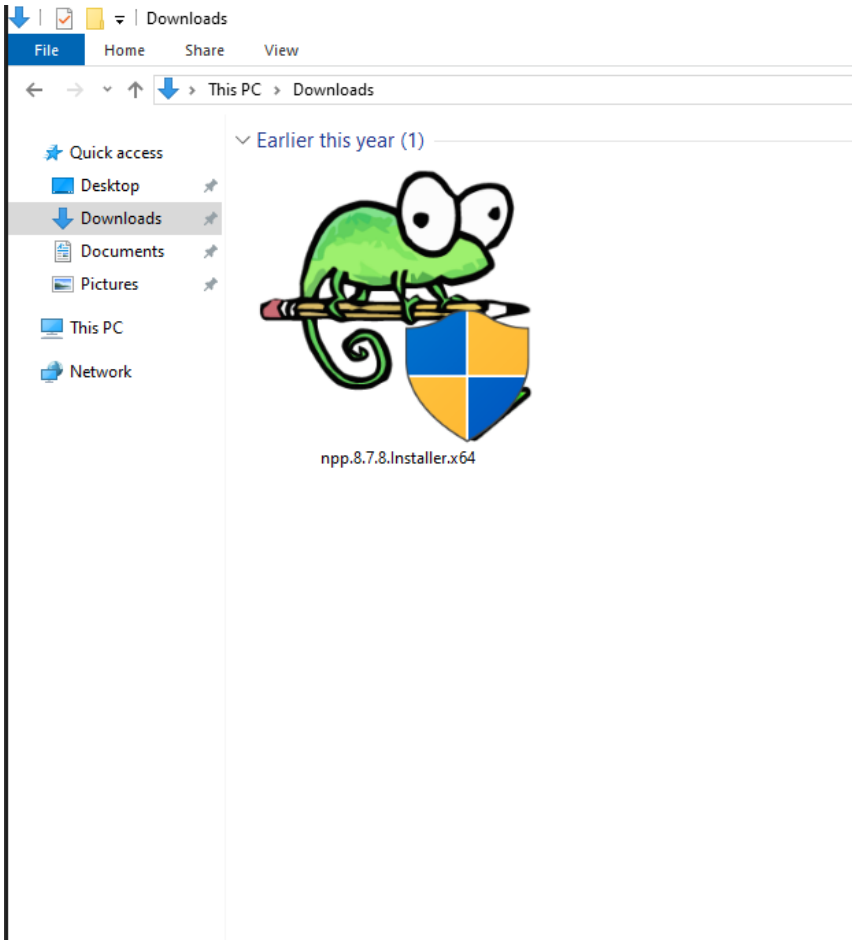


Σχήμα 4.8: Επαλήθευση της δημιουργίας του αρχείου *fotis.txt* μετά την εκτέλεση της εντολής *New-Item*

- Για την επίδειξη της δυνατότητας διαγραφής, αποστέλλεται η εντολή: `Remove-Item fotis.txt`. Η επιτυχία επαληθεύεται ελέγχοντας ότι το αρχείο έχει πράγματι διαγραφεί από την εικονική μηχανή. Αυτή η ενέργεια συνδυάζει την εκτέλεση μέσω PowerShell (T1059.001) με την τεχνική απόκρυψης ιχνών T1070.004 (File Deletion), που εντάσσεται στην τακτική Defense Evasion (TA0005).



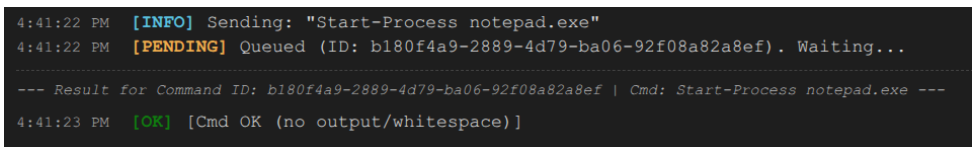
Σχήμα 4.9: Επαλήθευση της διαγραφής του αρχείου *fotis.txt* από την εικονική μηχανή-στόχο μέσω του Web UI.



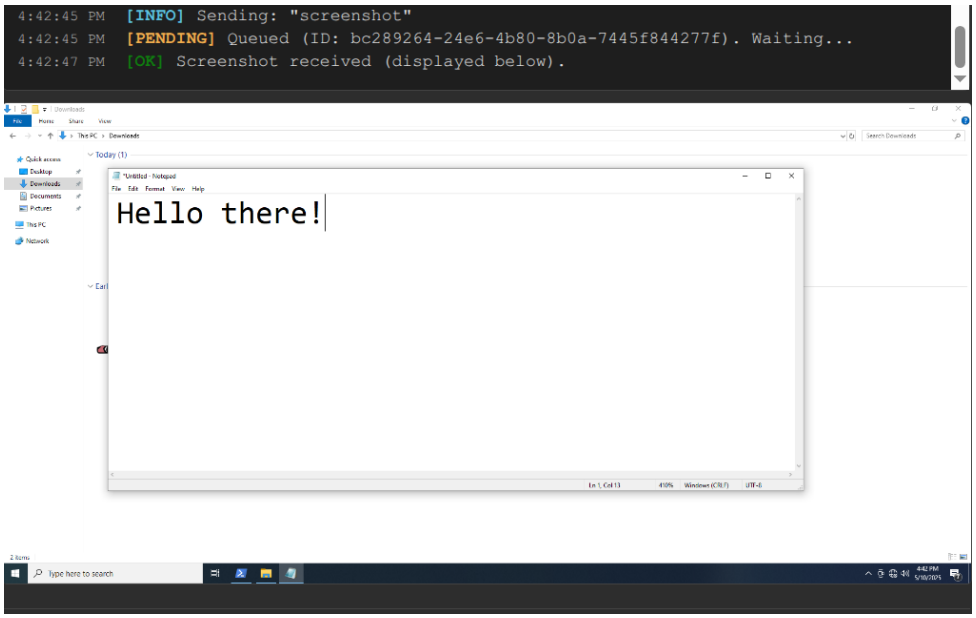
Σχήμα 4.10: Επαλήθευση της διαγραφής του αρχείου *fotis.txt* από την εικονική μηχανή-στόχο.

4. **Αποστολή Εντολής Εκκίνησης Διεργασίας:** Για την επίδειξη της δυνατότητας εκκίνησης εφαρμογών:

   - Ο χειριστής στέλνει την εντολή: `Start-Process notepad.exe`.
   - Το Web UI ενδέχεται να μην εμφανίσει συγκεκριμένη έξοδο για αυτή την εντολή, καθώς η `Start-Process` δεν επιστρέφει δεδομένα στο pipeline υπό αυτές τις συνθήκες. Ωστόσο, η επιτυχής εκτέλεση της εντολής επιβεβαιώνεται οπτικά, παρατηρώντας την εμφάνιση του παραθύρου της εφαρμογής Σημειωματάριο (Notepad) στην επιφάνεια εργασίας της εικονικής μηχανής-στόχου (Σχήμα 4.12).
   - Η εκτέλεση αυτής της εντολής αντιστοιχεί στην τεχνική T1059.001 (PowerShell). Η ίδια η πράξη της εκκίνησης μιας διεργασίας μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς από έναν επιτιθέμενο, όπως η εκτέλεση κακόβουλου λογισμικού ή βοηθητικών εργαλείων.



Σχήμα 4.11: Χρήση της εντολής *Start-Process notepad.exe* για την εκκίνηση της εφαρμογής «Σημειωματάριο» στα Windows.



Σχήμα 4.12: Η εφαρμογή «Σημειωματάριο» (Notepad) πράγματι, έχει εκκινηθεί.

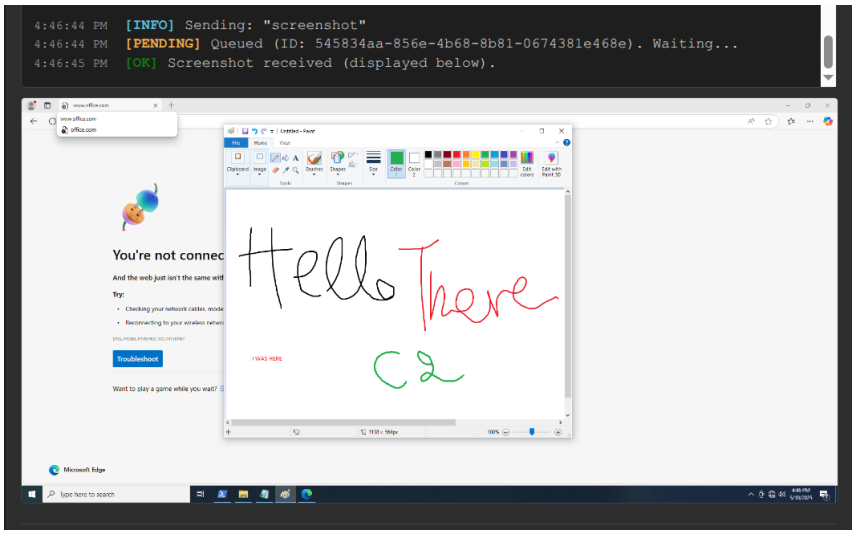
Το σενάριο αυτό καταδεικνύει την ευελιξία του C2 framework στην εκτέλεση εντολών που παρέχουν βαθύτερη κατανόηση του συστήματος αλλά και τη δυνατότητα άμεσης αλληλεπίδρασης και τροποποίησης του συστήματος-στόχου. Συγκεκριμένα, επιδείχθηκε η δημιουργία και η διαγραφή αρχείων, καθώς και η εκκίνηση νέων διεργασιών στο παραβιασμένο σύστημα. Αυτές οι βασικές ενέργειες αποτελούν δομικά στοιχεία για πολύ πιο σύνθετες επιθέσεις. Η δυνατότητα εκτέλεσης αυθαίρετων εντολών PowerShell (T1059.001) είναι αυτή που επιτρέπει στον επιτιθέμενο να προχωρήσει σε ενέργειες όπως η πλευρική κίνηση (lateral movement), η κλιμάκωση δικαιωμάτων (privilege escalation), η περαιτέρω συλλογή δεδομένων, η απόκρυψη ιχνών μέσω της διαγραφής αρχείων (T1070.004), και τελικά η ανάπτυξη κακόβουλου λογισμικού ή άλλων επιθετικών εργαλείων. Η ευκολία με την οποία μπορούν να εκτελεστούν αυτές οι ενέργειες μέσω μιας απλής διαδικτυακής διεπαφής υπογραμμίζει τους κινδύνους που εγκυμονεί ένα παραβιασμένο σύστημα υπό τον έλεγχο ενός πλαισίου C2 .

4.2.3 Συλλογή Οπτικών Δεδομένων (Λήψη Στιγμιότυπου Οθόνης)

Στόχος: Να επιδειχθεί η ικανότητα λήψης στιγμιότυπων οθόνης από το σύστημα-στόχο. Αντιστοιχεί στην τακτική Συλλογής (Collection - TA0009) και την τεχνική T1113 (Screen Capture).

Βήματα Εκτέλεσης:

1. **Αποστολή Εντολής Screenshot:** Ο χειριστής χρησιμοποιεί το κουμπί «screenshot» στο Web UI.
2. **Επεξεργασία και Αποστολή Δεδομένων:** Ο agent εκτελεί τη συνάρτηση `Get-ScreenshotAsBase64`, κωδικοποιεί την εικόνα σε Base64 και την επιστρέφει κρυπτογραφημένη.
3. **Λήψη και Εμφάνιση στη Διεπαφή Ιστού:** Ο server αποκρυπτογραφεί την απάντηση. Το Web UI, ανακτώντας την κατάσταση της εντολής, λαμβάνει τα δεδομένα Base64 και εμφανίζει την εικόνα (Σχήμα 4.13). Ο server αποθηκεύει την εικόνα και τοπικά (Σχήμα 4.14).



Σχήμα 4.13: Εμφάνιση του ληφθέντος στιγμιότυπου οθόνης εντός του Web UI.

Σχήμα 4.14: Το αρχείο PNG του στιγμιότυπου οθόνης όπως αποθηκεύτηκε στον φάκελο «screenshots» του C2 server.

Η λήψη στιγμιότυπων οθόνης παρέχει άμεση οπτική πρόσβαση στο περιβάλλον του χρήστη, αποκαλύπτοντας δυνητικά ευαίσθητες πληροφορίες. Είναι μια ισχυρή τεχνική συλλογής δεδομένων.

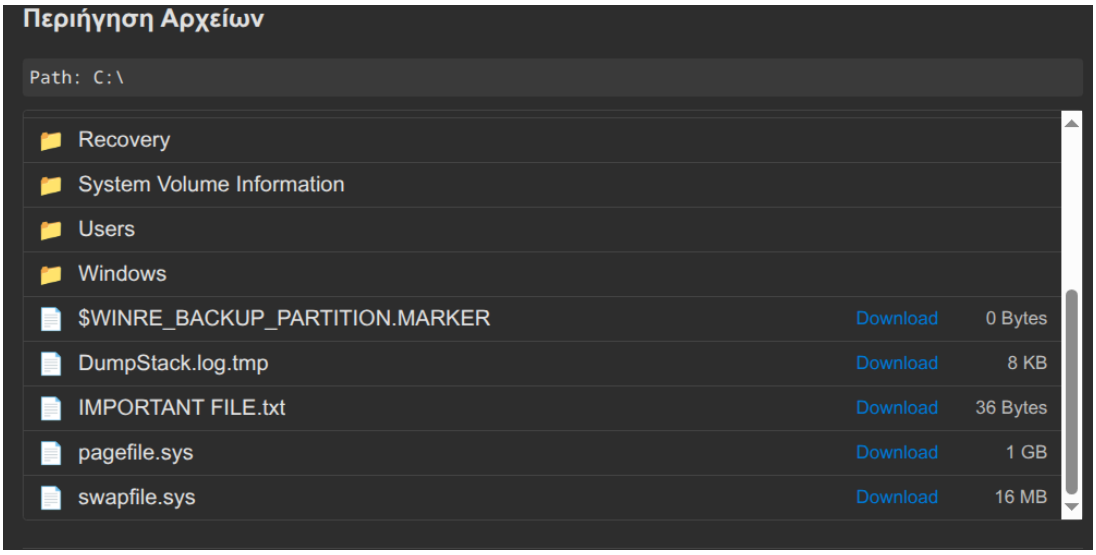
4.2.4 Διαχείριση Αρχείων (Περιήγηση, Λήψη, Μεταφόρτωση)

Στόχος: Να επιδειχθεί η αλληλεπίδραση με το σύστημα αρχείων του agent, συμπεριλαμβανομένης της περιήγησης, της λήψης αρχείων στον server και της μεταφόρτωσης αρχείων στον agent.

Βήματα Εκτέλεσης:

1. Περιήγηση στο Σύστημα Αρχείων (File Browser):

1. Ο χειριστής στέλνει την εντολή `browse "C:\"` (ή άλλη διαδρομή) μέσω του UI.
2. Ο agent εκτελεί την `Get-DirectoryListingAsJson` και επιστρέφει τα περιεχόμενα.
3. Το Web UI εμφανίζει τα αρχεία και τους φακέλους, επιτρέποντας την πλοήγηση (Σχήμα 4.15).



Σχήμα 4.15: Η λειτουργία περιήγησης αρχείων, εμφανίζοντας τα περιεχόμενα του φακέλου 'C:\' στο παραβιασμένο σύστημα.

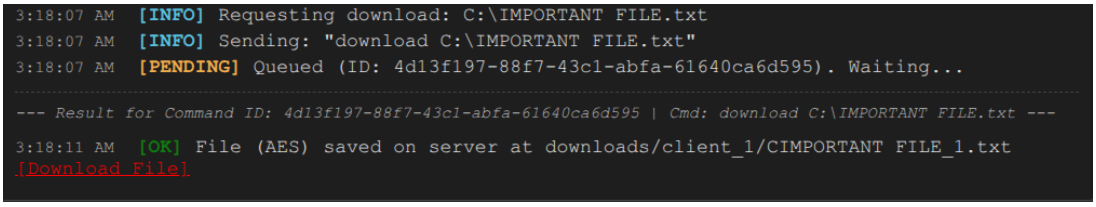
2. Λήψη Αρχείου (Download):

1. Από τον File Browser, ο χειριστής επιλέγει ένα αρχείο για λήψη, π.χ. `C:\IMPORTANT FILE.txt`.

2. Η εντολή `download "C:\IMPORTANT FILE.txt"` αποστέλλεται.

3. Ο agent εκτελεί την `Get-FileAsBase64` και επιστρέφει τα δεδομένα Base64.

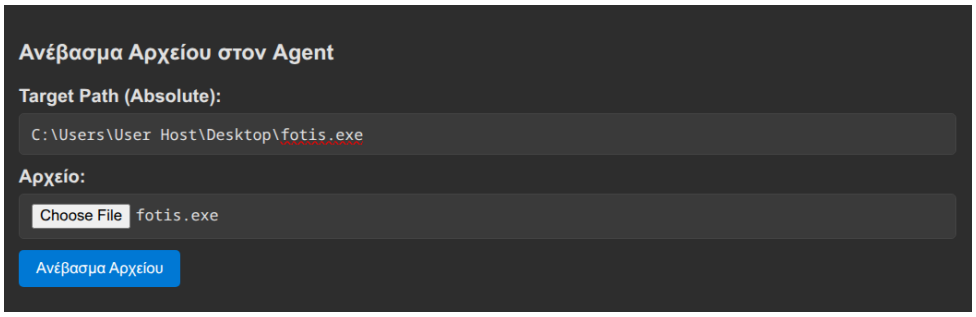
4. Ο server αποθηκεύει το αρχείο στον φάκελο `downloads` (Σχήμα 4.16) και το UI παρέχει σύνδεσμο λήψης.



Σχήμα 4.16: Το αρχείο *IMPORTANT FILE.txt* όπως αποθηκεύτηκε στον φάκελο *downloads* του C2 server μετά την επιτυχή λήψη από τον agent. Δίνεται επιπλέον σύνδεσμος για χειροκίνητη αποθήκευση.

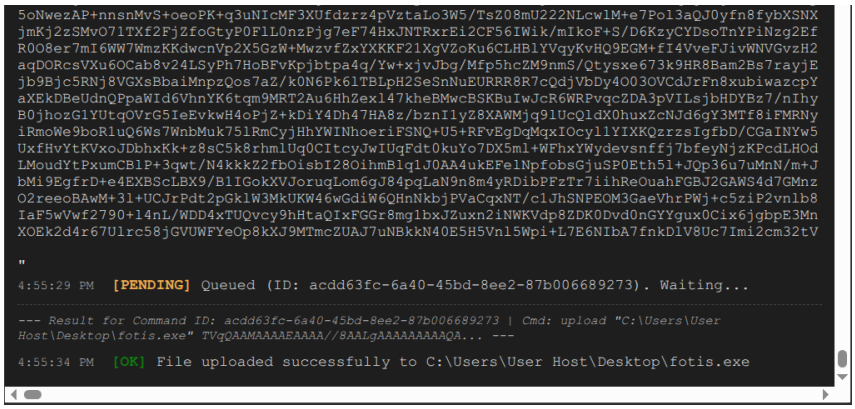
3. Μεταφόρτωση Αρχείου (Upload):

1. Ο χειριστής χρησιμοποιεί τη φόρμα «Upload File», επιλέγει ένα τοπικό αρχείο (`fotis.exe`) και ορίζει διαδρομή προορισμού στον agent (παράδειγμα: `C:\Users\User Host\Desktop\fotis.exe`) (Σχήμα 4.17).

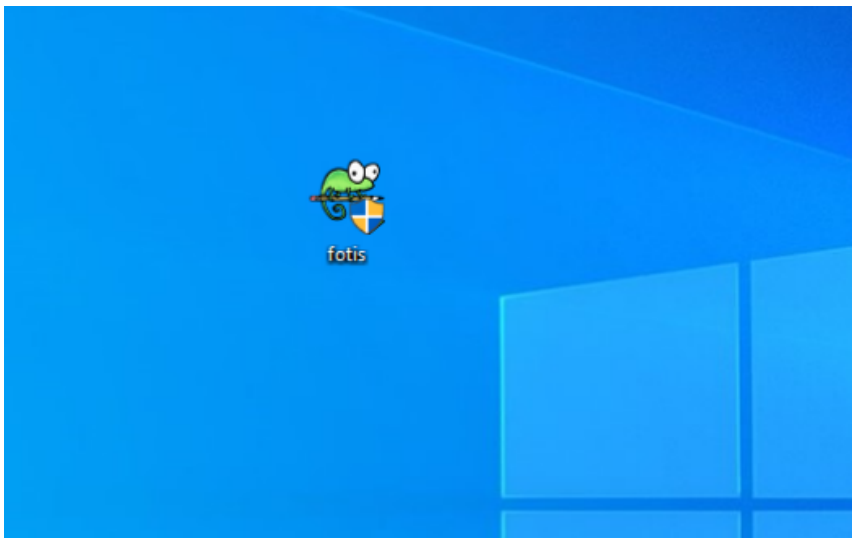


Σχήμα 4.17: Η φόρμα μεταφόρτωσης αρχείου.

2. Το UI στέλνει τα δεδομένα Base64 του αρχείου και τις πληροφορίες στον server, ο οποίος κατασκευάζει και στέλνει την εντολή `upload` στον agent.
3. Ο agent εκτελεί την `Save-FileFromBase64`, αποθηκεύοντας το αρχείο.
4. Ο agent επιστρέφει επιβεβαίωση, η οποία εμφανίζεται στο UI (Σχήμα 4.18). Ο χειριστής μπορεί να επαληθεύσει την ύπαρξη του αρχείου στην VM (Σχήμα 4.19).



Σχήμα 4.18: Επιβεβαίωση επιτυχούς μεταφόρτωσης αρχείου στον agent.



Σχήμα 4.19: Επιβεβαίωση επιτυχούς μεταφόρτωσης αρχείου στην επιφάνεια εργασίας του στοχευμένου συστήματος.

Αυτό το σενάριο επιδεικνύει τις εξής κρίσιμες δυνατότητες: περιήγηση (**T1083 File and Directory Discovery**), λήψη αρχείων (**T1005 Data from Local System / T1041 Exfiltration Over C2 Channel**), και μεταφόρτωση αρχείων (**T1105 Ingress Tool Transfer**). Αυτές οι λειτουργίες είναι θεμελιώδεις για την πλειονότητα των επιθέσεων μετά-εκμετάλλευσης.

4.2.5 Επίδειξη Κρυπτογραφημένης Επικοινωνίας

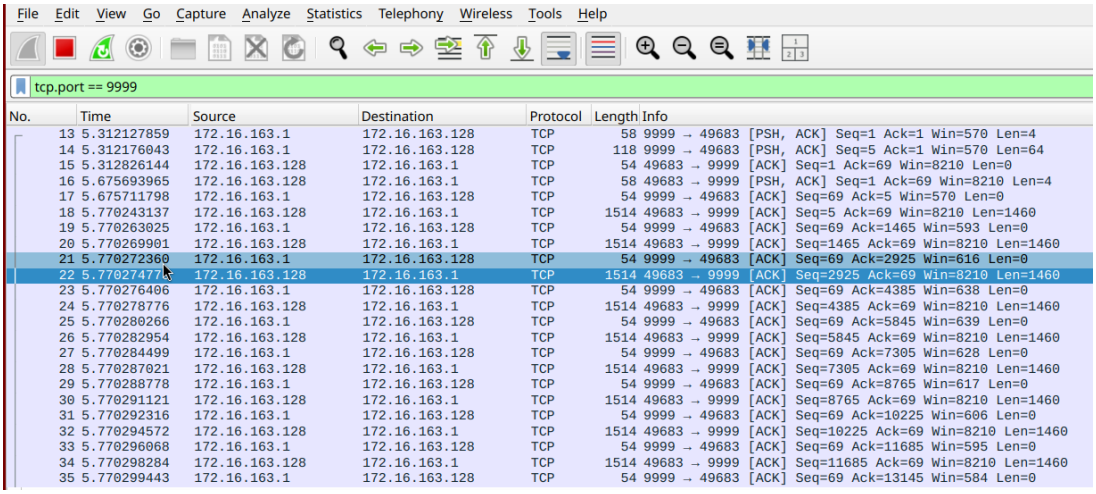
Στόχος: Να επιδειχθεί οπτικά ότι η επικοινωνία μεταξύ του C2 server και του agent είναι όντως κρυπτογραφημένη, καθιστώντας τα περιεχόμενα μη αναγνώσιμα από παθητικούς παρατηρητές δικτύου.

Βήματα Εκτέλεσης:

1. **Έναρξη Καταγραφής Κίνησης:** Στο host machine όπου εκτελείται ο C2 server ή σε ένα ενδιάμεσο σημείο που μπορεί να δει την κίνηση προς/ από την VM του agent, ξεκινά η καταγραφή της δικτυακής κίνησης χρησιμοποιώντας ένα εργαλείο όπως το Wireshark. Η καταγραφή φιλτράρεται για να δείχνει μόνο την κίνηση προς/από την IP της VM και την TCP θύρα του C2 server (9999), όπως φαίνεται και στο Σχήμα 4.20.

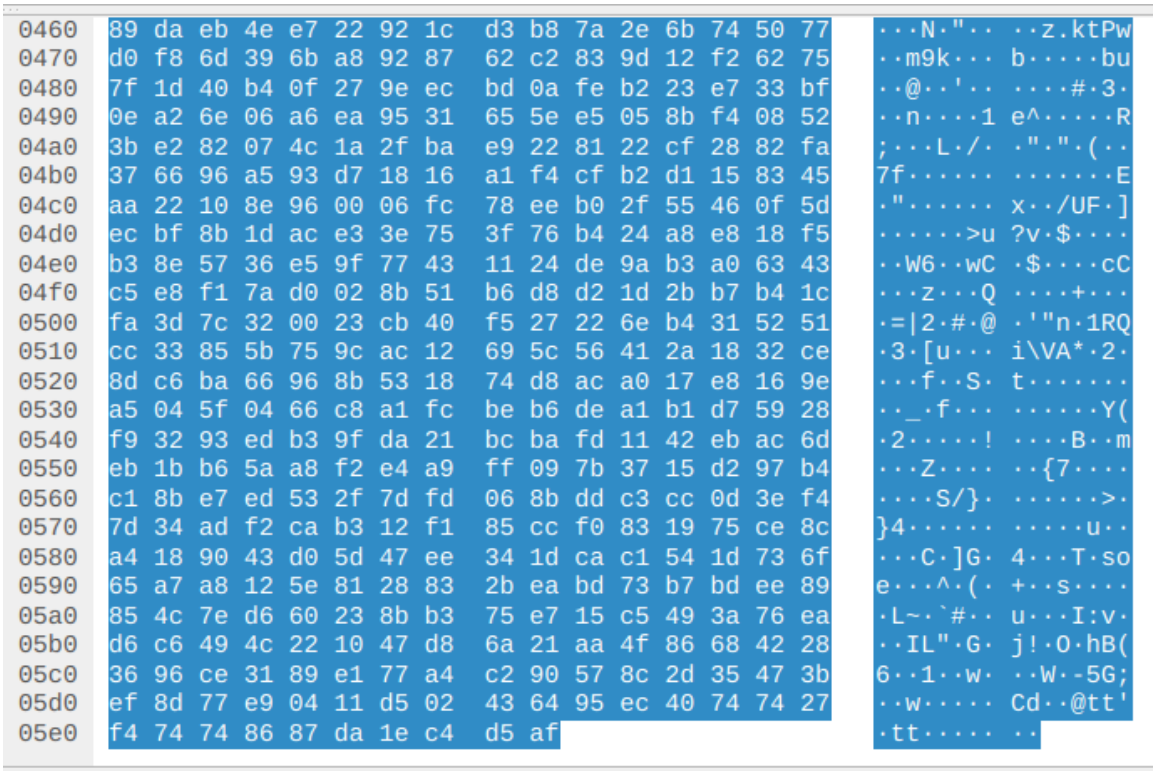
2. **Εκτέλεση Εντολών:** Από το Web UI, εκτελούνται μερικές απλές εντολές (`whoami`, `dir`) προς τον agent.

3. **Παρατήρηση Καταγεγραμμένης Κίνησης:** Μετά την εκτέλεση των εντολών και τη λήψη των απαντήσεων, η καταγραφή στο Wireshark σταματά. Εξετάζονται τα πακέτα TCP που ανταλλάχθηκαν.



Σχήμα 4.20: Το πρόγραμμα καταγραφής δικτυακής κίνησης Wireshark με εφαρμοσμένο το φίλτρο *tcp.port == 9999*.

4. **Ανάλυση Περιεχομένου Πακέτων:** Στο Wireshark, επιλέγοντας πακέτα που περιέχουν τα δεδομένα εφαρμογής (μετά το TCP handshake), το τμήμα «Data» ή «Payload» θα πρέπει να εμφανίζει ακατανόητα bytes, αντί για το καθαρό κείμενο των εντολών ή των αποτελεσμάτων. Αυτό οφείλεται στην κρυπτογράφηση AES που εφαρμόζεται από τον server και τον agent πριν την αποστολή. (Σχήμα 4.21).



Σχήμα 4.21: Το ωφέλιμο φορτίο των πακέτων περιέχει κρυπτογραφημένα δεδομένα και δεν είναι αναγνώσιμο ως απλό κείμενο (*πεδίο «Data»*).

Η ανάλυση της δικτυακής κίνησης με το Wireshark επιβεβαιώνει ότι το πρωτόκολλο επικοινωνίας που υλοποιήθηκε επιτυγχάνει τον στόχο της εμπιστευτικότητας των δεδομένων. Αν και η ίδια η TCP σύνδεση είναι ορατή, το περιεχόμενο των εντολών και των απαντήσεων που ανταλλάσσονται είναι προστατευμένο μέσω της κρυπτογράφησης AES, καθιστώντας αδύνατη την άμεση κατανόησή του από οποιονδήποτε υποκλέψει την επικοινωνία. Αυτό είναι ένα θεμελιώδες χαρακτηριστικό για οποιοδήποτε C2 framework που επιδιώκει να λειτουργεί με κάποιο βαθμό μυστικότητας.

4.2.6 Σύνοψη Τεχνικών

Τα σενάρια που παρουσιάστηκαν στις προηγούμενες υποενότητες είχαν ως στόχο την πρακτική επίδειξη των βασικών δυνατοτήτων του εκπαιδευτικού C2 framework. Μέσα από αυτά τα σενάρια, αναδείχθηκε πώς το πλαίσιο διευκολύνει την εκτέλεση διαφόρων ενεργειών μετά-εκμετάλλευσης, οι οποίες αντιστοιχούν σε συγκεκριμένες τακτικές και τεχνικές του πλαισίου MITRE ATT&CK.

Ο παρακάτω πίνακας (Πίνακας 4.1) συνοψίζει τις κύριες ενέργειες που επιδείχθηκαν και την αντιστοίχισή τους στις σχετικές τακτικές και τεχνικές/υπο-τεχνικές του ATT&CK.

Πίνακας 4.1: Αντιστοίχιση Επιδειχθεισών Ενεργειών C2 με Τακτικές/Τεχνικές MITRE ATT&CK

Ενέργεια / Εντολή (Παράδειγμα)	**ATT&CK Τακτική (ID)**	**ATT&CK Τεχνική / Υπο-τεχνική (ID)**
Σύνδεση Agent και Επικοινωνία	Command and Control (TA0011)	T1071 (Application Layer Protocol) T1573.001 (Symmetric Cryptography - AES) (T1571: Non-Standard Port)
`whoami`	Discovery (TA0007)	T1033 (System Owner/User Discovery)
`$env:COMPUTERNAME`	Discovery (TA0007)	T1082 (System Information Discovery)
`ipconfig`	Discovery (TA0007)	T1016 (System Network Config. Discovery)
`Get-CimInstance Win32_OperatingSystem`	Discovery (TA0007)	T1082 (System Information Discovery)
`Get-Process` / `ps`	Discovery (TA0007)	T1057 (Process Discovery)
Εκτέλεση Αυθαίρετης Εντολής (`New-Item`, κ.ο.κ.)	Execution (TA0002)	T1059 (Cmd and Scripting Interpreter) - T1059.001 (PowerShell)
`Remove-Item` (Διαγραφή Αρχείου)	Defense Evasion (TA0005)	T1070 (Indicator Removal on Host) - T1070.004 (File Deletion)
Λήψη Screenshot	Collection (TA0009)	T1113 (Screen Capture)
Περιήγηση Αρχείων	Discovery (TA0007)	T1083 (File and Directory Discovery)
Λήψη Αρχείου	Collection (TA0009) / Exfiltration (TA0010)	T1005 (Data from Local System) / T1041 (Exfiltration Over C2 Channel)
Μεταφόρτωση Αρχείου	Command and Control (TA0011)	T1105 (Ingress Tool Transfer)
Αποστολή Αποτελεσμάτων/Δεδομένων	Exfiltration (TA0010)	T1041 (Exfiltration Over C2 Channel)

Όπως είναι φανερό στον πίνακα 4.1, το εκπαιδευτικό C2 framework που αναπτύχθηκε, παρά την εστίασή του στην απλότητα, επιτρέπει την υλοποίηση ενός σημαντικού εύρους τεχνικών που καλύπτουν κρίσιμες τακτικές της επιθετικής αλυσίδας.

4.2.7 Ανακεφαλαίωση Κεφαλαίου 4

Το παρόν κεφάλαιο επικεντρώθηκε στην πρακτική επίδειξη του εκπαιδευτικού C2 framework μέσω μιας μελέτης περίπτωσης. Αρχικά, περιγράφηκε λεπτομερώς το απομονωμένο εργαστηριακό περιβάλλον. Στη συνέχεια, παρουσιάστηκαν διαδοχικά σενάρια που κάλυψαν την αρχική σύνδεση του agent και τη συλλογή πληροφοριών, την απομακρυσμένη εκτέλεση αυθαίρετων εντολών, τη λήψη στιγμιότυπων οθόνης, τη διαχείριση αρχείων (περιήγηση, λήψη, μεταφόρτωση) και την επιβεβαίωση της κρυπτογραφημένης επικοινωνίας. Κάθε σενάριο αναλύθηκε ως προς τα βήματα εκτέλεσης, τα αποτελέσματα και τη σημασία του, με αντιστοίχιση στις τακτικές και τεχνικές του MITRE ATT&CK. Η μελέτη περίπτωσης κατέδειξε έμπρακτα τις βασικές και ορισμένες προχωρημένες δυνατότητες του πλαισίου. Με την ολοκλήρωση της επίδειξης, το επόμενο κεφάλαιο θα προχωρήσει στη συζήτηση των ευρημάτων, την κριτική ανάλυση των

περιορισμών και την εξέταση πιθανών μελλοντικών κατευθύνσεων.

Κεφάλαιο 5

Συζήτηση και Αξιολόγηση

Μετά την παρουσίαση του θεωρητικού υποβάθρου (Κεφάλαιο 2), του σχεδιασμού και της μεθοδολογίας (Κεφάλαιο 3), και την πρακτική επίδειξη μέσω της μελέτης περίπτωσης (Κεφάλαιο 4), το παρόν κεφάλαιο προχωρά σε μια κριτική συζήτηση και ανάλυση της εργασίας. Στόχος είναι να αξιολογηθεί η λειτουργικότητα του εκπαιδευτικού C2 framework που αναπτύχθηκε, να ερμηνευθούν τα ευρήματα της μελέτης περίπτωσης στο ευρύτερο πλαίσιο της κυβερνοασφάλειας, να αναγνωριστούν με ειλικρίνεια οι περιορισμοί της υλοποίησης, να θιχτούν τα ηθικά ζητήματα που προκύπτουν, και τέλος, να προταθούν πιθανές κατευθύνσεις για μελλοντική έρευνα και ανάπτυξη.

5.1 Ανάλυση Λειτουργικότητας του Πλαισίου

Το εκπαιδευτικό C2 framework που αναπτύχθηκε πέτυχε τους κύριους σχεδιαστικούς του στόχους, όπως αυτοί τέθηκαν στην Εισαγωγή και στην Ενότητα 3.1. Δημιουργήθηκε ένα λειτουργικό σύστημα με κεντροποιημένη αρχιτεκτονική Server-Agent, ικανό να διαχειρίζεται την επικοινωνία με έναν Powershell agent σε περιβάλλον Windows. Η θεμελιώδης απαίτηση για κρυπτογραφημένη επικοινωνία ικανοποιήθηκε μέσω της υλοποίησης κρυπτογράφησης AES σε επίπεδο εφαρμογής, χρησιμοποιώντας ένα προκαθορισμένο κοινό κλειδί και IV, το οποίο διασφάλισε την εμπιστευτικότητα των ανταλλασσόμενων δεδομένων (εντολές, αποτελέσματα, δεδομένα αρχείων και screenshots) στο κανάλι επικοινωνίας TCP.

Το πλαίσιο επέδειξε την ικανότητα να υλοποιεί τις ακόλουθες βασικές και προχωρημένες λειτουργίες C2:

- Οι agents μπορούν να συνδεθούν στον server και να καταγραφούν, επιτρέποντας την αρχική συλλογή πληροφοριών συστήματος.
- Ο χειριστής, μέσω της web διεπαφής, μπορεί να δει τους ενεργούς agents και τις βασικές τους πληροφορίες.
- Είναι δυνατή η ανάθεση εργασιών με την αποστολή αυθαίρετων εντολών Powershell σε συγκεκριμένους agents.

- Τα αποτελέσματα της εκτέλεσης των εντολών επιστρέφονται επιτυχώς στον server και παρουσιάζονται στον χειριστή.
- Υποστηρίζεται η λήψη στιγμιότυπων οθόνης από το σύστημα του agent.
- Παρέχεται λειτουργικότητα περιήγησης στο σύστημα αρχείων του agent.
- Υλοποιήθηκε η δυνατότητα μεταφόρτωσης αρχείων από τον χειριστή στον agent και λήψης αρχείων από τον agent στον server.

Οι υλοποιητικές προκλήσεις που αντιμετωπίστηκαν, όπως η αρχική προσπάθεια ενσωμάτωσης του TLS και οι επακόλουθες δυσκολίες, η διαχείριση του parsing των εντολών και των ορισμάτων στον Powershell agent, ειδικά για τις διαδρομές με κενά, και η λεπτομερής διαμόρφωση του πρωτοκόλλου επικοινωνίας με πρόθεμα μήκους για τα κρυπτογραφημένα δεδομένα, αναδεικνύουν τις πρακτικές δυσκολίες στην ανάπτυξη ακόμη και βασικών δικτυακών και κρυπτογραφικών συστημάτων. Αυτές οι προκλήσεις, ωστόσο, ξεπεράστηκαν, οδηγώντας σε ένα σταθερό και λειτουργικό πρωτότυπο που εξυπηρετεί τους εκπαιδευτικούς σκοπούς της εργασίας.

5.2 Ερμηνεία Αποτελεσμάτων Μελέτης Περίπτωσης

Η μελέτη περίπτωσης στο Κεφάλαιο 4 έδειξε έμπρακτα την αξία και τους κινδύνους που συνδέονται με τις δυνατότητες ενός C2 framework, ακόμη και ενός απλού όπως το παρόν. Η ευκολία με την οποία ο χειριστής μπόρεσε να αποκτήσει επίγνωση της κατάστασης του συστήματος-στόχου κάνει ολοφάνερο το πόσο γρήγορα μπορεί ένας επιτιθέμενος να «χαρτογραφήσει» ένα παραβιασμένο περιβάλλον.

Η επιτυχής απομακρυσμένη εκτέλεση αυθαίρετων εντολών Powershell (π.χ. δημιουργία ή διαγραφή αρχείων, εκκίνηση διεργασιών) απέδειξε τη θεμελιώδη ισχύ που παρέχει το C2: τη μετατροπή ενός παραβιασμένου συστήματος σε τηλεχειριζόμενο εργαλείο. Η επίδειξη της λήψης screenshot, της περιήγησης στο σύστημα αρχείων, της λήψης και μεταφόρτωσης αρχείων τόνισε περαιτέρω τους κινδύνους για την ιδιωτικότητα, την εμπιστευτικότητα των δεδομένων και τη συνολική ασφάλεια του συστήματος.

Η αντιστοίχιση των επιδειχθεισών ενεργειών με τις τακτικές και τεχνικές του MITRE ATT&CK (Execution, Discovery, Collection, Command and Control, Exfiltration, Defense Evasion, Ingress Tool Transfer) έδειξε ότι οι υλοποιημένες λειτουργίες C2 έχουν άμεση εφαρμογή και συνάφεια με τις μεθόδους που χρησιμοποιούνται σε πραγματικές επιθέσεις. Αυτό ενισχύει τον σκοπό της ευαισθητοποίησης, καθιστώντας σαφές ότι ο έλεγχος ενός συστήματος μέσω C2 ανοίγει ένα ευρύ φάσμα επιθετικών δυνατοτήτων. Η επίδειξη της κρυπτογραφημένης επικοινωνίας μέσω Wireshark επιβεβαίωσε ότι, αν και απλή, η κρυπτογράφηση AES παρέχει ένα βασικό επίπεδο προστασίας των μεταδιδόμενων δεδομένων.

5.3 Περιορισμοί και Ζητήματα Ασφάλειας

Είναι ζωτικής σημασίας να αναγνωριστούν οι σημαντικοί περιορισμοί του εκπαιδευτικού C2 framework που αναπτύχθηκε, οι οποίοι το καθιστούν ακατάλληλο για οποιαδήποτε χρήση πέραν της ελεγχόμενης

εκπαιδευτικής επίδειξης:

- **Έλλειψη Προηγμένης Αποφυγής Ανίχνευσης (Evasion):** Δεν περιλαμβάνει καμία εξελιγμένη τεχνική για την αποφυγή ανίχνευσης από σύγχρονα συστήματα AV/EDR. Η χρήση `Invoke-Expression`, η απλή δομή του agent, και η δικτυακή συμπεριφορά, ακόμα και κρυπτογραφημένη με AES σε μη τυπική θύρα, χωρίς μίμηση γνωστών πρωτοκόλλων, το καθιστούν πιθανό στόχο για ανίχνευση βάσει συμπεριφοράς ή ανάλυσης δικτύου.

- **Προκαθορισμένο Κλειδί AES και IV:** Η χρήση ενός σταθερού, προκαθορισμένου κλειδιού AES και IV για την κρυπτογράφηση είναι ο σημαντικότερος περιορισμός ασφαλείας. Αν το κλειδί αυτό αποκαλυφθεί (πιθανώς μέσω reverse engineering του agent ή του server), όλη η προηγούμενη και μελλοντική επικοινωνία μπορεί να αποκρυπτογραφηθεί. Δεν παρέχεται αυθεντικοποίηση του server στον agent (ή το αντίστροφο), ούτε Perfect Forward Secrecy. Οποιοσδήποτε με το κλειδί μπορεί να υποδυθεί τον server ή τον agent.

- **Έλλειψη Εδραίωσης Παρουσίας (Persistence):** Ο agent δεν επιβιώνει μετά από επανεκκίνηση του συστήματος-στόχου και πρέπει να εκτελείται χειροκίνητα.

- **Πρωτόκολλο Επικοινωνίας:** Αν και το υιοθετηθέν πρωτόκολλο με πρόθεμα μήκους για τα κρυπτογραφημένα δεδομένα και η δομή CID/Type/Data για τα αποκρυπτογραφημένα μηνύματα είναι λειτουργικό, παρουσιάζει περιορισμούς:

  - *Στερείται Απόκρυψης:* Δεν μιμείται γνωστά πρωτόκολλα όπως HTTP/S ή DNS, καθιστώντας την κίνηση, αν και κρυπτογραφημένη, ύποπτη όταν χρησιμοποιείται σε μη τυπικές θύρες.
  - *Δυσκαμψία:* Δεν διαθέτει μηχανισμούς παραμετροποίησης της κίνησης όπως τα Malleable C2 Profiles.
  - *Μεταφορά Δεδομένων:* Η ενσωμάτωση δυαδικών δεδομένων (screenshots, uploads, downloads) μέσω κωδικοποίησης Base64 εντός του κύριου καναλιού C2 αυξάνει τον όγκο των δεδομένων και μπορεί να είναι αναποτελεσματική ή να προκαλέσει προβλήματα με πολύ μεγάλα αρχεία, υπερβαίνοντας πιθανώς τα όρια μεγέθους μηνυμάτων ή προκαλώντας καθυστερήσεις.

- **Διαχείριση Κατάστασης στη Μνήμη:** Η αποθήκευση όλων των πληροφοριών στη μνήμη του server σημαίνει ότι τα δεδομένα χάνονται με τον τερματισμό του server.

- **Βασικός Χειρισμός Σφαλμάτων:** Αν και υπάρχει χειρισμός σφαλμάτων, το σύστημα ενδέχεται να μην είναι πλήρως ανθεκτικό σε όλα τα πιθανά σενάρια αποτυχίας.

- **Έμφαση στο Powershell:** Η εξάρτηση από το Powershell περιορίζει τον agent σε περιβάλλοντα Windows.

Αυτοί οι περιορισμοί αναδεικνύουν την τεράστια διαφορά μεταξύ ενός εκπαιδευτικού πρωτοτύπου και ενός ώριμου, επιχειρησιακού C2 framework.

5.4 Ηθικά Ζητήματα

Η ανάπτυξη και η μελέτη εργαλείων που σχετίζονται με την επιθετική κυβερνοασφάλεια, όπως τα C2 frameworks, εγείρει αναπόφευκτα σημαντικά ηθικά ζητήματα. Ενώ η κατανόηση αυτών των εργαλείων είναι απαραίτητη για την ανάπτυξη αποτελεσματικών αμυντικών στρατηγικών («know your enemy»), η ίδια η γνώση και τα εργαλεία μπορούν να χρησιμοποιηθούν για κακόβουλους σκοπούς.

Είναι κρίσιμο να τονιστεί ότι το πλαίσιο που αναπτύχθηκε στην παρούσα εργασία δημιουργήθηκε αποκλειστικά για εκπαιδευτικούς και ερευνητικούς σκοπούς εντός ενός ελεγχόμενου ακαδημαϊκού περιβάλλοντος. Η οποιαδήποτε χρήση του για μη εξουσιοδοτημένη πρόσβαση ή έλεγχο συστημάτων είναι παράνομη και ανήθικη. Η δημοσίευση ή η κοινοποίηση του κώδικα πρέπει να γίνεται με υπευθυνότητα, λαμβάνοντας υπόψη τον δυνητικό κίνδυνο κατάχρησης. Η ισορροπία μεταξύ της ανάγκης για ανοιχτή έρευνα και εκπαίδευση στην κυβερνοασφάλεια και του κινδύνου παροχής εργαλείων σε κακόβουλους παράγοντες αποτελεί μια διαρκή πρόκληση για την ακαδημαϊκή και ερευνητική κοινότητα.

5.5 Μελλοντική Εργασία και Πιθανές Βελτιώσεις

Το εκπαιδευτικό C2 framework, ως βασικό πρωτότυπο, προσφέρει πολλές δυνατότητες για μελλοντική επέκταση και βελτίωση. Ορισμένες πιθανές κατευθύνσεις περιλαμβάνουν:

- **Βελτιωμένη Απόκρυψη/Evasion:** Ενσωμάτωση τεχνικών Malleable C2 για την παραμετροποίηση του πρωτοκόλλου εφαρμογής, χρήση DNS ή HTTP/S ως εναλλακτικών καναλιών, τεχνικές memory obfuscation στον agent, AMSI bypass, απόκρυψη της κονσόλας Powershell και έξυπνη ονομασία της διεργασίας για να μην είναι εύκολη η ανακάλυψη της.
- **Μηχανισμοί Persistence:** Υλοποίηση διαφόρων τεχνικών εδραίωσης παρουσίας στον agent έτσι ώστε να παραμένει ενεργός στο σύστημα πάση θυσία.
- **Προηγμένες Λειτουργίες Agent:** Προσθήκη δυνατοτήτων όπως η καταγραφή πληκτρολόγησης (keylogging), η συλλογή διαπιστευτηρίων (μέσω Mimikatz ή παρόμοιων τεχνικών), και βασικές τεχνικές πλευρικής κίνησης (lateral movement).
- **Βελτιωμένη Ασφάλεια Επικοινωνίας:** Εξέταση μεθόδων για δυναμική ανταλλαγή κλειδιών AES αντί για προκαθορισμένο, ή υιοθέτηση ενός υβριδικού συστήματος όπως για παράδειγμα: ασύμμετρη κρυπτογραφία για την αρχική ανταλλαγή ενός συμμετρικού κλειδιού συνόδου. Εναλλακτικά, επανεξέταση της υλοποίησης TLS ή χρήση Websockets. Ενσωμάτωση nonces ή timestamps για προστασία από replay attacks με την τρέχουσα υλοποίηση AES.
- **Βελτιωμένος Server Backend:** Χρήση βάσης δεδομένων για την αποθήκευση της κατάστασης.
- **Ασύγχρονος Server/Agent:** Μετάβαση σε ασύγχρονο προγραμματισμό για καλύτερη διαχείριση συνδέσεων.
- **Εναλλακτικοί Agents:** Ανάπτυξη agents σε άλλες γλώσσες ή για άλλα λειτουργικά συστήματα.

- **Βελτιωμένη Διεπαφή Χρήστη:** Ανάπτυξη ενός πιο πλούσιου Web UI.
- **Βελτιστοποίηση Απόδοσης:** Μείωση των καθυστερήσεων μέσω τεχνικών όπως το beaconing από τον agent ή η βελτιστοποίηση των timeouts.

Αυτές οι πιθανές επεκτάσεις αναδεικνύουν την πολυπλοκότητα και το εύρος των δυνατοτήτων που μπορούν να ενσωματωθούν σε ένα C2 framework.

Κεφάλαιο 6

Συμπεράσματα

6.1 Σύνοψη Σκοπού και Αποτελεσμάτων

Η παρούσα πτυχιακή εργασία ανέλαβε το εγχείρημα της διερεύνησης των συστημάτων διοίκησης και ελέγχου (C2), τα οποία αποτελούν κεντρικό μηχανισμό για τους επιτιθέμενους μετά την αρχική παραβίαση ενός συστήματος. Στόχος ήταν η θεωρητική κατανόηση αυτών των υποδομών – από την ιστορία και τις αρχιτεκτονικές τους έως τις τεχνικές που χρησιμοποιούν οι σύγχρονοι δράστες – και η πρακτική αποτύπωση αυτής της γνώσης μέσω της ανάπτυξης ενός εκπαιδευτικού C2 πλαισίου.

Το πλαίσιο που δημιουργήθηκε, αποτελούμενο από έναν κεντρικό εξυπηρετητή και έναν πράκτορα για συστήματα Windows, σχεδιάστηκε με γνώμονα την απλότητα και την εκπαιδευτική του αξία. Η επικοινωνία μεταξύ των δύο μερών προστατεύτηκε με συμμετρική κρυπτογράφηση (AES), επιδεικνύοντας τη θεμελιώδη ανάγκη για εμπιστευτικότητα σε τέτοια κανάλια. Μέσω μιας μελέτης περίπτωσης, το σύστημα απέδειξε την ικανότητά του να εκτελεί βασικές ενέργειες που συναντώνται σε πραγματικές επιθέσεις, όπως η συλλογή πληροφοριών από το παραβιασμένο σύστημα, η απομακρυσμένη εκτέλεση εντολών, η διαχείριση αρχείων και η λήψη στιγμιότυπων οθόνης.

Τα κύρια ευρήματα της εργασίας επιβεβαιώνουν ότι ακόμη και απλές υλοποιήσεις C2 μπορούν να παρέχουν σημαντικό έλεγχο επί ενός συστήματος, υπογραμμίζοντας την κρισιμότητα αυτών των υποδομών για την επιτυχία εξελιγμένων κυβερνοεπιθέσεων. Παράλληλα, αναδείχθηκε η ευελιξία εργαλείων όπως το Powershell, αλλά και οι πρακτικές προκλήσεις που ενέχει η ανάπτυξη ασφαλών και αξιόπιστων δικτυακών εφαρμογών, ακόμη και σε εκπαιδευτικό επίπεδο.

6.2 Συνεισφορά, Περιορισμοί και Προοπτικές

Η κύρια συνεισφορά αυτής της πτυχιακής έγκειται στην παροχή ενός λειτουργικού παραδείγματος που αποσκοπεί στην αποσαφήνιση των μηχανισμών διοίκησης και ελέγχου. Λειτουργεί ως εκπαιδευτικό εργαλείο, βοηθώντας στην κατανόηση των κινδύνων που απορρέουν από την παραβίαση ενός συστήματος και τον απομακρυσμένο έλεγχό του, συνδέοντας την πρακτική επίδειξη με αναγνωρισμένα πλαίσια ανά-

λυσης απειλών όπως το MITRE ATT&CK. Η διαδικασία σχεδιασμού και υλοποίησης, μαζί με τις προκλήσεις που αντιμετωπίστηκαν, προσφέρει επίσης μια ρεαλιστική εικόνα της πολυπλοκότητας αυτών των συστημάτων.

Είναι, ωστόσο, απαραίτητο να αναγνωριστούν οι περιορισμοί του αναπτυχθέντος πλαισίου. Καθώς σχεδιάστηκε για εκπαιδευτικούς σκοπούς, στερείται προηγμένων μηχανισμών απόκρυψης από σύγχρονα συστήματα ασφαλείας και η υλοποιηθείσα κρυπτογράφηση δεν θα ήταν επαρκής για πραγματικές επιχειρησιακές συνθήκες. Οι δυνατότητες εδραίωσης παρουσίας και οι πιο εξελιγμένες τεχνικές επίθεσης δεν καλύφθηκαν. Αυτοί οι περιορισμοί καθιστούν το πλαίσιο κατάλληλο μόνο για χρήση σε αυστηρά ελεγχόμενα και απομονωμένα περιβάλλοντα.

Η ενασχόληση με τέτοια εργαλεία εγείρει σημαντικά ηθικά ζητήματα, και τονίζεται ότι η γνώση αυτή πρέπει να αξιοποιείται αποκλειστικά για την ενίσχυση της άμυνας και την εκπαίδευση. Για μελλοντική εργασία, υπάρχουν πολλές κατευθύνσεις, όπως η ενίσχυση της ασφάλειας της επικοινωνίας με πιο ισχυρά πρωτόκολλα, η προσθήκη τεχνικών απόκρυψης, και η επέκταση των λειτουργιών του πράκτορα.

Συνοψίζοντας, η κατανόηση της λειτουργίας των επιθετικών εργαλείων και υποδομών, όπως τα συστήματα C2, είναι θεμελιώδης για την ανάπτυξη αποτελεσματικών στρατηγικών κυβερνοάμυνας. Η παρούσα εργασία, μέσω της θεωρητικής ανάλυσης και της πρακτικής υλοποίησης, φιλοδοξεί να συμβάλει στην προσπάθεια αυτή, ενισχύοντας την γνώση και την ευαισθητοποίηση στον κρίσιμο τομέα της ασφάλειας πληροφοριακών συστημάτων.

6.3 Τελικές Σκέψεις του Συγγραφέα για την Εργασία

Πέραν των τεχνικών προκλήσεων που προέκυψαν κατά τη διάρκεια της ενασχόλησης μου με την εργασία, ένα από τα πιο δύσκολα κομμάτια ήταν η μετάφραση και η απόδοση της τεχνικής ορολογίας από την αγγλική γλώσσα στην ελληνική. Φυσικά, δεδομένου ότι οι περισσότεροι όροι της πληροφορικής και ειδικότερα, της κυβερνοασφάλειας δημιουργήθηκαν στην αγγλική γλώσσα, πολλές φορές έπρεπε να γραφτούν ως έχει. Για παράδειγμα ο όρος Command and Control μπορεί να μεταφραστεί ως *Διοίκηση και Έλεγχος* ή *Εντολή και Έλεγχος*. Επέλεξα τον πρώτο όρο καθώς τον θεώρησα καταλληλότερο, ωστόσο το νόημα παραμένει το ίδιο και στις δύο περιπτώσεις. Επίσης, σε αυτό το σημείο, θέλω να αναφερθώ και στον τίτλο της εργασίας, ο οποίος αναφέρεται σε «περιβάλλον» C2 και όχι «πλαίσιο» C2. Επέλεξα τον όρο «περιβάλλον» έχοντας κατά νού όχι μόνο το πλαίσιο που θα αναπτυσσόταν, αλλά και το περιβάλλον δοκιμών στο οποίο θα δοκιμαζόταν. Δυστυχώς, για τη διατήρηση του ευρύτερου νοήματος, το κείμενο είναι γεμάτο με παρενθέσεις που ακολουθούν μια ορολογία στα ελληνικά και διευκρινίζουν την αντίστοιχη αγγλική (ή το αντίστροφο), κάνοντας την συνολική εικόνα του κειμένου και την εμπειρία ανάγνωσης πιο περίπλοκη.

Συνολικά, η δημιουργία αυτής της πτυχιακής εργασίας ήταν μια εξαντλητική αλλά αρκετά εποικοδομητική εμπειρία που με βοήθησε να καταλάβω καλύτερα μια πτυχή της κυβερνοασφάλειας, η οποία είναι πιο επίκαιρη από ποτέ. Ταυτόχρονα, η πτυχιακή αυτή με βοήθησε να ενισχύσω περαιτέρω τα θεμέλια των γνώσεων μου επάνω στη δικτύωση και τον προγραμματισμό. Θεωρώ πως προέκυψε μια επαρκής βάση κώδικα για μελλοντική μελέτη, ενασχόληση και βελτίωση, και από εμένα και από τυχόν ενδιαφε-

ρόμενους, καθώς ο κώδικας θα φιλοξενηθεί σύντομα σε github ή/ και gitlab αποθετήριο, στο προσωπικό μου προφίλ (*@shirakoisnull*).

Ευελπιστώ λοιπόν, πως οι γνώσεις που αποκόμισα από την παρούσα εργασία, θα μου φανούν χρήσιμες σε μια πιθανή μελλοντική καριέρα στον τομέα της κυβερνοασφάλειας. Θα ήθελα να ευχαριστήσω τους αναγνώστες για τον χρόνο τους και θέλω να πιστεύω πως μέσω της εργασίας αυτής, ευαισθητοποιήθηκαν έστω και λίγο πάνω στα κρίσιμα ζητήματα της ασφάλειας του χώρου της πληροφορικής.

Βιβλιογραφία

[1] I. Cvitić, D. Perakovic, M. Periša, A. Jevremović και A. Shalaginov, «An Overview of Smart Home IoT Trends and related Cybersecurity Challenges», *Mobile Networks and Applications*, τόμ. 28, σσ. 1–15, Οκτ. 2022. doi: 10.1007/s11036-022-02055-w.

[2] ISC2, *ISC2 2024 Cybersecurity Workforce Study*, https://www.isc2.org/Insights/2024/10/ISC2-2024-Cybersecurity-Workforce-Study, Οκτ. 2024.

[3] «Κυβερνοεπιθέσεις & κυβερνοασφάλεια», Εθνική Αρχή Κυβερνοασφάλειας, επίσκεψη 2 Μάι. 2025. διεύθν.: https://cyber.gov.gr/kyvernoepitheseis/kyvernoepitheseis-kyvernoasfaleia/.

[4] European Union Agency for Cybersecurity (ENISA), *ENISA Threat Landscape 2024*, Heraklion, Greece / Athens, Greece, Σεπτ. 2024. διεύθν.: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024.

[5] Verizon, *2025 Data Breach Investigations Report*, Basking Ridge, NJ, USA, Μάι. 2025. διεύθν.: https://www.verizon.com/business/resources/Tb62/reports/2025-dbir-data-breach-investigations-report.pdf.

[6] OWASP Foundation, *Open Web Application Security Project*, 2024. διεύθν.: https://owasp.org/.

[7] M. S. Souppaya, E. M. Scarfone και D. F. Dodson, *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities*, Gaithersburg, MD, USA, Φεβ. 2022. doi: 10.6028/NIST.SP.800-218. διεύθν.: https://csrc.nist.gov/publications/detail/sp/800-218/final.

[8] Synopsys Cybersecurity Research Center (CyRC), *2025 OSSRA: Open Source Security and Risk Analysis*, Sunnyvale, CA, USA, Φεβ. 2025. διεύθν.: https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html.

[9] Lockheed Martin, *Cyber Kill Chain® | Lockheed Martin*. επίσκεψη 21 Μαρ. 2025. διεύθν.: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[10] The MITRE Corporation. «MITRE ATT&CK®», επίσκεψη 21 Μαρ. 2025. διεύθν.: https://attack.mitre.org/.

[11] M. S. Vassiliou, D. S. Alberts και J. R. Agre, *C2 Re-envisioned: The Future of the Enterprise*. CRC Press, Δεκ. 2014.

[12] E. Hutchins, M. Cloppert και R. Amin, «Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains», *Leading Issues in Information Warfare & Security Research*, τόμ. 1, Ιαν. 2011.

[13] J. Postel και J. K. Reynolds, *Telnet protocol specification*, 1983.

[14] «Hacker Group Says Program Can Exploit Microsoft Security Hole», επίσκεψη 10 Απρ. 2025. διεύθν.: https://archive.nytimes.com/www.nytimes.com/library/tech/98/08/cyber/articles/04hacker.html.

[15] *Internet Relay Chat Protocol*, RFC 1459, Μάι. 1993. doi: 10.17487/RFC1459. διεύθν.: https://www.rfc-editor.org/info/rfc1459.

[16] D. Dagon, G. Gu, C. P. Lee και W. Lee, «A Taxonomy of Botnet Structures», στο *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, 2007, σσ. 325–339. doi: 10.1109/ACSAC.2007.44.

[17] G. Kambourakis, M. Anagnostopoulos, W. Meng και P. Zhou, *Botnets : Architectures, Countermeasures, and Challenges*. CRC Press, 2019, σσ. 15–27.

[18] G. Gu, J. Zhang και W. Lee, «BotSniffer: Detecting botnet command and control channels in network traffic», 2008.

[19] J. Zhuge, T. Holz, X. Han, J. Guo και W. Zou, «Characterizing the irc-based botnet phenomenon», *None*, 2007.

[20] Strike Source, *The Evolution of Command-and-Control Servers*, https://strikesource.com/the-evolution-of-command-and-control-servers/, Σεπτ. 2023. επίσκεψη 25 Μαρ. 2025.

[21] Malware Patrol, *The Evolution of C2 Communication: Custom TCP Protocols*, https://www.malwarepatrol.net/the-evolution-of-c2-communication-channels/, 2024. επίσκεψη 25 Μαρ. 2025.

[22] S. Josefsson, *The base16, base32, and base64 data encodings*, 2006.

[23] *Zeus Malware: Threat Banking Industry*, 2010. διεύθν.: https://botnetlegalnotice.com/citadel/files/Guerrino_Decl_Ex1.pdf.

[24] A. K. Sood, R. J. Enbody και R. Bansal, «Dissecting SpyEye--Understanding the design of third generation botnets», *Computer Networks*, τόμ. 57, αρθμ. 2, σσ. 436–450, 2013.

[25] B. Stone-Gross κ.ά., «Your botnet is my botnet: analysis of a botnet takeover», στο *Proceedings of the 16th ACM Conference on Computer and Communications Security*, σειρά CCS '09, Chicago, Illinois, USA: Association for Computing Machinery, 2009, σσ. 635–647, isbn: 9781605588940. doi: 10.1145/1653662.1653738. διεύθν.: https://doi.org/10.1145/1653662.1653738.

[26] Symantec Security Response, *SpyEye Bot versus Zeus Bot*, White Paper, 2010. επίσκεψη 28 Μαρ. 2025. διεύθν.: https://community.broadcom.com/symantecenterprise/viewdocument/spyeye-bot-versus-zeus-bot.

[27] J. Pake. «Binary Protocol - Complex Security», Complexsecurity.io, επίσκεψη 11 Απρ. 2025. διεύθν.: https://knowledge.complexsecurity.io/protocols/binary/.

[28] Mandiant (Google Cloud), *M-Trends 2024*, 2024. επίσκεψη 29 Μαρ. 2025. διεύθν.: https://services.google.com/fh/files/misc/m-trends-2024.pdf.

[29] «Technique T1572: Protocol Tunneling», The MITRE Corporation, επίσκεψη 29 Μαρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1572/.

[30] «Technique T1102: Web Service», The MITRE Corporation, επίσκεψη 29 Μαρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1102/.

[31] «Technique T1568: Dynamic Resolution», The MITRE Corporation, επίσκεψη 29 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1568/.

[32] N. Falliere, L. O. Murchu και E. Chien, *W32.Stuxnet Dossier*, Security Response Deep Dive, Φεβ. 2011. επίσκεψη 29 Μαρ. 2025. διεύθν.: https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en.

[33] Kaspersky, *Stuxnet explained: What it is, who created it and how it works*, www.kaspersky.com, Απρ. 2023. διεύθν.: https://www.kaspersky.com/resource-center/definitions/what-is-stuxnet.

[34] CrySyS Lab (Laboratory of Cryptography and System Security), *sKyWIper (a.k.a. Flame): A complex malware for targeted attacks*, Technical Report, Μάι. 2012. επίσκεψη 30 Μαρ. 2025. διεύθν.: https://www.crysys.hu/skywiper/skywiper.pdf.

[35] Kaspersky Lab Expert. «The Mystery of Duqu: Part One (Q&A)», επίσκεψη 29 Μαρ. 2025. διεύθν.: https://securelist.com/the-mystery-of-duqu-part-one/31177/.

[36] «Metasploit Framework», Rapid7, επίσκεψη 30 Μαρ. 2025. διεύθν.: https://www.metasploit.com/.

[37] «Cobalt Strike: Adversary Simulation and Red Team Operations», Fortra, επίσκεψη 30 Μαρ. 2025. διεύθν.: https://www.cobaltstrike.com/.

[38] BC Security. «BC-SECURITY/Empire: Empire is a post-exploitation framework», επίσκεψη 30 Μαρ. 2025. διεύθν.: https://github.com/BC-SECURITY/Empire.

[39] «Sliver Framework», Bishop Fox, επίσκεψη 30 Μαρ. 2025. διεύθν.: https://github.com/BishopFox/sliver.

[40] «Malleable Command and Control», Fortra / Cobalt Strike, επίσκεψη 30 Μαρ. 2025. διεύθν.: https://www.cobaltstrike.com/help-malleable-c2.

[41] S. Matthews, «Internet Ethics», *International Encyclopedia of Ethics*, σσ. 1–12, 2013.

[42] T. Riebe και C. Reuter, «Dual-use and dilemmas for cybersecurity, peace and technology assessment», *Information Technology for Peace and Security: IT Applications and Infrastructures in Conflicts, Crises, War, and Peace*, σσ. 165–183, 2019.

[43] The Tor Project, *Domain fronting is critical to the open internet — and it's disappearing*, https://blog.torproject.org/domain-fronting-critical-open-web/, Μάι. 2018. επίσκεψη 1 Απρ. 2025.

[44] Lsec, *Utilizing Discord as C2 Traffic Broker - Lsec - Medium*, Medium, Απρ. 2024. επίσκεψη 15 Απρ. 2025. διεύθν.: https://medium.com/@lsecqt/utilizing-discord-as-c2-traffic-broker-b6a4d2f93bb3.

[45] «LOLBAS - Living Off The Land Binaries, Scripts and Libraries», επίσκεψη 1 Απρ. 2025. διεύθν.: https://lolbas-project.github.io/.

[46] «Technique T1218: System Binary Proxy Execution», The MITRE Corporation, επίσκεψη 15 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1218/.

[47] B. Guembe, A. Azeta, S. Misra, V. C. Osamor, L. Fernandez-Sanz και V. Pospelova, «The Emerging Threat of Ai-driven Cyber Attacks: A Review», *Applied Artificial Intelligence*, τόμ. 36, 1 2022, issn: 10876545. doi: 10.1080/08839514.2022.2037254.

[48] B. T. Bakken, I. Lund-Kordahl και E. Bjurström, «AI in Future C2--Who's in Command When AI Takes Control?», 2023.

[49] Secureworks Counter Threat Unit (CTU) Research Team, *Storm Worm DDoS Attack Threat Analysis & Report*, Φεβ. 2007. επίσκεψη 2 Απρ. 2025. διεύθν.: https://www.secureworks.com/research/storm-worm.

[50] C. Kanich κ.ά., «Spamalytics: An empirical analysis of spam marketing conversion», στο *Proceedings of the 15th ACM conference on Computer and communications security*, 2008, σσ. 3–14.

[51] «Tactic TA0011: Command and Control», The MITRE Corporation, επίσκεψη 6 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0011/.

[52] M. Sikorski και A. Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. San Francisco, CA, USA: No Starch Press, 2012.

[53] R. Sharpe, «Just what is SMB?», *Oct*, τόμ. 8, σ. 9, 2002.

[54] K. Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*. New York: Crown, 2014, isbn: 978-0770436179.

[55] FireEye Threat Intelligence, *Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor*, Δεκ. 2020. επίσκεψη 16 Απρ. 2025. διεύθν.: https://cloud.google.com/blog/topics/threat-intelligence/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor/.

[56] Microsoft Threat Intelligence Center (MSTIC) and Microsoft 365 Defender Research Team, *Analyzing Solorigate, the compromised SolarWinds binary*, Microsoft Security Blog, 2020. επίσκεψη 16 Απρ. 2025. διεύθν.: https://www.microsoft.com/en-us/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/.

[57] M. R. Gupta, Y. P. Koli, V. A. Patiyane και K. P. Wagh, «Eternal blue vulnerability», *International Journal for Research in Applied Science & Engineering Technology*, τόμ. 11, αρθμ. 6, 2023.

[58] M. Aljaidi κ.ά., «Nhs wannacry ransomware attack: Technical explanation of the vulnerability, exploitation, and countermeasures», στο *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, IEEE, 2022, σσ. 1–6.

[59] Malwarebytes Labs. «The worm that spreads WanaCrypt0r», επίσκεψη 16 Απρ. 2025. διεύθν.: https://www.malwarebytes.com/blog/news/2017/05/the-worm-that-spreads-wanacrypt0r.

[60] Kaspersky Global Research & Analysis Team (GReAT). «WannaCry FAQ: What you need to know today», επίσκεψη 16 Απρ. 2025. διεύθν.: https://securelist.com/wannacry-faq-what-you-need-to-know-today/78411/.

[61] Cybersecurity and Infrastructure Security Agency (CISA), *Alert (AA20-280A): Emotet Malware*, Οκτ. 2020. επίσκεψη 17 Απρ. 2025. διεύθν.: https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-280a.

[62] M. Antonakakis κ.ά., «Understanding the Mirai Botnet», στο *Proceedings of the 26th USENIX Security Symposium (USENIX Security '17)*, 2017, σσ. 1093–1110. επίσκεψη 17 Απρ. 2025. διεύθν.: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis.

[63] CrowdStrike Intelligence Team, *Big Game Hunting with Ryuk: Another Lucrative Targeted Ransomware*, Ιαν. 2019. επίσκεψη 17 Απρ. 2025. διεύθν.: https://www.crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/.

[64] «Meterpreter», Offensive Security, επίσκεψη 18 Απρ. 2025. διεύθν.: https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/.

[65] D. Kennedy, J. O'Gorman, D. Kearns και M. Aharoni, *Metasploit: The Penetration Tester's Guide*. San Francisco, CA, USA: No Starch Press, 2011, isbn: 978-1593272883.

[66] J. A. Donenfeld, «WireGuard: Next Generation Kernel Network Tunnel.», στο *NDSS*, 2017, σσ. 1–12.

[67] «BishopFox/sliver: Adversary Emulation Framework», Bishop Fox, επίσκεψη 18 Απρ. 2025. διεύθν.: https://github.com/BishopFox/sliver.

[68] Microsoft Security Threat Intelligence, «Looking for the 'Sliver' lining: Hunting for emerging command-and-control frameworks», Αύγ. 2022. επίσκεψη 19 Απρ. 2025. διεύθν.: https://www.microsoft.com/en-us/security/blog/2022/08/24/looking-for-the-sliver-lining-hunting-for-emerging-command-and-control-frameworks/.

[69] BC Security. «BC-SECURITY/Starkiller: Starkiller is a Frontend for PowerShell Empire.», επίσκεψη 19 Απρ. 2025. διεύθν.: https://github.com/BC-SECURITY/Starkiller.

[70] T. Nelson και H. Kettani, «Open source powershell-written post exploitation frameworks used by cyber espionage groups», στο *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, IEEE, 2020, σσ. 451–456.

[71] Verizon Enterprise Solutions, *2018 Data Breach Investigations Report*, 2018. επίσκεψη 20 Απρ. 2025. διεύθν.: https://www.verizon.com/business/resources/reports/DBIR_2018_Report.pdf.

[72] C5pider. «The Havoc Framework», επίσκεψη 20 Απρ. 2025. διεύθν.: https://github.com/HavocFramework/Havoc.

[73] Zscaler ThreatLabz, *Havoc Across the Cyberspace*, Φεβ. 2023. επίσκεψη 21 Απρ. 2025. διεύθν.: https://www.zscaler.com/blogs/security-research/havoc-across-cyberspace.

[74] «Execution - Enterprise Tactic TA0002», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0002/.

[75] «T1059: Command and Scripting Interpreter», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1059/.

[76] «T1059.001: PowerShell», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1059/001/.

[77] «T1059.003: Windows Command Shell», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1059/003/.

[78] «Discovery - Enterprise Tactic TA0007», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0007/.

[79] «T1082: System Information Discovery», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1082/.

[80] «T1033: System Owner/User Discovery», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1033/.

[81] «T1016: System Network Configuration Discovery», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1016/.

[82] «T1057: Process Discovery», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1057/.

[83] «Collection - Enterprise Tactic TA0009», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0009/.

[84] «T1113: Screen Capture», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1113/.

[85] «Command and Control - Enterprise Tactic TA0011», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0011/.

[86] «T1071: Application Layer Protocol», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1071/.

[87] «T1071.001: Web Protocols», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1071/001/.

[88] «T1573: Encrypted Channel», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1573/.

[89] «T1573.001: Symmetric Cryptography», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1573/001/.

[90] «T1571: Non-Standard Port», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1571/.

[91] «Exfiltration - Enterprise Tactic TA0010», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0010/.

[92] «T1041: Exfiltration Over C2 Channel», The MITRE Corporation, επίσκεψη 21 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1041/.

[93] «PowerShell Documentation», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/.

[94] «Creating .NET and COM Objects in Powershell», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/scripting/samples/creating-.net-and-com-objects--new-object-?view=powershell-7.5.

[95] «Invoke-Expression (IEX) - PowerShell», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/invoke-expression?view=powershell-7.4.

[96] «About PowerShell.exe - PowerShell (EncodedCommand parameter)», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_powershell_exe?view=powershell-7.4#-encodedcommand.

[97] «PowerShell - About Execution Policies», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.4.

[98] Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_logging?view=powershell-7.4.

[99] «Antimalware Scan Interface (AMSI) - Win32 apps», Microsoft, επίσκεψη 22 Απρ. 2025. διεύθν.: https://learn.microsoft.com/en-us/windows/win32/amsi/antimalware-scan-interface-portal.

[100] «S3cur3Th1sSh1t/Amsi-Bypass-Powershell», S3cur3Th1sSh1t, επίσκεψη 22 Απρ. 2025. διεύθν.: https://github.com/S3cur3Th1sSh1t/Amsi-Bypass-Powershell.

[101] A. J. Rose, S. R. Graham, C. M. Schubert Kabban, J. J. Krasnov και W. C. Henry, «ScriptBlock Smuggling: Uncovering Stealthy Evasion Techniques in PowerShell and. NET Environments», *Journal of Cybersecurity and Privacy*, τόμ. 4, αρθμ. 2, σσ. 153–166, 2024.

[102] J. F. Kurose και K. W. Ross, *Computer Networking: A Top-Down Approach*, 8th. Hoboken, NJ, USA: Pearson, 2020, σσ. 181–299, isbn: 978-0136681557.

[103] Postel, Jon (Editor). «Transmission Control Protocol - DARPA Internet Program Protocol Specification», επίσκεψη 23 Απρ. 2025. διεύθν.: https://www.rfc-editor.org/info/rfc793.

[104] Rescorla, E. «The Transport Layer Security (TLS) Protocol Version 1.3», επίσκεψη 23 Απρ. 2025. διεύθν.: https://www.rfc-editor.org/info/rfc8446.

[105] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*. Boston, MA, USA: Addison-Wesley Professional, 2001, isbn: 978-0201615982.

[106] I. Fette και A. Melnikov, *The websocket protocol*, 2011.

[107] E. Chatzoglou και G. Kambourakis, «C3: Leveraging the Native Messaging Application Programming Interface for Covert Command and Control», *Future Internet*, τόμ. 17, αρθμ. 4, σ. 172, 2025.

[108] J. B. Althouse, J. Atkinson και J. Atkins, *TLS Fingerprinting with JA3 and JA3S*, Salesforce Engineering Blog, Σεπτ. 2017. επίσκεψη 25 Απρ. 2025. διεύθν.: https://engineering.salesforce.com/tls-fingerprinting-with-JA3-and-JA3S-247362855967/.

[109] «JA3/S - TLS Fingerprinting», Salesforce, επίσκεψη 25 Απρ. 2025. διεύθν.: https://github.com/salesforce/ja3.

[110] A. Chuvakin, K. Schmidt και C. Phillips, *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*. Syngress, 2013, isbn: 978-1597496353.

[111] Malavika Balachandran Tadeusz. «What is DNS tunneling? | How to detect DNS tunneling», επίσκεψη 26 Απρ. 2025. διεύθν.: https://blog.cloudflare.com/cloudflare-one-intel/.

[112] V. Palacín, *Practical Threat Intelligence and Data-Driven Threat Hunting: A hands-on guide to threat hunting with the ATT&CK™ Framework and open source tools*, 1η έκδοση. Packt Publishing, 2021, Καλύπτει τις αρχές της ανίχνευσης και απόκρισης σε τελικά σημεία, συμπεριλαμβανομένης της ανάλυσης συμπεριφοράς, σάρωσης μνήμης και παρακολούθησης μηχανισμών εδραίωσης παρουσίας., isbn: 1838556370; 9781838556372.

[113] «Defense Evasion - Enterprise Tactic TA0005», The MITRE Corporation, επίσκεψη 28 Απρ. 2025. διεύθν.: https://attack.mitre.org/tactics/TA0005/.

[114] «T1562.001: Disable or Modify Tools», The MITRE Corporation, επίσκεψη 28 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1562/001/.

[115] «Technique T1090: Proxy», The MITRE Corporation, επίσκεψη 28 Απρ. 2025. διεύθν.: https://attack.mitre.org/techniques/T1090/.