ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Συστήματα Ανίχνευσης/Πρόληψης Εισβολής και MISP – συνδυάζοντας τεχνολογίες Ανοιχτού Κώδικα για ενίσχυση της Ασφάλειας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Αυγουστάκη Χρυσοβαλάντη

Επιβλέπων : Μαλιάτσος Κωνσταντίνος

Μέλη εξεταστικής επιτροπής: Μαρία Καρύδα
Βασιλική Διαμαντοπούλου

Σάμος, Ιούνιος 2024

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Καταρχάς, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κύριο Κωνσταντίνο Μαλιάτσο για την εμπιστοσύνη που μου έδειξε, καθώς και για την καθοδήγησή του κατά την εκπόνηση της διπλωματικής εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω πρώτα την οικογένειά μου, αλλά και όλους εκείνους, που με ενθάρρυναν και με υποστήριζαν αδιάλειπτα στη διάρκεια αυτού του Μεταπτυχιακού Προγράμματος Σπουδών.

ΑΥΓΟΥΣΤΑΚΗ ΧΡΥΣΟΒΑΛΑΝΤΗ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Ακρωνύμια

ATT&CK	Adversarial Tactics, Techniques and Common Knowledge	Σελ.51
(τριάδα) **CIA**	Confidentiality, Integrity and Availability	Σελ.10
CTI	Cyber Threat Intelligence	Σελ. 87
CVE	Common Vulnerabilities and Exposures	Σελ. 56
CWE	Common Weakness Enumeration	Σελ. 56
D3FEND	Detection, Denial and Disruption Framework Empowering Network Defense	Σελ. 57
DoS	Denial of Service	Σελ. 43
DDoS	Distributed Denial of Service	Σελ. 44
FOSS (**ΕΛ/ΛΑΚ**)	Free and Open Source Software (Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα)	Σελ. 5, 7, 8 (υποσημ. 14)
(**GNU**) **GPL**	(GNU) General Public License	Σελ. 6
HIDS	Host-based IDS	Σελ. 73
HIPS	Host-based IPS	Σελ. 73
IDPS	Intrusion Detection and Prevention System	Σελ. 70
IDS	Intrusion Detection System	Σελ. 69
IoC	Indication of Compromise	Σελ. 88
IPS	Intrusion Prevention System	Σελ. 69, 70
NIDS	Network-based IDS	Σελ. 74
NIPS	Network-based IPS	Σελ. 74
NVD	National Vulnerability Database	Σελ. 56
TIP	Threat Intelligence Platform	Σελ. 87

Περίληψη

Σε έναν κόσμο που η εξάρτηση από τις ψηφιακές τεχνολογίες και το Διαδίκτυο μεγαλώνει διαρκώς, η Ασφάλεια Πληροφοριών έχει βαρύνουσα σημασία. Οι τεχνολογικές εξελίξεις αλλάζουν τάχιστα το τοπίο και δίνουν νέες δυνατότητες στους απανταχού κακόβουλους. Τεχνολογίες και μέτρα που στο πρόσφατο παρελθόν ήταν επαρκή, πλέον εμφανίζουν αδυναμίες. Οι κακόβολοι παράγοντες διαθέτουν ένα μεγάλο πλήθος από εξειδικευμένα εργαλεία, τεχνικές, τακτικές και διαδικασίες που χρησιμοποιούν ως οπλοστάσιο για να επιτύχουν τους σκοπούς τους.

Τα Συστήματα Ανίχνευσης Εισβολής και τα Συστήματα Πρόληψης Εισβολής είναι τα εργαλεία που καλούνται να δώσουν απαντήσεις σε αυτά τα προβλήματα. Παλιές και αξιόπιστες μέθοδοι, όπως η χρήση υπογραφών, αξιοποιούνται σε συνδυασμό με καινοτόμες λύσεις, όπως η συμπεριφοριστική ανάλυση, για να ανιχνεύσουν και να αποτρέψουν κακόβουλες δραστηριότητες.

Σε ένα τέτοιο περιβάλλον αναδεικνύεται και μια άλλη προσέγγιση· όπως ακριβώς η φιλοσοφία του ανοικτού κώδικα υποστηρίζει τον διαμοιρασμό της γνώσης και του κώδικα για την παραγωγή καλύτερου λογισμικού, έτσι και ο διαμοιρασμός της γνώσης και των πληροφοριών που υπάρχουν για τους κακόβουλους παράγοντες μπορεί να βοηθήσει ώστε τα συστήματα να γίνουν πιο ασφαλή. Αυτή η φιλοσοφία προωθείται από τις Πλατφόρμες Διαμοιρασμού Πληροφοριών για τις Απειλές ασφάλειας· ειδικοί συλλέγουν πληροφορίες σχετικά με κακόβουλους παράγοντες και τις δράσεις τους και τις μοιράζονται με άλλους, με σκοπό την έγκαιρη και αποτελεσματική εξουδετέρωσή των απειλών.

Αυτή η διπλωματική εστιάζει στον συνδυασμό των δύο παραπάνω προσεγγίσεων. Στην αύξηση της ασφάλειας μέσω του συνδυασμού ενός Συστήματος Ανίχνευσης Εισβολής / Συστήματος Πρόληψης Εισβολής με μια Πλατφόρμα Διαμοιρασμού Πληροφοριών για τις Απειλές (πιο συγκεκριμένα το MISP). Τα δύο συστήματα ανταλλάσσουν τα απαραίτητες δεδομένα με αυτόματο τρόπο, ώστε να επιταχύνουν τη διαδικασία της εξουδετέρωσης της απειλής.

Λέξεις Κλειδιά: *Συστήματα Ανίχνευσης Εισβολής, IDS, Συστήματα Πρόληψης Εισβολής, IPS, MISP, Τεχνολογίες Ανοικτού Κώδικα, Ασφάλεια Πληροφοριών*

Abstract

In a world where dependance on digital technologies and the Internet is constantly increasing, Information Security is becoming more significant than ever. Technological developments are changing the landscape and offer new capabilities to the malicious actors everywhere in the world. Technologies and measures that where sufficient in the recent past, now become weak. Malicious actors have numerous specialized tools, techniques, tactics and procedures at their disposal, to use as an arsenal to accomplish their goals.

Intrusion Detection Systems and Intrusion Prevention Systems are the tools to give answers to these problems. Old and trustworthy methods, like the use of signatures, are used in combination with innovative solutions, like behavioral analysis, to detect and prevent malicious actions.

In an environment like this, there is another approach as well; just like open source philosophy advocates for sharing knowledge and code to make better software, sharing knowledge and information about the malicious actors can help to make systems more secure. This philosophy is promoted by Threat Intelligence Platforms; specialists gather information about malicious actors and their actions and share this intelligence with others to help them neutralize the threats efficiently and on time.

This dissertation focuses on the combination of the two approaches. Increasing security by combining the use of an Intrusion Detection System / Intrusion Prevention System with a Threat Intelligence Platform (MISP in particular). These two systems exchange the necessary data in an automatic way to speed up the process of neutralizing a threat.

Keywords: *Intrusion Detection Systems, IDS, Intrusion Prevention Systems, IPS, MISP, Open Source Technologies, Information Security*

1

Εισαγωγή

1.1 Ασφάλεια Πληροφοριών στον Κυβερνοχώρο

Στις μέρες μας, ολοένα και μεγαλύτερο κομμάτι των οικονομικών, των εμπορικών, των δραστηριοτήτων διακυβέρνησης, αλλά και άλλων δραστηριοτήτων, διεξάγεται στον κυβερνοχώρο [1] (Yuchong & Qinghui, 2021). Αυτό είναι δυνατό χάρη στην ταχεία ανάπτυξη των τεχνολογιών των Πληροφοριακών Συστημάτων, που δίνουν στον κυβερνοχώρο μεγαλύτερη ανοικτότητα, δυναμικότητα και ευελιξία (Wu, Li, & Ji, 2018).

Ταυτόχρονα όμως, η αυξανόμενη έκθεση των ανθρώπινων δραστηριοτήτων στον κυβερνοχώρο, σε συνδυασμό με το χαμηλό κόστος χρήσης, την παροχή ανωνυμίας, την έλλειψη δημόσιας διαφάνειας κ.α. έχουν δημιουργήσει προβλήματα όπως το *κυβερνοέγκλημα* (*cyber-crime*), οι *κυβερνοεπιθέσεις* (*cyber-attacks*), η *κυβερνοκατασκοπία* (*cyber-espionage*), η *κυβερνοτρομοκρατία* (*cyber-terrorism*) και ο *κυβερνοπόλεμος* (*cyber-warfare*) (Yuchong & Qinghui, 2021).

Απέναντι σε αυτές τις απειλές που εμφανίζονται στον κυβερνοχώρο, ο τομέας της Ασφάλειας Πληροφοριών ασχολείται με τη μελέτη πρακτικών, αλλά και εργαλείων για την αντιμετώπισή τους.

[1] Ο **κυβερνοχώρος** (**cyberspace**) ως έννοια δεν έχει έναν ορισμό που να έχει επικρατήσει. Ο Jose Antonio Hernandez Castillo (2023) λέει πως «*αναφέρεται σε έναν εικονικό χώρο, όπου άτομα και οργανισμοί αλληλοεπιδρούν και διαμοιράζονται πληροφορίες σε παγκόσμιο επίπεδο, ανεξαρτήτως της φυσικής τους θέσης».*

Συχνά ο όρος κυβερνοχώρος χρησιμοποιείται ως συνώνυμο του Διαδικτύου, όμως ως έννοιες δεν επικαλύπτονται πλήρως. Ο Folsom (2007) αναφέρει πως το Διαδίκτυο είναι απλά μια πύλη για τον κυβερνοχώρο, αλλά σίγουρα όχι η μόνη (λ.χ. αναφέρει πως το τηλεφωνικό δίκτυο είναι μια άλλη πύλη).

1.2 Αντικείμενο της διπλωματικής

Σκοπός της παρούσας διπλωματικής είναι να αναδείξει πως η προστασία των Πληροφοριακών Συστημάτων (μεγάλων, μεσαίων ή μικρών) απέναντι στις απειλές που υπάρχουν στο Διαδίκτυο, δεν είναι και δεν μπορεί να είναι στατική. Ο τομέας της Ασφάλειας Πληροφοριών έχει βασικές αρχές, που όμως δεν μπορούν να καλύψουν όλα τα ενδεχόμενα. Ο κυβερνοχώρος είναι ένα δυναμικό περιβάλλον που διαρκώς εμπλουτίζεται με νέες τεχνολογίες και προσφέρει σε κακόβουλους χρήστες την ευκαιρία να αναπροσαρμόζουν τις τακτικές, τις μεθόδους και τα εργαλεία τους για να επιτύχουν τους στόχους τους.

Από την πλευρά τους, οι νόμιμοι χρήστες του κυβερνοχώρου θα πρέπει επίσης να προσαρμόζονται διαρκώς για να προστατευτούν από τις νέες απειλές. Στα πλαίσια της εργασίας θα αναδειχθεί πώς εργαλεία που κάποτε ήταν επαρκή, πλέον δεν είναι. Οι νέες λύσεις για την Ασφάλεια Πληροφοριών έρχονται μέσα από την ενδελεχή μελέτη για την κατανόηση αφενός των κινήτρων και των στόχων και αφετέρου των τακτικών, των μεθόδων, ακόμα και των εργαλείων που αξιοποιούνται από κακόβουλα άτομα για κάθε είδους παραβίαση της ασφάλειας.

Εστιάζοντας στα Συστήματα Ανίχνευσης / Πρόληψης Εισβολής η διπλωματική εργασία αναδεικνύει πως αξιοποιούνται για να καλύψουν τις αδυναμίες των κλασικών εργαλείων προστασίας, περιλαμβάνοντας τους απαραίτητους μηχανισμούς που είτε θα βοηθήσουν στον εντοπισμό των σημείων όπου απαιτείται κάποια αναπροσαρμογή, είτε θα προκαλέσουν αυτόματα μια αναπροσαρμογή. Επιπλέον, εξετάζεται πώς το πνεύμα και η φιλοσοφία της ανοικτότητας, που προωθείται από τις τεχνολογίες ανοικτού κώδικα, μπορούν να προσφέρουν ακόμα μεγαλύτερα οφέλη. Σε αυτή την περίπτωση η φιλοσοφία της ανοικτότητας παίρνει τη μορφή του διαμοιρασμού πληροφοριών που αφορούν κακόβουλες δράσεις και λαμβάνουν χώρα στον κυβερνοχώρο. Σκοπός είναι να βοηθηθεί η αναγνώριση των ίδιων απειλών σε άλλα συστήματα και τελικά να επιταχυνθεί η αντιμετώπισή τους, προτού καταφέρουν να προκαλέσουν ζημιά στο προσβαλλόμενο σύστημα.

Τέλος, η εργασία έχει σκοπό να αναδείξει πώς η ενοποίηση ενός Συστήματος Ανίχνευσης / Πρόληψης Εισβολής με Πλατφόρμες Διαμοιρασμού Πληροφοριών που αφορούν κακόβουλες δραστηριότητες (TIP) μπορεί να πάει την προστασία ένα επίπεδο παραπάνω. Το Σύστημα Ανίχνευσης παρακολουθεί το υπολογιστικό σύστημα και αν κρίνει ότι υπάρχει μια ύποπτη δραστηριότητα, αξιοποιεί την πλατφόρμα TIP για να εξακριβώσει αν πράγματι είναι απειλή. Σε περίπτωση που επιβεβαιωθεί ως κακόβουλη, τότε το Σύστημα Πρόληψης μπορεί να προχωρήσει αυτόματα στις κατάλληλες ενέργειες για την εξουδετέρωσή της. Μια άλλη δυνατότητα προκύπτει από την περίπτωση που αναγνωριστεί μια άγνωστη μέχρι εκείνη τη στιγμή απειλή· σ' αυτή την περίπτωση είναι το Σύστημα Ανίχνευσης που ενημερώνει τη πλατφόρμα TIP, ώστε την επόμενη φορά που θα εμφανιστεί η ίδια απειλή, το σύστημα να την εξουδετερώσει ταχύτερα.

Αξίζει να σημειωθεί πως η ενοποίηση δεν είναι τετριμμένη υπόθεση. Σε περιπτώσεις που δεν παρέχεται ως λύση out of the box, απαιτείται καλή μελέτη των συστημάτων που επιλέγονται και κατανόηση των δυνατοτήτων που προκύπτουν από την παραμετροποίησή τους. Σε αυτή την ενοποίηση, ρόλο παίζει και το υποκείμενο λειτουργικό σύστημα, αφού, πέρα από την παραμετροποίηση των δύο πλατφόρμων, πρέπει να αναπτυχθεί και ένα σενάριο εντολών στον ρόλο του διαμεσολαβητή, που θα «μεταφράζει» τα δεδομένα εκατέρωθεν.

1.3 Δομή της διπλωματικής

Στο κεφάλαιο 2 παρουσιάζονται η φιλοσοφία και τα χαρακτηριστικά του Ανοικτού Κώδικα. Στο κεφάλαιο 3 γίνεται η απαραίτητη αναφορά στις βασικές έννοιες της Ασφάλειας Πληροφοριών, ενώ δίνεται ιδιαίτερη έμφαση στον αντίκτυπο των απειλών. Το κεφάλαιο 4 ασχολείται με τις κυβερνοεπιθέσεις, αναλύοντας ποια είναι τα κυριότερα διανύσματα επίθεσης, αλλά και ποια είναι τα κυριότερα είδη επιθέσεων. Επίσης, γίνεται ιδιαίτερη αναφορά στα μοντέλα μελέτης των επιθέσεων και γίνεται μια σύγκρισή τους. Στο κεφάλαιο 5 εξετάζεται η κυβερνοάμυνα:· αναλύονται βασικές έννοιες, παρουσιάζονται εργαλεία που βοηθούν στην οργάνωσή της και παρουσιάζονται οι βασικότεροι μηχανισμοί και συστήματα για την επίτευξή της. Το Κεφάλαιο 6 εξειδικεύει στα Συστήματα Ανίχνευσης / Πρόληψης Εισβολής. Αναφέρεται στις τεχνικές ανίχνευσης που χρησιμοποιούν, τις πηγές από τις οποίες αντλούν τα δεδομένα προς έλεγχο, τους διαφορετικούς τύπους των συστημάτων αυτών, την τυπική δομή τους, τις καταστάσεις λειτουργίας τους, τις βασικές αλλά και τις πρόσθετες δυνατότητές τους. Ξεχωριστή αναφορά γίνεται για τις δυνατότητες απόκρισης που έχουν τα Συστήματα Πρόληψης Εισβολής, αλλά και τη σχέση που υπάρχει ανάμεσα στα Συστήματα Ανίχνευσης / Πρόληψης Εισβολής με άλλα συστήματα κυβερνοάμυνας. Παρουσιάζονται επίσης και επιθέσεις που στοχεύουν στα ίδια τα συστήματα αυτά. Το κεφάλαιο ολοκληρώνεται με μια ενότητα στην οποία εξετάζονται τα πλεονεκτήματα του διαμοιρασμού πληροφοριών και τα οφέλη της χρήσης των πλατφόρμων TIP. Στο κεφάλαιο 7 παρουσιάζονται το Σύστημα Ανίχνευσης / Πρόληψης Εισβολής Wazuh και η πλατφόρμα διαμοιρασμού πληροφοριών που αφορούν κακόβουλες δραστηριότητες MISP. Το κεφάλαιο παρουσιάζει τρία σενάρια μελέτης που σκοπό έχουν να αποδείξουν πώς μέσα από τη συνεργασία των δύο προαναφερόμενων μερών η ασφάλεια του συστήματος γίνεται πιο αποδοτική. Τέλος, στο κεφάλαιο 8 παρουσιάζονται τα συμπεράσματα της εργασίας.

2

Τεχνολογίες Ανοικτού Κώδικα

Η εργασία αυτή ασχολείται με τεχνολογίες **Ανοικτού Κώδικα** (**Open Source**), οπότε είναι πρέπον να ξεκινήσουμε με μια αναφορά στη φιλοσοφία και τα ιδιαίτερα χαρακτηριστικά του. Όμως, για να γίνει κατανοητή η φιλοσοφία του, είναι απαραίτητο να γνωρίσουμε πώς διαμορφώθηκε μέσα στην ευρύτερη ιστορία της *Τεχνολογίας των Πληροφοριών και των Επικοινωνιών* (*ΤΠΕ*) και ειδικότερα αυτή των *Ηλεκτρονικών Υπολογιστών* (*Η/Υ*).

Η ιστορία [2] λοιπόν, ξεκινά μέσα σε ακαδημαϊκά ιδρύματα, όπως το Τεχνολογικό Ινστιτούτο Μασαχουσέτης (MIT), με την απόκτηση των πρώτων *μίνι υπολογιστών* (*mini computers*). Μέσα από την ακαδημαϊκή έρευνα δημιουργήθηκε και αναπτύχθηκε μια κοινότητα ατόμων που ασχολούνταν με την επιστήμη των υπολογιστών, συνεργάζονταν, εφευρίσκαν εργαλεία προγραμματισμού και έχτιζαν μια νέα κουλτούρα – αυτήν του *χάκερ* (*hacker*) [3]. Η εξάπλωση του *ARPANET* [4] σε όλη την ηπειρωτική επικράτεια των ΗΠΑ και η διασύνδεση των πανεπιστημίων σε αυτό, έδωσε το μέσο για την εξάπλωση και της κουλτούρας των χάκερ – μιας κουλτούρας που ενθάρρυνε τη συζήτηση, τον διαμοιρασμό σκέψεων, ιδεών, αλλά και του λογισμικού. (Raymond, A Brief History of Hackerdom, 1999; Bretthauer, 2001)

[2] Όσον αφορά το λογισμικό, γιατί σε άλλους τομείς των ανθρώπινων εγχειρημάτων τα πρώτα ψήγματα αυτής της φιλοσοφίας μπορούν να εντοπιστούν νωρίτερα.

[3] Αρχικά, ο όρος *χάκερ* (*hacker*) αναφερόταν σε έναν επιδέξιο επαγγελματία, που διέθετε υψηλές γνώσεις προγραμματισμού, αλλά και μεγάλο πάθος για την πρόοδο της επιστήμης των υπολογιστών. (Bretthauer, 2001) Ο όρος άρχισε να αποκτά την «κακή» του έννοια γύρω στα μέσα της δεκαετίας του 1980, όταν οι δημοσιογράφοι άρχισαν να τον χρησιμοποιούν για να αναφερθούν σε άτομα που εμπλέκονταν σε παράνομες ενέργειες με χρήση υπολογιστών. (Raymond, A Brief History of Hackerdom, 1999)

[4] *ARPANET*: ακρωνύμιο του *Advanced Research Projects Agency Network*. Το δίκτυο αυτό αποτελεί τον πρόγονο του σημερινού Διαδικτύου. Αξίζει να σημειωθεί ότι παρότι το ARPANET ήταν ένα έργο του Υπουργείου Άμυνας των ΗΠΑ, εξ αρχής συμπεριέλαβε τη διασύνδεση υπολογιστικών συστημάτων πανεπιστημιακών ιδρυμάτων.

Ένα δεύτερο ορόσημο σε αυτή την ιστορία είναι η εφεύρεση και η διάδοση χρήσης του λειτουργικού συστήματος *UNIX* και της γλώσσας προγραμματισμού *C* [5], που έδωσαν τη δυνατότητα της φορητότητας (portability) των προγραμμάτων. (Raymond, A Brief History of Hackerdom, 1999) Τα πρώτα κιόλας χρόνια του UNIX, το πανεπιστήμιο της Καλιφόρνια στο Berkeley απέκτησε ένα αντίγραφό του και έχτισε μια συνεργασία με τα εργαστήρια Bell για την εξέλιξη του συστήματος. Η στροφή του UNIX στον εμπορικό τομέα μερικά χρόνια αργότερα, άφησε την ερευνητική κοινότητα να ασχολείται με την ανάπτυξή μιας έκδοσης που ονομάστηκε *BSD* (*Berkeley Software Distribution*). Σημαντικές προσθήκες, όπως λ.χ. το σύστημα *TCP/IP* – που αποτελεί τη ραχοκοκαλιά του σύγχρονου Διαδικτύου – αναπτύχθηκαν και υλοποιήθηκαν στο BSD και από αυτό πέρασαν στο εμπορικό UNIX. (Bretthauer, 2001)

Σύντομα όμως προέκυψε ένα σοβαρό πρόβλημα· οι καινοτόμες επεκτάσεις του BSD ήταν «δεμένες» με μια πολύ υψηλού κόστους εμπορική άδεια χρήσης του UNIX. Αυτό τελικά οδήγησε στην έκδοση των καινοτομιών του BSD που δεν δένονταν με τον ιδιόκτητο κώδικα του UNIX με μια φιλελεύθερη άδεια χρήσης, στην οποία η μόνη δέσμευση ήταν να γίνεται αναφορά στο πανεπιστήμιο και σε όσους συνεισέφεραν στη δημιουργία του κώδικα. (Bretthauer, 2001)

Τρίτο ορόσημο της ιστορίας είναι η εξάπλωση της χρήσης των *μικροϋπολογιστών* (*micro computers*). Πολύ σύντομα μετά την εμφάνισή τους, οι μίνι υπολογιστές άρχισαν να χάνουν τη θέση τους στους χώρους των πανεπιστημίων, ενώ το λογισμικό που είχε αναπτυχθεί για αυτούς δεν ήταν δυνατό να λειτουργήσει στους νέους μικροϋπολογιστές. (Raymond, A Brief History of Hackerdom, 1999) Επιπλέον, οι νέοι μικροϋπολογιστές συνοδεύονταν από ιδιόκτητο λογισμικό που παραδιδόταν σε εκτελέσιμο κώδικα και όχι πηγαίο κώδικα (Bretthauer, 2001), οπότε η δυνατότητα προσθήκης νέων λειτουργιών, μέσω της συνεργασίας των χρηστών του, ήταν μηδενική. (Raymond, A Brief History of Hackerdom, 1999)

Μέσα σε αυτό το κλίμα, ο Richard Stallman, νοσταλγός του πρόσφατου παρελθόντος, εμπνεύστηκε το *Ελεύθερο Λογισμικό* (*Free* [6] *Software*), το οποίο υπόσχεται τέσσερις ελευθερίες σε όποιον το χρησιμοποιεί:

- Την ελευθερία να το εκτελεί για όποιον σκοπό επιθυμεί.
- Την ελευθερία να το τροποποιεί για να ικανοποιεί τις ανάγκες του (απαραίτητη προϋπόθεση για να επιτευχθεί αυτό είναι να δίνεται πρόσβαση στον πηγαίο κώδικά του).
- Την ελευθερία να παράγει αντίγραφά του, είτε δωρεάν, είτε έναντι αμοιβής.
- Την ελευθερία να αναδιανέμει τροποποιημένες εκδόσεις του, προς όφελος της κοινότητας.

(Bretthauer, 2001)

[5] Αξίζει να μνημονευθεί ότι και τα δύο παράχθηκαν στα εργαστήρια Bell, με τον Ken Thompson και τον Dennis Ritchie να είναι οι πρωτεργάτες στα δύο έργα αντιστοίχως. (Raymond, A Brief History of Hackerdom, 1999)

[6] Η λέξη free έχει διφορούμενη σημασία στην αγγλική γλώσσα, καθώς μπορεί να ερμηνευτεί ως *ελευθερία* ή ως *δωρεάν*. Για τον λόγο αυτό ο Stallman διευκρινίζει: «*So think of free speech, not free beer*», δηλαδή «*σκέψου την ελευθερία του λόγου, όχι τη δωρεάν μπύρα*». (Bretthauer, 2001) Η σύγχυση ωστόσο παραμένει, γιατί στην πράξη το μεγαλύτερο μέρος του ελεύθερου λογισμικού διανέμεται δωρεάν.

Για την προώθηση της ιδέας του ελεύθερου λογισμικού, το οποίο θα αποτελούσε εναλλακτική στο ιδιόκτητο λογισμικό, οραματίστηκε ένα νέο λειτουργικό σύστημα, παρόμοιο με το δημοφιλές UNIX, το οποίο ονόμασε *GNU* [7]. Και για να προστατέψει το ελεύθερο λογισμικό από ενσωμάτωση σε ιδιόκτητο λογισμικό, εμπνεύστηκε την έννοια του *copyleft* [8], η οποία εφαρμόζεται στην άδεια χρήσης GNU General Public License (GNU GPL ή απλά GPL). Βασική παράμετρος αυτής της άδειας – πέρα από ότι εξασφαλίζονται οι 4 ελευθερίες – είναι πως κάθε παράγωγο έργο του λογισμικού που προστατεύεται από αυτή την άδεια, θα πρέπει να διανέμεται υπό την ίδια άδεια [9]. (Bretthauer, 2001)

Η προσπάθεια ανάπτυξης του GNU είχε ως αποτέλεσμα την παραγωγή αρκετών σημαντικών προγραμμάτων για το λειτουργικό σύστημα, όμως υστερούσε σε ένα κομμάτι· την ανάπτυξη του πυρήνα (kernel) του λειτουργικού συστήματος. (Bretthauer, 2001) Σε αυτό το σημείο της ιστορίας μπαίνει ένας Φιλανδός φοιτητής ονόματι Linus Torvalds. Το 1991 δημοσίευσε τον πυρήνα *Linux* [10], που είχε αναπτύξει με προγράμματα από την εργαλειοθήκη του GNU. Το έργο του Torvalds σήμανε δύο πράγματα: πρώτον, χάρη στον πυρήνα Linux, το έργο GNU είχε επιτέλους έναν πυρήνα τον οποίο μπορούσε να πλαισιώσει για να παραχθεί ένα ολοκληρωμένο λειτουργικό σύστημα. Δεύτερον, ο Torvalds εισήγαγε ένα νέο μοντέλο ανάπτυξης λογισμικού – αντί να αναπτύσσεται αποκλειστικά από μια μικρή, κλειστή ομάδα που λειτουργεί βάσει αυστηρού σχεδίου και συντονισμού πριν προχωρήσει σε κάποιο βήμα της υλοποίησης, όπως στο παραδοσιακό μοντέλο ανάπτυξης λογισμικού, το Linux βασίστηκε στη δημοσίευση συχνών εκδόσεων και τη λήψη ανατροφοδότησης (προτάσεων για διορθώσεις και επεκτάσεις) από ένα μεγάλο πλήθος εθελοντών, που συντονίζονταν μέσα από το νεόφερτο δημόσιο Διαδίκτυο. (Raymond, A Brief History of Hackerdom, 1999)

Λίγα χρόνια αργότερα, ο Eric S. Raymond στο έργο του *The Cathedral and the Bazaar* (1999) προχώρησε σε μια σύγκριση μεταξύ του παλαιού, παραδοσιακού μοντέλου και του νέου μοντέλου ανάπτυξης που εισήγαγε το Linux και έκανε τα δυνατό το Διαδίκτυο, και μέσα από αυτό διατύπωσε μια άποψη που έτυχε μεγάλης αποδοχής: «*Given enough eyeballs, all bugs are shallow*», που θα μπορούσε να ερμηνευτεί ως εξής: «*αν υπάρχουν αρκετά μάτια (δηλαδή αρκετά άτομα που να ελέγχουν τον κώδικα), τότε όλα τα λάθη είναι επιπόλαια (δηλαδή εύκολο να εντοπιστούν και να διορθωθούν)*».

Έτσι λοιπόν, μέσα από μια διαδρομή δεκαετιών, η κουλτούρα των αυθεντικών χάκερ, το πνεύμα των ερευνητών που συνεισέφεραν στην εξέλιξη του BSD και εκείνων που συμμετείχαν στην ανάπτυξη ελεύθερου λογισμικού, φάνηκε να συγκλίνουν ξανά και να δίνουν μια νέα πνοή μέσα από το συνεργατικό μοντέλο ανάπτυξης που εισήγαγε ο Torvalds για το Linux και παρουσίασε στο ευρύ κοινό ο Raymond, δημιουργώντας ένα νέο κίνημα.

[7] GNU: αναδρομικό ακρωνύμιο της φράσης GNU's Not Unix. (Bretthauer, 2001)

[8] Το όνομα είναι μια προφανής προσπάθεια να τονιστεί η αντίθεση με το copyright ©.

[9] Η παράμετρος αυτή έχει μια σημαντική επίπτωση: αν ένα ιδιόκτητο λογισμικό ενσωματώσει ελεύθερο λογισμικό, θα πάψει να θεωρείται ιδιόκτητο και θα μετατραπεί σε ελεύθερο λογισμικό.

[10] Ο Torvalds άντλησε έμπνευση για το Linux από το λειτουργικό σύστημα Minix, το οποίο με τη σειρά του ήταν «κλώνος» του UNIX. Συνεπώς, ο πυρήνας Linux ανήκει στην «οικογένεια» του UNIX.

Τα πρώτα χρόνια αυτού του νέου μοντέλου, η άδεια GPL παρείχε το νομικό πλαίσιο για τη διανομή του λογισμικού. Όμως, υπήρχαν κάποιες πτυχές του ελεύθερου λογισμικού που προκαλούσαν αρκετό προβληματισμό. Πιο συγκεκριμένα, ένα διαρκές θέμα ήταν πως ο όρος free software εκλαμβανόταν από τους νεόφερτους ως «*δωρεάν λογισμικό*» (δείτε υποσημείωση 6). Ακόμα πιο σοβαρή πηγή προβληματισμού ήταν η «εχθρότητα» της άδειας GPL προς το ιδιόκτητο λογισμικό (δείτε υποσημείωση 9). Μέσα σε αυτό το πλαίσιο λήφθηκε μια απόφαση που θα εξελισσόταν σε καταλύτη νέων εξελίξεων. Η εταιρεία Netscape, επηρεασμένη και από το σύγγραμμα του Raymond, πήρε την απόφαση να αρχίσει να διαθέτει τον περιηγητή της, Netscape Navigator, με τον πηγαίο κώδικά του [11]. Θέλοντας να αρπάξουν την ευκαιρία για να δείξουν ότι το νέο μοντέλο ανάπτυξης λογισμικού μπορεί να είναι προσιτό και σε μεγάλες επιχειρήσεις, μια ομάδα υποστηρικτών του νέου κινήματος, με πρωτεργάτες τους Eric Raymond και Bruce Berens, αποφάσισε να προτείνει έναν νέο ορισμό: **Λογισμικό Ανοικτού Κώδικα** (**Open Source Software**) [12]. Η προσπάθεια το κίνημα να αποκτήσει νέα δυναμική βρήκε άμεσα σημαντικούς υποστηρικτές – συμπεριλαμβανομένου και του Linus Torvalds. (Raymond, The Origins of 'Open Source', 1999)

Ο νέος ορισμός δηλώνει ότι, για να θεωρηθεί ένα λογισμικό ανοικτού κώδικα θα πρέπει να ικανοποιούνται τα κάτωθι κριτήρια:

- *Ελεύθερη Αναδιανομή*: η άδεια χρήσης του δεν πρέπει να περιορίζει οποιοδήποτε άλλο μέρος από την πώληση ή τη δωρεάν διάθεση του λογισμικού ως μέρους μιας διανομής λογισμικού που περιέχει προγράμματα από αρκετές διαφορετικές πηγές. Η άδεια δεν πρέπει να απαιτεί οποιαδήποτε δικαιώματα εκμετάλλευσης ή άλλο αντίτιμο για μια τέτοια πώληση.
- *Πηγαίος Κώδικας*: το πρόγραμμα πρέπει να περιέχει τον πηγαίο κώδικα και πρέπει να επιτρέπει τη διανομή τόσο σε μορφή πηγαίου κώδικα όσο και μορφή μεταγλωττισμένου προγράμματος. Σε περίπτωση που το πρόγραμμα δεν διανέμεται με τον πηγαίο κώδικά του, θα πρέπει να υπάρχει ένας δημοσιευμένος τρόπος απόκτησής του πηγαίου κώδικα για ένα εύλογο κόστος αναπαραγωγής του αντιτύπου· κατά προτίμηση με μεταφόρτωση από το Διαδίκτυο και χωρίς χρέωση. Πηγαίος κώδικας που έχει αποκρυφθεί εσκεμμένα δεν επιτρέπεται. Ενδιάμεσες μορφές όπως η έξοδος ενός προεπεξεργαστή ή μεταφραστή δεν επιτρέπονται.
- *Παράγωγα Έργα*: η άδεια πρέπει να επιτρέπει τροποποιήσεις και παράγωγα έργα, καθώς και τη διανομή τους υπό τους ίδιους όρους με την άδεια χρήσης του αρχικού λογισμικού.
- *Ακεραιότητα του Πηγαίου Κώδικα του Δημιουργού*: η άδεια χρήσης μπορεί να περιορίζει την διανομή του πηγαίου κώδικα σε τροποποιημένη μορφή μόνο αν επιτρέπει τη διανομή αρχείων patch με τον πηγαίο κώδικα με σκοπό την τροποποίηση του προγράμματος κατά τη μεταγλώττισή του. Η άδεια πρέπει να επιτρέπει ρητά τη διανομή του λογισμικού που παράγεται από τροποποιημένο πηγαίο κώδικα. Η άδεια μπορεί να απαιτεί τα παράγωγα έργα να έχουν ένα διαφορετικό όνομα ή αριθμό έκδοσης από το αρχικό λογισμικό.

[11] Μετά την απόφαση αυτή, ο Netscape Navigator μετεξελίχθηκε σε ένα από τα πιο γνωστά έργα ανοικτού κώδικα, το Mozilla Project, που περιλαμβάνει τον περιηγητή Mozilla Firefox και το πρόγραμμα ηλεκτρονικής αλληλογραφίας Mozilla Thunderbird.

[12] Αξίζει να μνημονευθεί πως ήταν η Christine Peterson που εμπνεύστηκε τον όρο. (Bretthauer, 2001)

- *Καμία Διάκριση εις βάρος Προσώπων ή Ομάδων*: η άδεια χρήσης δεν πρέπει να κάνει διακρίσεις εις βάρος οποιουδήποτε προσώπου ή ομάδας προσώπων.
- *Καμία Διάκριση εις βάρος Πεδίων Δραστηριοτήτων*: η άδεια χρήσης δεν πρέπει να περιορίζει οποιονδήποτε από τη χρήση του προγράμματος σε ένα συγκεκριμένο πεδίο δραστηριότητας. Για παράδειγμα, δεν θα πρέπει να περιορίζει τη χρήση του προγράμματος σε μια επιχείρηση ή για έρευνα στη γενετική.
- *Διανομή της Άδειας Χρήσης*: τα δικαιώματα χρήσης του προγράμματος πρέπει να ισχύουν για όλους εκείνους στους οποίους διανέμεται, χωρίς την ανάγκη εκτέλεσης κάποιας επιπρόσθετης άδειας από αυτά τα μέρη.
- *Η Άδεια Χρήσης δεν πρέπει να είναι Συγκεκριμένη για ένα Προϊόν*: τα δικαιώματα χρήσης του προγράμματος δεν πρέπει να εξαρτώνται από το αν το πρόγραμμα είναι μέρος μιας συγκεκριμένης διανομής λογισμικού. Αν το πρόγραμμα εξάγεται από τη αυτή τη διανομή και χρησιμοποιείται ή διανέμεται με βάση τους όρους της άδειας χρήσης του προγράμματος, όλα τα μέρη στα οποία το πρόγραμμα αναδιανέμεται θα πρέπει να έχουν τα ίδια δικαιώματα με αυτά που απονέμονται σε συνδυασμό με την αρχική διανομή λογισμικού.
- *Η Άδεια Χρήσης δεν πρέπει να Περιορίζει άλλο Λογισμικό*: η άδεια χρήσης δεν θα πρέπει να θέτει περιορισμούς σε άλλο λογισμικό που διανέμεται μαζί με το αδειοδοτούμενο λογισμικό. Για παράδειγμα, η άδεια δεν θα πρέπει να επιμένει ότι όλα τα άλλα προγράμματα που διανέμονται με το ίδιο μέσο πρέπει να είναι λογισμικό ανοικτού κώδικα.
- *Η Άδεια Χρήσης πρέπει να είναι Ουδέτερη προς την Τεχνολογία*: καμία διάταξη της άδειας χρήσης δεν μπορεί να βασίζεται σε οποιαδήποτε μεμονωμένη τεχνολογία ή διεπαφή.

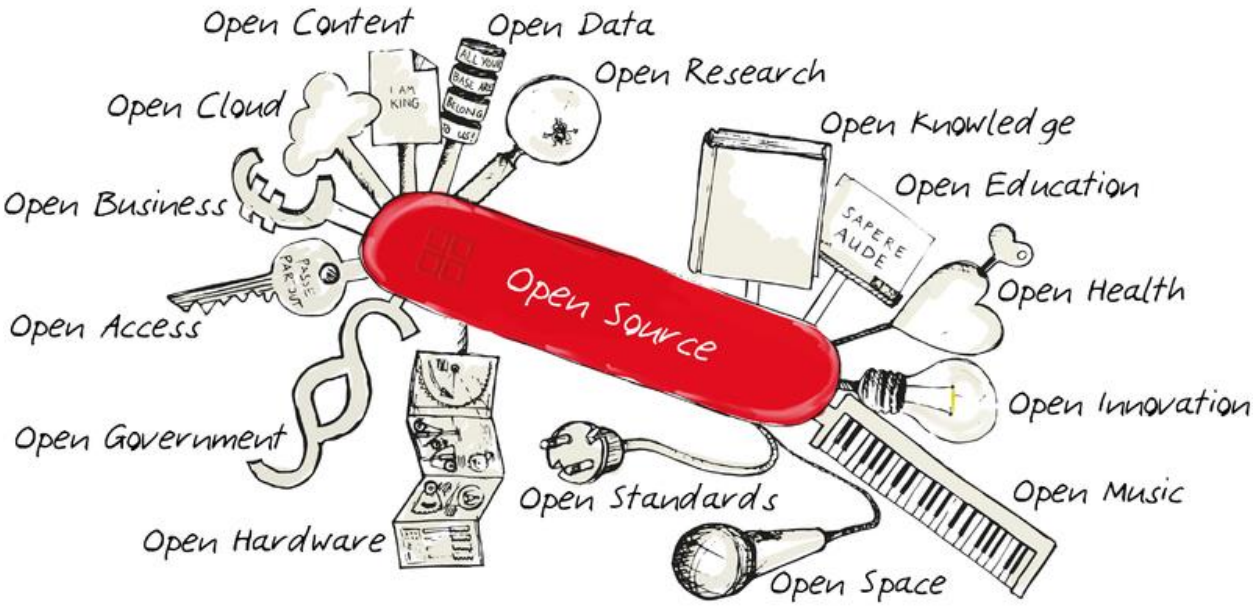
(Open Source Initiative, 2024)

Βασικό χαρακτηριστικό είναι πως ο ορισμός του ανοικτού κώδικα προσπαθεί να συμπεριλάβει περισσότερα έργα που δημιουργήθηκαν έξω από τα πλαίσια του ιδιόκτητου κώδικα [13]. Αυτό στην πράξη γίνεται υποστηρίζοντας πέρα από το ελεύθερο λογισμικό και λογισμικό που επιτρέπει κάποιου είδους ανάμειξη με ιδιόκτητο λογισμικό (όπως για παράδειγμα λογισμικό υπό την άδεια BSD). Αυτός είναι ο λόγος που Richard Stallman και οι υποστηρικτές του ελεύθερου λογισμικού θεωρούν πως ο ανοικτός κώδικας αγνοεί το φιλοσοφικό θέμα της ελευθερίας του λογισμικού. Συνεπώς, μεταξύ του ελεύθερου λογισμικού και του λογισμικού ανοικτού κώδικα δεν υπάρχει απόλυτη ταύτιση [14]· υπάρχει όμως η αντίθεση στο ιδιόκτητο λογισμικό. (Bretthauer, 2001)

[13] Με την εισαγωγή του όρου ανοικτός κώδικας, άρχισε παράλληλα να χρησιμοποιείται και ο όρος **κλειστός κώδικας** για τον ιδιόκτητο κώδικα, για να αναδειχθεί η αντίθεσή τους.

[14] Συχνά χρησιμοποιείται ο όρος **ΕΛ/ΛΑΚ** (ελληνική συντομογραφία των όρων **Ε**λεύθερο **Λ**ογισμικό / **Λ**ογισμικό **Α**νοικτού **Κ**ώδικα – εμπνευσμένο από το παρόμοιο αγγλικό ακρωνύμιο **FOSS** – **F**ree and **O**pen **S**ource **S**oftware) όταν δεν υπάρχει ανάγκη διάκρισης μεταξύ των δύο ειδών.

Ολοκληρώνοντας, αξίζει να σημειωθεί πως η αυτή η φιλοσοφία του ανοικτού κώδικα έχει αποκτήσει κοινωνικές και πολιτικές προεκτάσεις, με πολλούς να υποστηρίζουν την επέκτασή της και σε άλλους τομείς, με έννοιες όπως το *ανοικτό υλικό* (*open source hardware*), το *ανοικτό περιεχόμενο* (*open content*), η *ανοικτή πρόσβαση* (*open access*), η *ανοικτή έρευνα* (*open research*) και πολλές άλλες.



Εικόνα 1: «Ανοικτά κινήματα» που μοιράζονται κοινές αρχές με τον Ανοικτό Κώδικα
[από: Johannes Spielhagen, Bamberg, Germany, CC BY-SA 3.0, μέσω Wikimedia Commons]

3

Ασφάλεια Πληροφοριών

3.1 Βασικές έννοιες

Ο όρος **Ασφάλεια Πληροφοριών** [15] αναφέρεται στην «προστασία από μη εξουσιοδοτημένη [16] πρόσβαση, χρήση, αποκάλυψη, αλλοίωση, τροποποίηση ή ακόμα και καταστροφή με σκοπό την παροχή εμπιστευτικότητας, ακεραιότητας και διαθεσιμότητας» (NIST, χ.χ.). Οι όροι εμπιστευτικότητα, ακεραιότητα και διαθεσιμότητα με τη σειρά τους αναφέρονται ως η «τριάδα CIA» (CIA triad) [17], που αποτελεί το θεμέλιο των ελέγχων ασφάλειας στα Πληροφοριακά Συστήματα (Samonas & Coss, 2014) [18].

Η απαίτηση για **Εμπιστευτικότητα** σε ένα Πληροφοριακό Σύστημα σημαίνει πως πρέπει να υπάρχει προστασία, ώστε οι πληροφορίες να μη διαρρέουν σε μη εξουσιοδοτημένα τρίτα μέρη. Η απαίτηση για **Ακεραιότητα** σημαίνει πως μη εξουσιοδοτημένα τρίτα μέρη δεν θα επιτρέπεται να τροποποιούν τα δεδομένα. Και τέλος, η απαίτηση για **Διαθεσιμότητα** σημαίνει πως, όταν απαιτείται, τα εξουσιοδοτημένα μέρη θα έχουν τη δυνατότητα πρόσβασης στα δεδομένα. (Yee & Zolkipli, 2021)

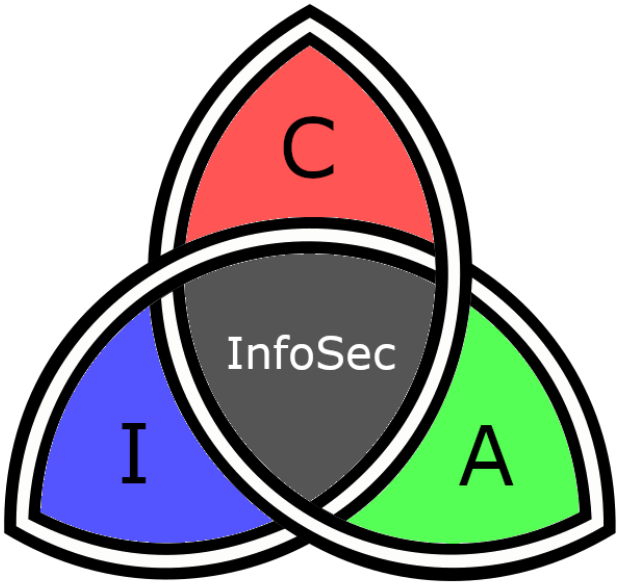
[15] Η Ασφάλεια Πληροφοριών (Information Security) αποδίδεται συχνά με το ακρωνύμιο ***InfoSec***.

[16] Για τον ορισμό της εξουσιοδότησης δείτε την ενότητα «*Μηχανισμοί εξουσιοδότησης*».

[17] **CIA**: αρκτικόλεξο των όρων *Confidentiality* (Εμπιστευτικότητα), *Integrity* (Ακεραιότητα) και *Availability* (Διαθεσιμότητα).

[18] Έχουν προταθεί και πρόσθετες αρχές / απαιτήσεις· με σημαντικότερες την **Αυθεντικοποίηση** (**Authentication**), που σημαίνει πως κάθε οντότητα πρέπει να αποδεικνύει την ταυτότητά της και τη **Μη Αποποίηση** (**Non Repudiation**), που σημαίνει πως κάθε οντότητα δεν θα μπορεί να αρνηθεί την τέλεση των πράξεων που έχει επιτελέσει.

Οι Samonas & Coss (2014) υποστηρίζουν πως στην πραγματικότητα, όλες οι πρόσθετες απαιτήσεις που έχουν προταθεί, ικανοποιούνται είτε από μία από τις αρχές της τριάδας CIA, είτε από κάποιον συνδυασμό τους.



Εικόνα 2: Η τριάδα CIA και η σχέση της με την Ασφάλεια Πληροφοριών

Η προσήλωση στις τρεις βασικές αρχές της Ασφάλειας Πληροφοριών στοχεύει στη μείωση του κινδύνου για το σύστημα. Ο κίνδυνος προέρχεται από **απειλές ασφαλείας** (**security threats**), δηλαδή *«γεγονότα ή καταστάσεις που μπορεί να προκαλέσουν απώλειες σε αγαθά [19], καθώς και τις ανεπιθύμητες επιπτώσεις από τέτοιες απώλειες»* (NIST, 2022). Αυτές οι απειλές ασφάλειας συνδέονται στενά με **ευπάθειες ασφαλείας** (**security vulnerabilities**), δηλαδή *«αδυναμίες στο σύστημα ασφάλειας που μπορεί να έχουν ως αποτέλεσμα την πρόκληση κάποιας απώλειας ή ζημίας»* [Pfleeger στο (Alhazmi, Malaiya, & Ray, 2007)]. Οι ευπάθειες με τη σειρά τους συσχετίζονται άμεσα με **εκμεταλλεύσεις ευπαθειών** (**vulnerability exploits** ή απλά exploits), που *«έχουν σκοπό να προκαλέσουν ένα ρήγμα στην ασφάλεια ενός Πληροφοριακού Συστήματος μέσω μιας ευπάθειας»* (ISO/IEC, 2015).

Ένας άλλος όρος με ιδιαίτερη σημασία για την Ασφάλεια Πληροφοριών είναι ο **παράγοντας απειλών** (**threat agent** ή **threat actor**), καθώς μέσω της δράσης του προκαλούνται απειλές για το σύστημα. Αυτοί οι παράγοντες απειλών μπορούν να διακριθούν σε τρεις κατηγορίες: *ανθρώπινοι*, *τεχνολογικοί* και *περιβαλλοντικοί* (Jouini, Rabai, & Aissa, 2014) [20].

[19] **Αγαθό** (**asset**) είναι κάθε δεδομένο, συσκευή ή άλλο τμήμα των συστημάτων ενός οργανισμού που έχει αξία για τον οργανισμό – συχνά επειδή σχετίζεται με ευαίσθητες πληροφορίες ή μπορεί να χρησιμοποιηθεί για την πρόσβαση σε τέτοιου είδους πληροφορίες. (Irwin, 2020)

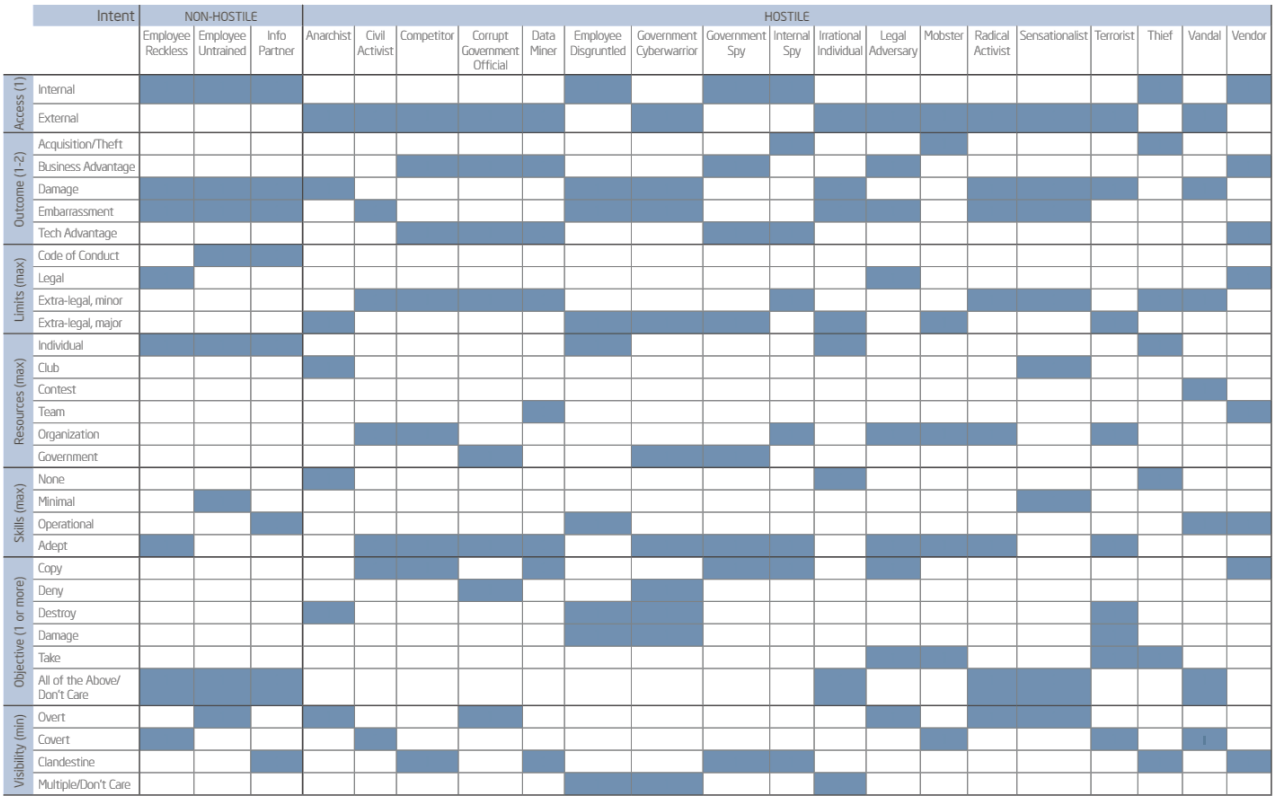
[20] Οι περιβαλλοντικοί παράγοντες απειλών, λ.χ. μια φυσική καταστροφή, καθώς και οι τεχνολογικοί παράγοντες απειλών είναι εν πολλοίς εκτός της εμβέλειας της εργασίας, που θα εστιάσει την προσοχή της στους ανθρώπινους παράγοντες απειλών.

Οι ανθρώπινοι παράγοντες απειλών μπορούν να διακριθούν περαιτέρω με βάση:

- Την *προέλευση* – ο παράγοντας που επιτελεί μια πράξη, προέρχεται από το εσωτερικό ή το εξωτερικό του οργανισμού;
- Το *κίνητρο* – η πράξη που επιτελεί ο παράγοντας, έχει κακόβουλο σκοπό ή όχι;
- Την *πρόθεση* – η πράξη που επιτελεί ο παράγοντας, είναι εσκεμμένη ή είναι αποτέλεσμα ατυχήματος;

(Jouini, Rabai, & Aissa, 2014)

Σε μια παρόμοια προσέγγιση, η Intel ανέπτυξε το Threat Agent Library, όπου με βάση 8 χαρακτηριστικά (όπως η *πρόθεση*, η *πρόσβαση*, οι *δεξιότητες*, οι *διαθέσιμοι πόροι* κ.α.) αναγνωρίζονται 21 αρχέτυπα για τους ανθρώπινους παράγοντες απειλών, με τα αρχέτυπα να περιλαμβάνουν ενδεικτικά: τον *ανεκπαίδευτο υπάλληλο*, τον *ανταγωνιστή*, τον *εσωτερικό κατάσκοπο*, τον *ακτιβιστή*, τον *τρομοκράτη* κ.α. [21] (Intel, 2007)



Εικόνα 3: Διάφοροι παράγοντες απειλής και τα χαρακτηριστικά τους
[πηγή: (Intel, 2007)]

[21] Η παρούσα εργασία δεν θα ασχοληθεί με τις ενέργειες όλων των ανθρώπινων παραγόντων απειλών, αλλά θα εστιάσει σε πράξεις που έχουν κακόβουλα κίνητρα και δεν θα ασχοληθεί με περιπτώσεις ατυχημάτων. Η αναφορά σε αυτούς τους ανθρώπινους παράγοντες απειλών θα γίνεται με τους όρους **κακόβουλος παράγοντας** ή **κακόβουλος χρήστης**.

Οι δράσεις ανθρώπινων παραγόντων που έχουν σκοπό *«να προκαλέσουν ζημία ή έστω να διαταράξουν την ομαλή λειτουργία του συστήματος»* ονομάζονται **επιθέσεις** (**attacks**) [22] (Humayun, Niazi, Jhanjhi, Alshayeb, & Mahmood, 2020). Για να συνδέσουμε τις επιθέσεις με τις προηγούμενες έννοιες, θα λέγαμε με άλλα λόγια πως *«οι επιθέσεις είναι επιτυχημένες εκμεταλλεύσεις των ευπαθειών ασφαλείας του συστήματος»* (Shahriar & Zulkernine, 2012).

Από τον τελευταίο ορισμό, αλλά και τη σχέση των απειλών με τις ευπάθειες που αναφέρθηκε στις προηγούμενες παραγράφους, προκύπτει πως οι επιθέσεις έχουν μεγάλη σχέση με τις απειλές ασφαλείας· όμως δεν ταυτίζονται. Και αυτό γιατί ο όρος απειλή ασφάλειας αναφέρεται σε μια πιθανότητα να συμβεί μια ανεπιθύμητη κατάσταση, ενώ ο όρος επίθεση αναφέρεται σε μια κατάσταση που εξελίσσεται στην πράξη.

Μελετώντας τα άρθρα των Panigrahi (2022) και swetha_vazhakkat (2022), που αναφέρονται στις διαφορές ανάμεσα στις απειλές και τις επιθέσεις, μπορούμε να συνθέσουμε τον παρακάτω ενδεικτικό πίνακα:

ΑΠΕΙΛΗ ΑΣΦΑΛΕΙΑΣ	ΕΠΙΘΕΣΗ
Με πρόθεση ή χωρίς πρόθεση	Με πρόθεση
Κακόβουλη ή μη κακόβουλη	Κακόβουλη
Δυνητικά μπορεί να προκαλέσει ζημία	Η ζημία προκαλείται βάσει σχεδίου
Προκαλείται είτε από φυσικούς, είτε από τεχνολογικούς, είτε από ανθρώπινους παράγοντες απειλών	Προκαλείται από ανθρώπινους παράγοντες απειλών
Σχετικά δύσκολη ανίχνευση [23]	Σχετικά εύκολη ανίχνευση [24]

Πίνακας 1: Χαρακτηριστικές διαφορές απειλών ασφάλειας και επιθέσεων

Αναλύοντας προσεκτικά τις διαφορές, μπορούμε εύλογα να συμπεράνουμε ότι οι επιθέσεις αποτελούν απλά ένα υποσύνολο των απειλών ασφάλειας [25] και έχουν συγκεκριμένα χαρακτηριστικά. Ένα από τα χαρακτηριστικά είναι πως οι επιθέσεις πραγματοποιούνται από ανθρώπινους παράγοντες απειλών που δρουν με πρόθεση και κακόβουλα· οι ανθρώπινοι παράγοντες απειλών που ικανοποιούν τα παραπάνω κριτήρια αποκαλούνται **επιτιθέμενοι** (**attackers**).

[22] Στην παρούσα εργασία οι επιθέσεις που εξετάζονται σχετίζονται με τον κυβερνοχώρο, οπότε θα αναφέρονται και ως **κυβερνοεπιθέσεις** (**cyber-attacks**).

[23] Αν ανιχνεύονταν εύκολα, θα λαμβάνονταν έγκαιρα και τα ανάλογα μέτρα για την εξάλειψή τους.

[24] Εξ ορισμού η επίθεση διαταράσσει την ομαλή λειτουργία του συστήματος, οπότε είναι ένα εύκολα ανιχνεύσιμο γεγονός. Δυστυχώς όμως, η ανίχνευση μιας επίθεσης δεν μπορεί να γίνει πριν την πραγματοποίησή της.

[25] Ένας τρόπος ερμηνείας αυτής της πρότασης είναι ο εξής: κάθε επίθεση είναι και απειλή προς την ασφάλεια, όμως οι απειλές ασφαλείας δεν μπορούν όλες να χαρακτηριστούν ως επιθέσεις.

Επανεξετάζοντας λοιπόν τα αρχέτυπα που παρουσίασε η Intel και αναλύονται στην Εικόνα 3, μπορούμε να πούμε ότι ο ανεκπαίδευτος υπάλληλος παρότι αποτελεί έναν ανθρώπινο παράγοντα απειλών, δεν μπορεί χαρακτηριστεί ως επιτιθέμενος, ενώ ο τρομοκράτης είναι ένας ανθρώπινος παράγοντας απειλών που μπορεί χαρακτηριστεί ως επιτιθέμενος.

Οι επιτιθέμενοι γενικά επιζητούν να εκμεταλλευτούν πιθανές ευπάθειες του συστήματος (NIST, χ.χ.). Αν η επίθεσή τους επιτύχει και καταφέρουν να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση στο σύστημα, τότε ονομάζεται **εισβολή** (**intrusion** ή **penetration**) [Anderson στο (McHugh, 2001)]. Σε αυτό το πλαίσιο ο επιτιθέμενος ονομάζεται και *εισβολέας* (*intruder* ή *penetrator*) [26].

Εικόνα 4: Σχέση απειλών, επιθέσεων και εισβολών

Πέρα από τις περιπτώσεις των εισβολών, υπάρχουν επιθέσεις που δεν απαιτείται η πρόσβαση στο ίδιο το σύστημα [27] · σε κάθε περίπτωση πάντως, οι επιτιθέμενοι, για να επιτύχουν τους κακόβουλους σκοπούς τους χρησιμοποιούν μέσα που ονομάζονται **διανύσματα επίθεσης** (**attack vectors**) (Shacklett, 2021).

[26] Με την αυστηρή ερμηνεία, κάθε εισβολέας είναι και επιτιθέμενος, αλλά κάθε επιτιθέμενος δεν είναι απαραίτητα εισβολέας, αφού η επίθεσή του μπορεί να μην επιτύχει. Η γενικότητα του όρου επιτιθέμενος επιτρέπει τη χρήση του ακόμα και μετά την εισβολή, όπως θα γίνεται και σε αυτή την εργασία.

[27] Χαρακτηριστικό παράδειγμα είναι οι *επιθέσεις DDoS*, που θα εξεταστούν σε επόμενο κεφάλαιο.

Κλείνοντας την ενότητα, αξίζει να αναφερθεί πως οι σοβαρότερες απειλές ασφαλείας θεωρείται πως είναι αυτές που εντάσσονται στην κατηγορία *APT* [28]. Ο όρος χρησιμοποιείται για να περιγράψει τη δραστηριότητα ανθρώπινων παραγόντων απειλών με υψηλή επιδεξιότητα και υψηλή διαθεσιμότητα πόρων, που αξιοποιούν πολλά διαφορετικά διανύσματα επίθεσης, στοχεύοντας, καταρχάς, στην απόκτηση πρόσβασης και την επέκτασή της σε μεγαλύτερο μέρος του Πληροφοριακού Συστήματος. Σε δεύτερο βαθμό στοχεύουν στην εξαγωγή πληροφοριών από το σύστημα ή/και την υπονόμευση της λειτουργίας του. Ιδιαίτερο χαρακτηριστικό μιας απειλής APT είναι η παρουσία της για ένα μεγάλο χρονικό διάστημα, με προσαρμογή στην άμυνα που αναπτύσσει το σύστημα – στόχος, καθώς επίσης και η επιμονή στη διατήρηση της παρουσίας της στο σύστημα για την πραγμάτωση των σκοπών της. (NIST, χ.χ.)

[28] *APT*: αρκτικόλεξο του όρου *Advanced Persistent Threat*, που στα ελληνικά μπορεί να αποδοθεί ως *Προηγμένη Επίμονη Απειλή*. Ο χαρακτηρισμός «προηγμένη» αναφέρεται στις δεξιότητες, τις γνώσεις και τη διαθεσιμότητα των πόρων του επιτιθέμενου. Ο χαρακτηρισμός «επίμονη» αναφέρεται στο μεγάλο χρονικό διάστημα στο οποίο εξελίσσεται η επίθεση, καθώς επίσης και στην επιθυμία να ανακτηθεί μελλοντικά η πρόσβαση ακόμα και αν ανακαλυφθεί και αναχαιτιστεί σε παρόντα χρόνο.

Οι απειλές APT θεωρείται πως στο μεγαλύτερο βαθμό πραγματώνονται υπό τις εντολές κάποιου κράτους, για να τελέσουν ενέργειες κυβερνοκατασκοπίας ή/και κυβερνοπολέμου κατά των εχθρών του.

3.2 Αντίκτυπος των απειλών ασφαλείας

Όπως αναλύθηκε πρωτύτερα, οι απειλές ασφαλείας είναι μια γενική κατηγορία που εμπεριέχει τις επιθέσεις και τις εισβολές· οπότε μεγέθη που χρησιμοποιούνται για την μελέτη των απειλών έχουν εφαρμογή και για τη μελέτη των άλλων δύο κατηγοριών [29]. Μια πολύ σημαντική πληροφορία για κάθε οργανισμό είναι ο *αντίκτυπος* (*impact*) [30] κάθε απειλής ασφαλείας, δηλαδή το μέγεθος της ζημίας που θα πρέπει να αναμένεται ως αποτέλεσμα της απειλής αυτής (NIST, χ.χ.).

Οι Vidalis & Jones (2005) μελετώντας τον αντίκτυπο μιας απειλής ασφαλείας, παρουσιάζουν μια κλίμακα βαθμονόμησης του αντικτύπου με βάση την επίδραση της ζημίας στη λειτουργία του οργανισμού και περιλαμβάνει τις εξής περιπτώσεις:

- *Ασήμαντος*: μη εξουσιοδοτημένη χρήση ενός αγαθού, χωρίς όμως να προκαλείται κάποια ουσιαστική απώλεια ή να επηρεάζεται με άλλον τρόπο ο οργανισμός.
- *Μικρός*: μικρή απώλεια αγαθών, που όμως δεν επηρεάζει την επιχειρησιακή λειτουργία του οργανισμού.
- *Μέτριος*: η επιχειρησιακή λειτουργία του οργανισμού διαταράσσεται και απαιτούνται αλλαγές στον τρόπο διεξαγωγής των εργασιών.
- *Σημαντικός*: η επιχειρησιακή λειτουργία διακόπτεται πλήρως, αν δεν ληφθούν αμέσως αντίμετρα.
- *Καταστροφικός*: η επιχειρησιακή λειτουργία διακόπτεται πλήρως από τη στιγμή που πραγματοποιείται η απειλή.

Οι Simmons, Ellis, Shiva, Dasgupta, & Wu (2009) ακολουθώντας μια άλλη λογική και όχι αυτή της βαθμονόμησης, επικέντρωσαν την προσοχή τους στις πληροφορίες και αναφέρουν τις εξής περιπτώσεις [31]:

- *Disclosure*, δηλαδή *αποκάλυψη* μιας νέας οπτικής για τις πληροφορίες που δεν είχε υπό κανονικές συνθήκες ο επιτιθέμενος, ανοίγοντας μια νέα πιθανότητα εκμετάλλευσης.
- *Discovery*, δηλαδή *ανακάλυψη* πληροφοριών που πρότινος δεν ήταν γνωστές στον επιτιθέμενο. Π.χ. με χρήση εργαλείου που σαρώνει το σύστημα για συγκέντρωση πληροφοριών που θα χρησιμοποιηθούν σε μια επικείμενη επίθεση κατά κάποιου συγκεκριμένου στόχου.
- *Disrupt*, δηλαδή *διακοπή* της πρόσβασης σε υπάρχουσες πληροφορίες. Τέτοιο παράδειγμα αποτελεί η περίπτωση *DoS* (*Denial of Service – Άρνηση Υπηρεσίας*).
- *Distort*, δηλαδή *διατάραξη* των πληροφοριών, συνήθως μέσω της τροποποίησης δεδομένων σε κάποιο αρχείο ή των πληροφοριών που παρέχει το θύμα.
- *Destruct*, δηλαδή *καταστροφή* πληροφοριών με διαγραφή αρχείων ή αφαίρεση πληροφοριών από το θύμα. Θεωρείται η πιο σοβαρή περίπτωση αντικτύπου της κλίμακας αυτής.

[29] Ωστόσο, η παρούσα εργασία δεν ασχολείται με όλες τις απειλές ασφαλείας, αλλά εστιάζει σε συγκεκριμένα είδη επιθέσεων και εισβολών που προέρχονται από τον κυβερνοχώρο.

[30] Ο όρος *impact* αποδίδεται στα ελληνικά και ως *επίπτωση*.

[31] Αναγράφονται με τους πρωτότυπους αγγλικούς όρους, καθώς η συγκριμένη πρόταση χρησιμοποιεί όρους που έχουν πρώτο γράμμα το *D* δημιουργώντας έναν μνημονικό κανόνα των *πέντε D*.

Παρόμοια, οι Jouini, Rabai & Aissa (2014) κατηγοριοποίησαν τον αντίκτυπο των απειλών ασφάλειας εστιάζοντας στο είδος της ζημίας που προκαλείται στα αγαθά και αναγνώρισαν τις εξής περιπτώσεις:

- *Καταστροφή πληροφοριών* (ή γενικά πόρων του συστήματος), που οδηγεί σε διακοπή της ορθής λειτουργίας του συστήματος.
- *Αλλοίωση δεδομένων*, που ακολούθως μπορεί να οδηγήσει το σύστημα σε εξαγωγή παραπλανητικών ή ψευδών πληροφοριών [32].
- *Αποκάλυψη πληροφοριών* σε μη εξουσιοδοτημένα τρίτα μέρη [33].
- *Κλοπή υπηρεσίας*, δηλαδή μη εξουσιοδοτημένη χρήση πόρων ή υπηρεσιών του Πληροφοριακού Συστήματος [34].
- *Παράνομη χρήση*, που αφορά τη χρήση πόρων ή υπηρεσιών του Πληροφοριακού Συστήματος για σκοπούς άσχετους προς τη λειτουργία του [35].
- *Άρνηση υπηρεσίας*, που σημαίνει πως πόροι ή υπηρεσίες του συστήματος θα πάψουν να είναι διαθέσιμοι στους νόμιμους χρήστες του Πληροφοριακού Συστήματος.
- *Κλιμάκωση δικαιωμάτων χρήστη*, που σημαίνει πως μια οντότητα θα αποκτήσει δικαιώματα χρήσης στο Πληροφοριακό Σύστημα που δεν θα έπρεπε να έχει υπό φυσιολογικές συνθήκες.

[32] Βασική αρχή της επιστήμης της Πληροφορικής είναι πως οι πληροφορίες είναι το αποτέλεσμα της επεξεργασίας των δεδομένων. Είναι λογικό λοιπόν, όταν τα δεδομένα είναι αλλοιωμένα, τότε και οι πληροφορίες που σχετίζονται με αυτά να είναι τουλάχιστον παραπλανητικές.

[33] Η περίπτωση αυτή αποκτά ιδιαίτερη σημασία αν αναλογιστούμε ότι η αποκάλυψη μπορεί να αφορά πληροφορίες που εμπίπτουν στο στρατιωτικό ή κρατικό απόρρητο, σε εμπορικά μυστικά ή σε προσωπικά δεδομένα. Η πρωτογενής βλάβη αφορά την άρση της μυστικότητας των πληροφοριών. Σε δευτερογενές επίπεδο βλάπτεται η φήμη του οργανισμού για την ανικανότητα αποτροπής της διαρροής. Σε τριτογενές επίπεδο υπάρχει πάντα η πιθανότητα ο οργανισμός να βρεθεί νομικά υπόλογος για την αποκάλυψη – λ.χ. όταν αυτή αφορά προσωπικά δεδομένα τρίτων μερών.

[34] Εδώ πρέπει να διευκρινιστεί πως εστιάζουμε στο δικαίωμα πρόσβασης – η χρήση των πόρων ή των υπηρεσιών είναι ορθή (η προβλεπόμενη από το Πληροφοριακό Σύστημα), όμως αξιοποιείται από μια οντότητα που δεν θα έπρεπε να έχει πρόσβαση σε αυτούς.

[35] Εδώ εστιάζουμε στην ορθή χρήση – πιθανόν να υπάρχει δικαίωμα πρόσβασης, όμως η χρήση δεν είναι η προβλεπόμενη από το Πληροφοριακό Σύστημα.

4

Κυβερνοεπιθέσεις

4.1 Διανύσματα επίθεσης

Στον πυρήνα της παρούσας εργασίας βρίσκονται οι επιθέσεις και οι εισβολές, ως απειλές ασφάλειας που χαρακτηρίζονται από κακόβουλα κίνητρα και εξελίσσονται βάσει σχεδίου, με απώτερο στόχο την επίτευξη ενός αποτελέσματος που κρίνεται ζημιογόνο για το υπολογιστικό σύστημα. Σε αυτό το κεφάλαιο θα εξεταστούν αμιγώς εξωγενείς επιθέσεις, αλλά και σύνθετες επιθέσεις που είναι μεν εξωγενείς στην προέλευσή τους, απαιτούν δε και την εκούσια ή την ακούσια δράση εσωτερικών παραγόντων. Για να καταφέρουν να εκτελέσουν τις επιθέσεις, οι κακόβουλοι παράγοντες χρειάζεται να αξιοποιήσουν μια σειρά από μέσα, εργαλεία λογισμικού, τεχνικές και μεθόδους, που συνολικά ονομάζονται διανύσματα επίθεσης. Επειδή το πλήθος των διανυσμάτων επίθεσης είναι πολύ μεγάλο, στην παρούσα εργασία θα αναλυθούν μόνο στο επίπεδο των τριών κυριότερων κατηγοριών τους: την αξιοποίηση *κοινωνικής μηχανικής*, την *εκμετάλλευση ευπαθειών* του λογισμικού και τη χρήση *κακόβολου λογισμικού* [36].

[36] Όπως θα φανεί στη σχετική ενότητα, και το κακόβουλο λογισμικό διαδίδεται είτε με αξιοποίηση της κοινωνικής μηχανικής, είτε με εκμετάλλευση ευπαθειών. Ωστόσο, κάποια είδη κακόβουλου λογισμικού χαρακτηρίζονται από έναν βαθμό αυτονομίας· για τον λόγο αυτό εξετάζεται ως ξεχωριστό διάνυσμα.

4.1.1 Κοινωνική Μηχανική

Ο όρος **κοινωνική μηχανική** (**social engineering**) «*αναφέρεται σε όλες τις τεχνικές που έχουν σκοπό να πείσουν έναν στόχο να αποκαλύψει συγκεκριμένες πληροφορίες ή να επιτελέσει συγκεκριμένες πράξεις για παράνομους λόγους*» (ENISA, 2023).

Στον τομέα της Ασφάλειας πιο συγκεκριμένα, εφαρμόζεται πρωτίστως με τη τεχνική **phishing**, στην κλασική μορφή της οποίας, ένας επιτιθέμενος αποστέλλει ένα παραπλανητικό email, που φαίνεται πως προέρχεται από κάποια έμπιστη πηγή, με σκοπό να εξαπατήσει τους παραλήπτες ώστε να αποκαλύψουν εμπιστευτικές πληροφορίες (Yuchong & Qinghui, 2021) ή να εγκαταστήσουν, εν αγνοία τους, κακόβουλο λογισμικό (Biju, Gopal, & Prakash, 2019).

Μια άλλη τεχνική, που συχνά χρησιμοποιείται σε συνδυασμό με το phishing, είναι το **website spoofing**, δηλαδή η δημιουργία ενός αντιγράφου ενός έμπιστου ιστότοπου που έχει σκοπό να εξαπατήσει τους χρήστες ώστε να εισάγουν ευαίσθητα δεδομένα (τα οποία θα γίνουν γνωστά στον κακόβουλο διαχειριστή του ιστότοπου) ή να μεταφορτώσουν και να εγκαταστήσουν, εν αγνοία τους, κακόβουλο λογισμικό στο υπολογιστικό σύστημα που χρησιμοποιούν. (Lai, Goh, Yap, & Ng, 2023)

Οι Oppliger & Gajek (2005) αναφέρουν ότι υπάρχουν 5 διακριτά επίπεδα επιθέσεων που εφαρμόζουν αυτή τη μέθοδο· ξεκινούν από την απλούστερη μορφή phishing και φτάνουν μέχρι τις μορφές που το phishing συνδυάζεται με το website spoofing, με κάθε επίπεδο να έχει μεγαλύτερη πολυπλοκότητα σε σχέση με το προηγούμενο, ενώ παράλληλα σε κάθε επίπεδο αυξάνεται και η δυσκολία αναγνώρισης της απάτης από το θύμα.

Γενικά, μπορούν να αναγνωριστούν αρκετές τεχνικές εφαρμογής της κοινωνικής μηχανικής σε επιθέσεις, όμως η αναφορά σε όλες είναι πέρα από τα όρια της παρούσας εργασίας. Ένα σημείο που πρέπει να σημειωθεί παρόλα αυτά, είναι πως η κοινωνική μηχανική αποτελεί βασικό κομμάτι του σχεδιασμού κάποιων ειδών κακόβουλου λογισμικού, αφού, όπως θα εξηγηθεί και σε επόμενη ενότητα, αποτελεί αναπόσπαστο κομμάτι του μηχανισμού διάδοσής τους.

4.1.2 Εκμετάλλευση ευπαθειών λογισμικού

Καθώς το σύγχρονο λογισμικό γίνεται πιο ογκώδες σε κώδικα, γίνεται και πιο πολύπλοκο. Όσο όμως αυξάνεται η πολυπλοκότητά του, τόσο πιο πιθανό είναι να περιέχει *σφάλματα* (ή *λάθη*) στον σχεδιασμό ή στην υλοποίησή του· υπάρχουν διαδικασίες για τον έλεγχο, την ανίχνευση και τη διόρθωσή τους, όμως πάντα υπάρχει η πιθανότητα κάποια σφάλματα να παραμείνουν ως *ελαττώματα* στο τελικό προϊόν. (Shahriar & Zulkernine, 2012)

Τα ελαττώματα αυτά μπορεί δυνητικά να αποτελούν ευπάθειες ασφαλείας· γιατί, όπως ειπώθηκε και σε προηγούμενη ενότητα, αν ένας κακόβουλος παράγοντας καταφέρει αρχικά να ανιχνεύσει μια ευπάθεια (στην προκειμένη περίπτωση το ελάττωμα στο λογισμικό) και στη συνέχεια καταφέρει να αναπτύξει και έναν μηχανισμό για το εκμεταλλευτεί (το **exploit** [37]), τότε μπορεί να πραγματοποιήσει μια επιτυχημένη επίθεση.

Πιο συγκεκριμένα, στην τεχνική **Code Injection** [38], συνήθης στόχος του exploit είναι τα πεδία εισαγωγής δεδομένων από τους χρήστες μιας εφαρμογής [39] · αν ένας κακόβουλος χρήστης εντοπίσει ότι δεν γίνεται σωστός έλεγχος εγκυρότητας των δεδομένων (αυτό είναι το *ελάττωμα – ευπάθεια*), τότε μπορεί να επιχειρήσει να εισάγει στα πεδία μη έγκυρες τιμές δεδομένων, που θα φέρουν σε «σύγχυση» την εφαρμογή (αυτό είναι το *exploit*), μαζί με κώδικα, που θα του δώσει μη εξουσιοδοτημένη πρόσβαση σε πόρους της εφαρμογής (λ.χ. δεδομένα από μια Βάση Δεδομένων – αυτό είναι η *επίθεση – εισβολή*). Παρόμοια, στην τεχνική **Arbitrary Code Execution** (**ACE**) [40], γίνεται code injection στα δεδομένα εισόδου μιας διεργασίας, με στόχο να αλλάξει η ροή εκτέλεσης του προγράμματος και να εκτελεστεί ο κακόβουλος κώδικας που θα δώσει στον κακόβουλο χρήστη μη εξουσιοδοτημένη πρόσβαση στο σύστημα (ως απλός χρήστης). Όταν η τεχνική αυτή είναι δυνατόν να εκτελεστεί απομακρυσμένα, τότε λέγεται **Remote Code Execution** (**RCE**) [41]. Η τελευταία περίπτωση είναι από τις πιο σοβαρές απειλές ασφάλειας, καθώς, αν ο κακόβουλος παράγοντας καταφέρει, αρχικά να αποκτήσει πρόσβαση ως απλός χρήστης, και στη συνέχεια να εκμεταλλευτεί μια άλλη ευπάθεια για να επιτύχει **κλιμάκωση προνομίων** [42], τότε θα έχει αποκτήσει απομακρυσμένα τον απόλυτο έλεγχο ολόκληρου του συστήματος. (Zheng & Zhang, 2013; Sommestad, Holm, & Ekstedt, 2012)

---

[37] **Exploit** (**εκμετάλλευση**): ένα πρόγραμμα που υλοποιείται ή μια μέθοδος που εφαρμόζεται για την εκμετάλλευση μιας ευπάθειας και την πραγματοποίηση μιας επίθεσης. (Shahriar & Zulkernine, 2012)

[38] *Code Injection*: στα ελληνικά αποδίδεται ως *Έγχυση Κώδικα*.

[39] Εφαρμόζεται κατά κόρον σε εφαρμογές που εκτελούνται ως σενάρια εντολών (scripts) μιας διερμηνευόμενης γλώσσας (λ.χ. HTML, JavaScript, PHP, SQL κ.α.). Εφαρμόζεται επίσης και σε περιβάλλοντα CLI (Command Line Interface – λ.χ. Linux Shell, Windows PowerShell και Command Line), καθώς και σε αυτά εκτελούνται διερμηνευόμενα σενάρια εντολών.

[40] Arbitrary Code Execution: στα ελληνικά αποδίδεται ως Εκτέλεση Αυθαίρετου Κώδικα.

[41] Remote Code Execution: στα ελληνικά αποδίδεται ως Απομακρυσμένη Εκτέλεση Κώδικα.

[42] **Κλιμάκωση προνομίων** (**privilege escalation**): όταν απλοί χρήστες καταφέρνουν να εκτελούν προγράμματα με δικαιώματα υπερχρήστη, δηλαδή ενός χρήστη με απεριόριστα προνόμια ανάγνωσης και εγγραφής σε όλες τις περιοχές του συστήματος αρχείων.

Όπως είναι λογικό, οι κατασκευαστές λογισμικού ενδιαφέρονται για την ανίχνευση τέτοιων ελαττωμάτων – ευπαθειών και μόλις ανιχνευτούν ή τους γνωστοποιηθούν αρχίζει μια διαδικασία επιδιόρθωσης μέσω *ενημερώσεων ασφαλείας* (*security updates* ή *patches*). (Shahriar & Zulkernine, 2012; Simmons, Ellis, Shiva, Dasgupta, & Wu, 2009)

Εξαιτίας αυτής της διαδικασίας διαρκούς «εξουδετέρωσης» των ευπαθειών στο λογισμικό, υπάρχει μια κατηγορία ευπαθειών με τεράστια σημασία. Είναι οι λεγόμενες **ευπάθειες zero-day** (ή *0-day*), δηλαδή οι ευπάθειες που είναι άγνωστες στον ίδιο τον κατασκευαστή του λογισμικού. Ως ημέρα μηδέν (zero day) θεωρείται η μέρα που η ευπάθεια γίνεται γνωστή στον κατασκευαστή. Το πρόβλημα που προκύπτει είναι μήπως κάποιος κακόβουλος παράγοντας κατάφερε να ανιχνεύσει, να δημιουργήσει και να χρησιμοποιήσει ένα *zero-day exploit* στο διάστημα που η ευπάθεια υπήρχε χωρίς να τη γνωρίζει ο κατασκευαστής του λογισμικού· διότι δεν υπάρχει άμυνα σε μια επίθεση που εκμεταλλεύεται ένα τέτοιο exploit [43]. (Lee, 2023) Το μόνο θετικό είναι ότι μετά την πραγματοποίηση μιας επίθεσης, ανοίγει ο δρόμος για την ανίχνευση και την επιδιόρθωσή της ευπάθειας zero-day. (Groš, 2021)

[43] Για τον λόγο αυτό, οι ευπάθειες zero-day ή/και τα exploits τους κοστολογούνται στη «μαύρη αγορά» στην κλίμακα των εκατομμυρίων δολαρίων. (Groš, 2021)

4.1.3 Κακόβουλο λογισμικό

Ο όρος **κακόβουλο λογισμικό** προκύπτει από μετάφραση του αντίστοιχου αγγλικού όρου *malicious software*, που συχνότερα αναφέρεται με το ακρωνύμιό του, **malware**. Κακόβουλο ονομάζεται κάθε λογισμικό που σχεδιάζεται με πρόθεση να προκαλέσει ζημία ή βλάβη σε ένα υπολογιστικό σύστημα ή τους χρήστες αυτού (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020). Τυπικά, εγκαθίστανται χωρίς να λαμβάνουν γνώση και να συναινούν οι χρήστες του συστήματος [44] (Biju, Gopal, & Prakash, 2019).

Δημιουργείται και χρησιμοποιείται από κακόβουλα άτομα ή ομάδες που θέλουν να εισβάλουν σε ένα σύστημα με σκοπό την υποκλοπή προσωπικών ή εμπιστευτικών δεδομένων, τη μη εξουσιοδοτημένη χρήση πόρων του συστήματος, τη διάπραξη εκβιασμών κ.α. Πέρα από τους κυβερνοεγκληματίες, που στις περισσότερες περιπτώσεις έχουν ως κίνητρο την αποκόμιση οικονομικού οφέλους (Groš, 2021), έχει αξιοποιηθεί από εταιρείες πρόθυμες να εφαρμόσουν πρακτικές αμφισβητούμενης ηθικής για να προστατέψουν τα συμφέροντά τους (λ.χ. *Sony BMG rootkit* [45]) (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020) ή ακόμα και κράτη στα πλαίσια του κυβερνοπολέμου κατά άλλων κρατών (λ.χ. *Stuxnet worm* [46]) (Lee, 2023).

[44] Υπάρχει η περίπτωση το κακόβουλο λογισμικό να εγκατασταθεί με πρόθεση από έναν χρήστη – εσωτερικό επιτιθέμενο, αλλά αυτή η περίπτωση δεν είναι σε καμία περίπτωση ο κανόνας.

[45] Το 2005 η εταιρεία Sony BMG ενσωμάτωσε κρυφά ένα rootkit στα CD μουσικής της, ως μέτρο προστασίας της πνευματικής ιδιοκτησίας της. Αν ένας αγοραστής προσπαθούσε να αντιγράψει παράνομα το CD στον Η/Υ του, το λογισμικό θα τίθετο σε λειτουργία και θα το απέτρεπε. Ωστόσο, η παρουσία του λογισμικού στον Η/Υ του χρήστη δημιουργούσε ευπάθειες, που εκμεταλλεύονταν άλλα κακόβουλα προγράμματα. Η αποκάλυψη του λογισμικού στα CD οδήγησε σε ομαδικές δικαστικές προσφυγές κατά της εταιρείας. (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020)

[46] Το *Stuxnet* είναι από τα πιο χαρακτηριστικά παραδείγματα απειλής APT. Αποκαλύφθηκε το 2010 και θεωρείται πως σχεδιάστηκε από τις ΗΠΑ και το Ισραήλ για να σαμποτάρει το πυρηνικό πρόγραμμα του Ιράν. Πρόκειται για ένα εξαιρετικά πολύπλοκο λογισμικό που αρχικά προσέβαλε Η/Υ των πυρηνικών εγκαταστάσεων του Ιράν μέσω USB. Στη συνέχεια εκμεταλλευόμενο ευπάθειες zero-day άρχισε να εξαπλώνεται στο δίκτυο όπως ένα worm. Αφού εγκαθίστατο σε έναν Η/Υ εκτελούσε τυπικές λειτουργίες ενός rootkit για την απόκρυψη της παρουσίας του και τη διασφάλιση ότι θα επανεκτελεστεί μετά από μια επανεκκίνηση. Επιπλέον, όπως ένα bot συνδεόταν σε ένα C&C δίνοντας τη δυνατότητα στους χειριστές του να το αναβαθμίσουν απομακρυσμένα. Το λογισμικό εξαπλωνόταν με απώτερο σκοπό να προσβάλει υπολογιστές ειδικού σκοπού, τους ελεγκτές, οι οποίοι δεν ήταν συνδεδεμένοι στο Διαδίκτυο. Το Stuxnet επενέβαινε στην εκτέλεση συγκεκριμένων προγραμμάτων που εκτελούνταν στους ελεγκτές και ήλεγχαν τη λειτουργία των φυγοκεντρωτών ουρανίου. Εξαιτίας του, τα προγράμματα αυτά έστελναν αναφορές για εύρυθμη απόδοση στο λογισμικό παρακολούθησης, ενώ την ίδια στιγμή έβαζαν τους φυγοκεντρωτές να λειτουργούν εκτός των φυσιολογικών ορίων. Έτσι, κατάφερε να καταστρέψει περίπου 1.000 φυγοκεντρωτές ουρανίου χωρίς να σημάνει συναγερμός. (Lee, 2023)

Από τεχνικής απόψεως, το κακόβουλο λογισμικό σχεδιάζεται με δύο λογικά μέρη:

- Το πρώτο είναι ο μηχανισμός διάδοσης, που επικεντρώνεται στον τρόπο με τον οποίο το κακόβουλο λογισμικό θα εισβάλει στο σύστημα. Κάποια είδη, όπως οι ιοί και τα σκουλήκια, αναπαράγουν αντίγραφα του εαυτού τους για να περάσουν σε όσο περισσότερα συστήματα μπορούν. Άλλα, όπως οι δούρειοι ίπποι, βασίζονται στην κοινωνική μηχανική, ώστε κάποιος χρήστης να προκαλέσει, χωρίς να το γνωρίζει, την εγκατάστασή τους στο σύστημα.
- Το δεύτερο είναι το λεγόμενο «*φορτίο*» (*payload*), που ασχολείται με την εκτέλεση της καθαυτού κακόβουλης λειτουργίας του λογισμικού. Η εκτέλεση αυτού του φορτίου είναι που επιφέρει τον μεγαλύτερο αντίκτυπο [47] στο υπολογιστικό σύστημα· ο αντίκτυπος αυτός αξιολογείται με τις μεθόδους που αναλύθηκαν στην ενότητα «*Αντίκτυπος των απειλών ασφαλείας*».

Υπάρχουν πολλά διαφορετικά είδη κακόβουλου λογισμικού, ωστόσο με την εξέλιξή τους τα όρια μεταξύ του ενός είδους από το άλλο γίνονται πιο θολά. Στις επόμενες ενότητες αναλύεται ο τρόπος λειτουργίας των πιο σημαντικών από αυτά.

[47] Πέρα από το φορτίο, αντίκτυπο στο σύστημα έχουν και οι μηχανισμοί διάδοσης του κακόβουλου λογισμικού. Για παράδειγμα, ο μηχανισμός εξάπλωσης ενός ιού έχει ως αποτέλεσμα την κατανάλωση μέρους του αποθηκευτικού χώρου του συστήματος. Παρόμοια, ο μηχανισμός διάδοσης ενός σκουληκιού έχει ως αποτέλεσμα τη μείωση του εύρους ζώνης (bandwidth) του δικτύου.

4.1.3.1 Ιός

Ένας **ιός** (**virus**) είναι αυτό-αναπαραγόμενος κώδικας που εξαπλώνεται προσαρτώντας αντίγραφα του εαυτού του σε αρχεία ή boot sectors (The Virus Encyclopedia, 2023) [48, 49]. Ο τρόπος εξάπλωσής του μοιάζει με αυτόν ενός βιολογικού ιού που αναπαράγεται στα κύτταρα του ξενιστή του (Yuchong & Qinghui, 2021)· κατ' αναλογία με έναν βιολογικό ιό, η διαδικασία εξάπλωσής του ονομάζεται *μόλυνση* (*infection*) και το αρχείο που μολύνεται ονομάζεται *ξενιστής* (*host*). Και όπως ένας βιολογικός ιός εξαπλώνεται από οργανισμό σε οργανισμό, έτσι και αυτό το κακόβουλο λογισμικό εξαπλώνεται από υπολογιστικό σύστημα σε υπολογιστικό σύστημα χάρη στα μολυσμένα αρχεία.

Ένας ιός δεν μολύνει οποιοδήποτε αρχείο, αλλά μόνο όσα προσφέρουν την ευκαιρία εκτέλεσης του κώδικά του. Γενικά, οι ιοί μπορούν να διακριθούν σε:

- *Ιούς boot sector*, που έχουν την ικανότητα να μολύνουν τους boot sectors μέσων αποθήκευσης, όπως δισκέτες, σκληροί δίσκοι και οπτικοί δίσκοι. Αυτοί οι ιοί εκτελούνται πριν από τη φόρτωση του λειτουργικού συστήματος και μπορούν να έχουν τα περισσότερα προνόμια εκτέλεσης. (WhatIs.com, 2023)
- *Ιούς αρχείων*, που μολύνουν εκτελέσιμα προγράμματα (λ.χ. .exe) ή αρχεία με σενάρια εντολών (scripts) του λειτουργικού συστήματος (λ.χ. .com, .sys, .inf κ.α.) [50]. (WhatIs.com, 2023)
- *Ιούς macro*, που μολύνουν έγγραφα εφαρμογών στις οποίες υπάρχει η δυνατότητα εκτέλεσης μακροεντολών – τυπικά, έγγραφα του MS Office. Σε αντίθεση με τα άλλα είδη ιών που περιορίζονται σε ένα συγκεκριμένο λειτουργικό σύστημα, αυτοί οι ιοί μπορούν να επηρεάσουν όλα τα λειτουργικά συστήματα. (WhatIs.com, 2023)

Ως προς τη διάδοσή τους, οι πρώτοι ιοί εξαπλώνονταν με τη χρήση φορητών μέσων αποθήκευσης (π.χ. δισκετών, μονάδων USB flash), ενώ πλέον εξαπλώνονται κατά βάση μέσω υπηρεσιών του Διαδικτύου (π.χ. με επισύναψη μολυσμένων αρχείων σε email ή μέσω κάποιας εφαρμογής διαμοιρασμού αρχείων). (McHugh, 2001)

Ένα θέμα στο οποίο αξίζει να γίνει ιδιαίτερη αναφορά είναι οι τεχνικές που αναπτύχθηκαν για την απόκρυψη της μόλυνσης ενός αρχείου από έναν ιό, καθώς οι ιδέες αυτές εφαρμόστηκαν αργότερα και σε άλλες κατηγορίες κακόβουλου λογισμικού.

[48] Ιστορικά, ο John von Neumann ήταν ο πρώτος που περιέγραψε πως ένα πρόγραμμα θα μπορούσε να αναπαράγει τον εαυτό του, στις διαλέξεις του με τίτλο «*Θεωρία των αυτο-αναπαραγόμενων αυτόματων*» (*Theory of self-reproducing automata, 1949)*. (The Virus Encyclopedia, 2023)

[49] Ο όρος «ιός» για προγράμματα με την ικανότητα αυτο-αναπαραγωγής καθιερώθηκε από το έργο του Fred Cohen «*Ιοί Υπολογιστών – Θεωρία και Πειράματα*» (*Computer Viruses – Theory and Experiments, 1984*). (The Virus Encyclopedia, 2023)

[50] Αξίζει να σημειωθεί ότι οι ιοί που εξαπλώνονται μέσω της αυτόματης εκτέλεσης των μονάδων USB flash δεν ανήκουν στην κατηγορία των ιών boot sector, αλλά στην κατηγορία των ιών αρχείων, γιατί μολύνουν το αρχείο autorun.inf.

Οι Kamal, Ali, Alani, & Abdulmajed (2016) εξηγούν πως τα πρώτα αντι-ιικά προγράμματα αναζητούσαν στα αρχεία μια συγκεκριμένη ακολουθία κώδικα [51], που είχαν ταυτίσει με έναν ιό. Για αυτό τον λόγο, όπως αναφέρουν, η πρώτη μέθοδος που εφάρμοσαν οι δημιουργοί των ιών την απόκρυψή τους ήταν η κρυπτογράφηση του κώδικά τους. Πιο συγκεκριμένα, ένας *κρυπτογραφημένος ιός* (*encrypted virus*) αποτελείται από δύο τμήματα: το τμήμα του κρυπτογραφημένου κυρίως κώδικά [52] του και το τμήμα αποκρυπτογράφησης (*decryptor*). Όταν εκτελείται ένα μολυσμένο αρχείο, πρώτα παίρνει τον έλεγχο το τμήμα αποκρυπτογράφησης, που αποκρυπτογραφεί τον κυρίως κώδικα του ιού και στη συνέχεια ο κυρίως κώδικας εκτελείται μολύνοντας νέα αρχεία, κρυπτογραφώντας όμως τον εαυτό του με διαφορετικά κλειδιά κρυπτογράφησης κάθε φορά. Το αποτέλεσμα είναι πως το κρυπτογραφημένο τμήμα του ιού διαφέρει σε κάθε αρχείο. Η αδυναμία αυτής της τεχνικής είναι πως ο ιός μπορεί να αναγνωριστεί από τον κώδικα του τμήματος αποκρυπτογράφησης.

Συνεχίζοντας, οι συγγραφείς αναφέρονται σε μια εξέλιξη της προηγούμενης μεθόδου, που ονομάζεται **πολυμορφισμός** (**polymorphism**). Ένας *πολυμορφικός ιός* χωρίζεται σε τρία λογικά συστατικά μέρη: το τμήμα του κυρίως κώδικα, το τμήμα αποκρυπτογράφησης και ένα τρίτο τμήμα, που λέγεται *μηχανή μετάλλαξης* (*mutation engine*). Ο κυρίως κώδικας του ιού και η μηχανή μετάλλαξης παραδίδονται κρυπτογραφημένα και, στη φάση της εκτέλεσης, το τμήμα της αποκρυπτογράφησης τα αποκρυπτογραφεί. Στη φάση της μόλυνσης νέων αρχείων, η μηχανή μετάλλαξης τροποποιεί και τον κυρίως κώδικα και το τμήμα αποκρυπτογράφησης, ώστε ο ιός να είναι διαφορετικός στο σύνολό του σε κάθε μολυσμένο αρχείο.

Τέλος, μια άλλη μέθοδος που αναφέρουν οι συγγραφείς είναι ο **μεταμορφισμός** (**metamorphism**). Ένας *μεταμορφικός ιός* δεν βασίζεται στην κρυπτογράφηση, αλλά υλοποιεί έναν εξαιρετικά σύνθετο μηχανισμό που έχει ως αποτέλεσμα ο κώδικας του ιού να ξαναγράφεται από την αρχή, χωρίς όμως να αλλάζει το τελικό αποτέλεσμα της εκτέλεσής του.

[51] Είναι η λεγόμενη *υπογραφή* (*signature*) ενός ιού. Στην ενότητα «Συστήματα Antimalware» θα εξηγηθεί με περισσότερες λεπτομέρειες η έννοια της υπογραφής ενός κακόβουλου προγράμματος.

[52] Ένας τυπικός ιός αποτελείται στην πραγματικότητα από δύο τουλάχιστον λογικά συστατικά μέρη: τον μηχανισμό μόλυνσης και το φορτίο. Κάποιοι ιοί περιλαμβάνουν και ένα τρίτο μέρος που ασχολείται με την ενεργοποίησή του υπό συγκεκριμένες συνθήκες. Για χάρη απλότητας στην επεξήγηση των μορφών ιών, στην παρούσα εργασία όλα αυτά τα συστατικά μέρη εκλαμβάνονται ως είναι ένα τμήμα που ονομάζεται *κυρίως κώδικας*.

4.1.3.2 Σκουλήκι

Ένα **σκουλήκι** (**worm**) είναι ένα αυτό-αναπαραγόμενο πρόγραμμα [53] που διαδίδεται αυτόνομα σε ένα δίκτυο υπολογιστών (Shah, Shah, & Shah, 2017), εκμεταλλευόμενο ευπάθειες ασφαλείας (Kamal, Ali, Alani, & Abdulmajed, 2016). Τα σκουλήκια μοιάζουν σε μεγάλο βαθμό με τους ιούς, αφού και αυτά εφαρμόζουν την αυτό-αναπαραγωγή, όμως σε αντίθεση με τους ιούς, τα σκουλήκια είναι ανεξάρτητα προγράμματα και δεν χρειάζονται αρχεία-ξενιστές για να εξαπλωθούν (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020) [54].

Οι Kama, Ali, Alani, & Abdulmajed (2016) χωρίζουν τα σκουλήκια σε δύο κατηγορίες:

- *Σκουλήκια που βασίζονται στη σάρωση* [55] για να διαδοθούν. Αυτά, αρχικά επιλέγουν μια διεύθυνση IP, στη συνέχεια σαρώνουν / ελέγχουν αν το υπολογιστικό σύστημα είναι επιρρεπές σε κάποια ευπάθεια που μπορούν να εκμεταλλευτούν και αν είναι, διαδίδονται σε αυτό. Για την επιλογή της IP υπάρχουν δύο στρατηγικές· είτε τυχαία, είτε με προτεραιότητα σε τοπικές IP. Είναι σημαντικό να παρατηρήσουμε πως αυτή η κατηγορία σκουληκιών, δεν απαιτεί καμία απολύτως ενέργεια από κάποιον χρήστη για μολύνει το σύστημα.
- *Σκουλήκια που βασίζονται στην τοπολογία των δικτύων* για να διαδοθούν. Αυτά βασίζονται σε πληροφορίες που αποκτούν από επιτυχημένες εισβολές, ώστε να κάνουν μια λίστα νέων στόχων επίθεσης.

  Έτσι για παράδειγμα, τα *σκουλήκια του ηλεκτρονικού ταχυδρομείου*, μετά από μια εισβολή – μόλυνση ενός στόχου, επισυνάπτουν ένα αντίγραφό τους σε ένα μήνυμα και το προωθούν σε όλες τις επαφές. Μόλις ένας χρήστης κατεβάσει το συνημμένο αρχείο [56], το σκουλήκι έχει κάνει μια νέα επιτυχημένη εισβολή και η διαδικασία ξεκινάει από την αρχή.

  Με την ίδια λογική λειτουργούν τα *σκουλήκια των μέσων κοινωνικής δικτύωσης*, τα *σκουλήκια P2P* κ.α. Είναι σημαντικό να παρατηρήσουμε ότι αυτή η κατηγορία βασίζεται στην κοινωνική μηχανική, αφού η εξάπλωση του σκουληκιού εξαρτάται από την εξαπάτηση των χρηστών.

Ένα σημείο που αξίζει να σημειωθεί είναι πως ο μηχανισμός διάδοσης των σκουληκιών επηρεάζει το εύρος ζώνης (bandwidth) του δικτύου, τόσο, ώστε σε πολλές περιπτώσεις να μειώνεται σημαντικά η αποδοτικότητά του. Επίσης, η αυτονομία με την οποία λειτουργούν, κάνει τα σκουλήκια μια από τις σημαντικότερες και διαρκείς απειλές στο Διαδίκτυο, καθώς ένας μόνο μολυσμένος υπολογιστής αρκεί, ώστε το σκουλήκι να προσπαθεί διαρκώς να διαδοθεί σε ολόκληρο το Διαδίκτυο.

---

[53] Όπως και οι ιοί, τα σκουλήκια κατατάσσονται θεωρητικά στα αυτό-αναπαραγόμενα αυτόματα που περιέγραψε ο John von Neumann (δείτε και υποσημείωση 48).

[54] Παρόμοια με έναν ιό, η διαδικασία διάδοσής ενός σκουληκιού αναφέρεται συχνά ως *μόλυνση* και το σύστημα που μολύνεται (και όχι το αρχείο), λέγεται *ξενιστής*.

[55] Άλλοι ερευνητές ονομάζουν τα σκουλήκια που λειτουργούν με αυτή τη μέθοδο *σκουλήκια του Διαδικτύου* (*internet worms*) – λ.χ. Groš (2021).

[56] Σε μια παραλλαγή της τεχνικής αυτής, το σκουλήκι δεν επισυνάπτει κάποιο αντίγραφο του εαυτού του στο email, αλλά περιλαμβάνει σε αυτό έναν υπερσύνδεσμο ενός ιστότοπου που μπορεί να το διαδώσει.

4.1.3.3 Δούρειος ίππος

Ένας **δούρειος ίππος** (**Trojan horse** ή απλά **Trojan**) είναι ένα χρήσιμο ή φαινομενικά χρήσιμο πρόγραμμα που περιέχει κρυφό και κακόβουλο κώδικα, που εκτελείται μαζί με το υπόλοιπο πρόγραμμα (NIST, χ.χ.). Η διάδοσή του βασίζεται στην κοινωνική μηχανική (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020), καθώς πρέπει να εξαπατηθεί κάποιος χρήστης για να το μεταφορτώσει και να το εκτελέσει στον υπολογιστή του.

Το όνομα του είδους παραπέμπει στον μύθο του Δούρειου ίππου, που άφησαν οι Αχαιοί ως «δώρο» και όταν οι Τρώες το έβαλαν μέσα στην πόλη τους, οι πολεμιστές που κρύβονταν μέσα του βγήκαν τη νύχτα και την κατέστρεψαν. Έτσι και το λογισμικό αυτό (*δούρειος ίππος*), με χρήση της κοινωνικής μηχανικής, εξαπατά *(«δώρο»*) τον χρήστη (*Τρώες*) για να το μεταφορτώσει στον υπολογιστή του (*πέρασμα στην πόλη*) και να το εκτελέσει (*καταστροφή*).

Όπως και σε έναν ιό, ο κακόβουλος κώδικας ενός δούρειου ίππου τίθεται σε λειτουργία κατά την εκτέλεση ενός προγράμματος· όμως σε αντίθεση με έναν ιό, ένας δούρειος ίππος δεν μολύνει κάποιο προϋπάρχον πρόγραμμα, αλλά το πρόγραμμά του δημιουργείται εξ αρχής με ενσωματωμένο τον κακόβουλο κώδικα. Άρα, ένας δούρειος ίππος είναι ανεξάρτητο πρόγραμμα· αλλά όχι αυτόνομο όπως ένα σκουλήκι, διότι χρειάζεται τις ενέργειες ενός χρήστη για να εκτελεστεί. Επίσης, σε αντίθεση με έναν ιό ή ένα σκουλήκι, δεν διαθέτει κώδικα για την εξάπλωσή του σε άλλα συστήματα. (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020)

Τέλος, αξίζει να σημειωθεί πως, παρότι οι δούρειοι ίπποι δεν διαθέτουν έναν εγγενή μηχανισμό που να τους επιτρέπει να διαδίδονται μαζικά από ένα *προσβεβλημένο* (*compromised*) [57] υπολογιστικό σύστημα στο επόμενο, στις μέρες μας, αποτελούν ίσως το πιο κοινό «όπλο» των επιτιθέμενων. Αυτό οφείλεται στο γεγονός ότι δεν εξαρτώνται από ευπάθειες ασφαλείας του λογισμικού, αλλά βασίζονται σε απειλές ασφαλείας που οφείλονται σε ανθρώπινους παράγοντες· αρκεί ένας χρήστης του συστήματος να εκτιμήσει λανθασμένα κάποια απόπειρα εξαπάτησης, ώστε ο δούρειος ίππος να εισβάλει στο σύστημα.

[57] Στα πλαίσια της εργασίας αυτής, ένα σύστημα αποκαλείται **προσβεβλημένο** (**compromised**), αν σε αυτό εκτελείται κακόβουλο λογισμικό ή γενικά εκτελείται κάποια κακόβουλη λειτουργία. Ο όρος **μολυσμένο** (**infected**) περιορίζεται στους ιούς και τα σκουλήκια, που έχουν τη δυνατότητα αυτόματης εξάπλωσης.

Πάντως, στην καθημερινή χρήση, αλλά και σε μέρος της βιβλιογραφίας, ο όρος μολυσμένο και προσβεβλημένο χρησιμοποιούνται ως συνώνυμοι.

4.1.3.4 Rootkit

Ο όρος **rootkit** [58] αναφέρεται σε προγράμματα που πρωτοεμφανίστηκαν στο λειτουργικό σύστημα UNIX επιτρέποντας σε έναν χρήστη την κλιμάκωση των προνομίων του. Αυτό επιτυγχανόταν με εγκατάσταση αλλοιωμένων (*subverted*) εκδόσεων των αυθεντικών προγραμμάτων του συστήματος. Ο όρος βρήκε χρήση και πέρα από το UNIX, όπου γενικά σήμαινε ένα πρόγραμμα που θα επέτρεπε σε κάποιον να έχει πρόσβαση διαχειριστή σε ένα σύστημα που έχει προσβληθεί (Bravo & García, 2020). Ωστόσο, αυτή η πρώτη γενιά rootkit ήταν εύκολα ανιχνεύσιμη [59], από τα αρχεία που εγκαθίσταντο στον σκληρό δίσκο του συστήματος. (Rudd, Rozsa, Günther, & Boult, 2017)

Για τον λόγο αυτό, πρώτιστος σκοπός της νεότερης γενιάς είναι να αποκρύψει την παρουσία του rootkit στο σύστημα. Αυτό γίνεται αποκρύπτοντας τα αρχεία του στον σκληρό δίσκο ή/και τις διεργασίες του που εκτελούνται στη μνήμη. (Rudd, Rozsa, Günther, & Boult, 2017) Για να πετύχει τον σκοπό του, χρησιμοποιεί την τεχνική της «*αγκίστρωσης*» (*hooking*) [60] σε διεργασίες της κύριας μνήμης.

Γενικά, η μεθοδολογία που εφαρμόζει ένα rootkit νέας γενιάς είναι:

- Αγκίστρωση σε βιβλιοθήκες DLL ή αντικατάσταση βιβλιοθηκών DLL με εκδόσεις που έχουν αλλοιωμένο κώδικα. Και στις δύο περιπτώσεις στόχος είναι βιβλιοθήκες DLL που εκτελούνται στο επίπεδο χρήστη και ελέγχουν την επικοινωνία των διεργασιών χρήστη με τον πυρήνα του λειτουργικού συστήματος.
- Αλλοίωση τμημάτων του ίδιου του πυρήνα του λειτουργικού συστήματος. Σε αυτή την περίπτωση το rootkit παίρνει τη θέση ενός οδηγού συσκευής (device driver) στα Windows ή ενός kernel module στο Linux.
- Αξιοποίηση της τεχνολογίας virtualization διαφόρων επεξεργαστών, ώστε το rootkit να λειτουργεί ως ένας υπερεπόπτης (hypervisor) που θα έχει στον έλεγχό του ολόκληρο το λειτουργικό σύστημα.
- Αλλοίωση του firmware του συστήματος. Στην περίπτωση που στόχος είναι το BIOS, το rootkit θα μπορούσε να κάνει οτιδήποτε, αφού θα λειτουργούσε πριν ακόμα γίνει η φόρτωση του λειτουργικού συστήματος.

(Bravo & García, 2020)

Στην πράξη, τα σύγχρονα rootkit δεν χρησιμοποιούνται ανεξάρτητα, αλλά οι δυνατότητές τους αξιοποιούνται για την απόκρυψη ενός πιο σύνθετου κακόβουλου λογισμικού.

[58] Ο όρος προέρχεται από τη σύνθεση των λέξεων *root* και *kit* και σε ελεύθερη μετάφραση σημαίνει «*εργαλειοθήκη του υπερχρήστη*». Να σημειωθεί πως root στο UNIX και τους απογόνους του, είναι ο υπερχρήστης.

[59] Στην περίπτωση που το rootkit ήταν αλλοιωμένη έκδοση ενός αυθεντικού προγράμματος, ένας έλεγχος checksum αρκούσε για την αποκάλυψή του. (Bravo & García, 2020)

[60] Η τεχνική αυτή στοχεύει στην αλλαγή της ροής του κώδικα των διεργασιών, ώστε να εκτελεστεί ο δικός τους κακόβουλος κώδικας, που συχνά αλλοιώνει δεδομένα των γνήσιων διεργασιών. (Rudd, Rozsa, Günther, & Boult, 2017)

4.1.3.5 Σύγχρονο κακόβουλο λογισμικό

Στις μέρες μας, ο διαχωρισμός του κακόβουλου λογισμικού σε ιούς, σκουλήκια, δούρειους ίππους, rootkit είναι αρκετά προβληματικός. Ο λόγος είναι πως το κακόβουλο λογισμικό είναι τόσο ανεπτυγμένο που περιλαμβάνει αρκετές διαφορετικές λειτουργίες, ώστε για παράδειγμα να εξαπλώνεται σε έναν υπολογιστή όπως ένας ιός, να διαδίδεται στο δίκτυο όπως ένα σκουλήκι και να χρησιμοποιεί τις τεχνικές ενός rootkit για να αποκρύψει την παρουσία του στο σύστημα (Rudd, Rozsa, Günther, & Boult, 2017).

Στις προηγούμενες ενότητες, που έγινε αναφορά στη λειτουργία των ιών, των σκουληκιών και των δούρειων ίππων, η προσοχή μας επικεντρώθηκε στον μηχανισμό διάδοσης και δεν έγινε πολύς λόγος για το φορτίο του κάθε είδους. Στη σύγχρονη μορφή τους αυτά τα είδη λειτουργούν απλά ως ο «μεταφορέας» και η κύρια κακόβουλη λειτουργία επιτελείται από το φορτίο· και τις περισσότερες φορές το ίδιο φορτίο δεν εξαρτάται από το είδος του κακόβουλου λογισμικού, αλλά μπορεί «παραδοθεί» από διαφορετικά είδη σε ένα υπολογιστικό σύστημα.

Έτσι λοιπόν, για την περιγραφή του σύγχρονου κακόβουλου λογισμικού δίπλα στους κλασικούς όρους ιός, σκουλήκι και δούρειος ίππος, εμφανίζονται όροι όπως:

- **backdoor** (*κερκόπορτα* [61]): ονομάζεται ένα πρόγραμμα ή μια μέθοδος που σκοπεύει στην παράκαμψη του τυπικού ελέγχου πρόσβασης σε ένα σύστημα, ώστε κρυφά να γίνει δυνατή η απομακρυσμένη πρόσβαση σε αυτό. (Hellenica World, 2023)
- **spyware** (***spy****ing soft****ware*** – *κατασκοπευτικό λογισμικό*): ονομάζεται το λογισμικό που υποκλέπτει προσωπικά δεδομένα (ονοματεπώνυμο, ΑΦΜ κ.α.), διαπιστευτήρια χρηστών (username και password), αριθμούς τραπεζικών λογαριασμών, αριθμούς πιστωτικών καρτών, διευθύνσεις IP, το ιστορικό περιήγησης στον Παγκόσμιο Ιστό κ.α. (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020) Μπορεί να χρησιμοποιεί *keyloggers*, που καταγράφουν όσα πληκτρολογούν οι χρήστες ή ακόμα και *sniffers*, που συλλαμβάνουν όλα τα πακέτα δεδομένων που διακινούνται από και προς τον υπολογιστή στο δίκτυο.
- **ransomware** (*λυτρισμικό*): ονομάζεται το λογισμικό που χρησιμοποιεί κρυπτογράφηση για να κάνει μη προσβάσιμα τα αρχεία σε ένα σύστημα. Στη συνέχεια, εμφανίζει ένα μήνυμα με το οποίο ζητάει να πληρωθούν «λύτρα» [62], σε αντάλλαγμα για την αποκρυπτογράφηση των αρχείων. (Ngo, Agarwal, Ramakrishna, & MacDonald, 2020)

[61] Αποδίδεται στα ελληνικά ως *κερκόπορτα*, παραπέμποντας στον θρύλο της άλωσης της Κωνσταντινούπολης.

[62] Συνήθως σε μορφή κρυπτονομισμάτων, ώστε να αξιοποιηθεί η ανωνυμία που προσφέρουν στον παραλήπτη.

- **λογισμικό cryptojacking** [63] ή **λογισμικό κακόβουλου cryptomining** [64]: ονομάζεται το λογισμικό που εγκαθίσταται και λειτουργεί κρυφά σε ένα υπολογιστικό σύστημα με σκοπό την αξιοποίηση των υπολογιστικών πόρων του για την εξόρυξη κρυπτονομισμάτων [65]. Η λειτουργία του έχει δύο επιπτώσεις στο σύστημα: πρώτον, αυξάνεται η κατανάλωση ηλεκτρικού ρεύματος και δεύτερον, μειώνεται δραματικά η υπολογιστική ισχύς που διατίθεται για την εκτέλεση των νόμιμων εργασιών. (Tekiner, Acar, Uluagac, Kirda, & Selcuk, 2021)
- **creepware** [66] ή **RAT** [67]: ονομάζεται μια ειδική κατηγορία δούρειου ίππου που επιτρέπει σε έναν κακόβουλο παράγοντα να αποκτήσει απομακρυσμένα μη εξουσιοδοτημένη πρόσβαση και έλεγχο σε ένα υπολογιστικό σύστημα. Λόγω των χαρακτηριστικών που συνδυάζει, συχνά κατατάσσεται και ως backdoor και ως spyware. Ως δούρειος ίππος διαδίδεται με τεχνικές κοινωνικής μηχανικής, αλλά ξεχωρίζει κυρίως επειδή χρησιμοποιείται για πιο «προσωπικές» επιθέσεις στοχοποιώντας ένα συγκεκριμένο άτομο.

  Το λογισμικό αυτό επιτρέπει στον κακόβουλο χειριστή του να παρακολουθεί ότι πληκτρολογεί, ότι ακούει ή ότι βλέπει ο χρήστης – θύμα, ή ακόμα και να παρακολουθεί ζωντανά το ίδιο το θύμα μέσα από τη webcam και το μικρόφωνο του υπολογιστή. Πέρα από την απομακρυσμένη παρακολούθηση, επιτρέπει στον κακόβουλο χειριστή να εκτελέσει απομακρυσμένα οποιαδήποτε ενέργεια επιθυμεί.

  Αυτά τα χαρακτηριστικά το κάνουν ιδανικό για υποκλοπές, κυβερνοκατασκοπία και κυβερνοεκβιασμούς. Είναι δυνατόν επίσης, να χρησιμοποιηθεί κρυφά για την εκτέλεση παράνομων ενεργειών στο Διαδίκτυο με σκοπό να ενοχοποιηθεί ο νόμιμος χρήστης, που όμως δεν θα έχει ιδέα ότι οι παράνομες ενέργειες έγιναν από το σύστημά του.

  (Farinholt, 2019; Roundy, και συν., 2020; In.gr, 2013)

---

[63] Ο όρος *cryptojacking* προέρχεται από τις λέξεις **crypto**currency και **jacking** (που στα ελληνικά σημαίνει κλοπή), και σημαίνει ότι γίνεται κλοπή – υπολογιστικών πόρων – για την εξόρυξη κρυπτονομισμάτων.

[64] Ο όρος *cryptomining* προέρχεται από τις λέξεις **crypto**currency και **mining**, και σημαίνει *εξόρυξη κρυπτονομισμάτων*. Η διαδικασία είναι νόμιμη και μπορεί να επιφέρει κέρδος σε αυτούς που την εκτελούν. Πέρα από το νόμιμο υπάρχει και το κακόβουλο cryptomining που χρησιμοποιεί κρυφά ξένους υπολογιστικούς πόρους για την επίτευξη ιδίου οικονομικού οφέλους.

[65] Δείτε και *επίθεση cryptojacking* που αναλύεται σε επόμενο κεφάλαιο.

[66] Ο όρος προέρχεται από τη σύνθεση των λέξεων ***creep*** και *soft**ware***. Η λέξη creep χρησιμοποιείται για να δείξει ένα αίσθημα ανατριχίλας, φρίκης ή τρόμου. Στα ελληνικά χρησιμοποιείται χωρίς μετάφραση, αλλά θα μπορούσε να αποδοθεί ως *λογισμικό φρίκης* (από το αίσθημα που προκαλεί στο θύμα, μόλις καταλάβει τι συμβαίνει στο σύστημά του).

[67] Το ακρωνύμιο **RAT** μπορεί να σημαίνει είτε Remote Administration Tool (Εργαλείο Απομακρυσμένης Διαχείρισης), είτε Remote Access Tool (Εργαλείο Απομακρυσμένης Πρόσβασης) ή Remote Access Trojan (Δούρειος ίππος Απομακρυσμένης Πρόσβασης).

Η αποσαφήνιση του όρου ως *εργαλείο* (*tool*) δείχνει ότι πρόκειται για ένα νόμιμο και χρήσιμο πρόγραμμα για τους διαχειριστές υπολογιστικών συστημάτων που συχνά πρέπει να συνδεθούν μέσω δικτύου και όχι ως τοπικοί χρηστές για να εκτελέσουν κάποιες λειτουργίες στα συστήματά τους.

Αντιθέτως, η αποσαφήνιση ως *δούρειος ίππος* (*Trojan*) δείχνει ότι πρόκειται για ένα κακόβουλο πρόγραμμα, που λειτουργεί κρυφά και ότι η πρόσβαση και ο έλεγχος που δίνει σε ένα απομακρυσμένο σύστημα είναι χωρίς εξουσιοδότηση.

Όπως γίνεται κατανοητό, το «οπλοστάσιο» των κακόβουλων παραγόντων μπορεί να προσαρμοστεί ανάλογα με τους σκοπούς και τα μέσα που διαθέτει. Για παράδειγμα, ένας κακόβουλος παράγοντας θα μπορούσε να «ελευθερώσει» ένα σκουλήκι που διαδίδεται αυτόνομα στο Διαδίκτυο και κάθε φορά που μολύνει ένα υπολογιστικό σύστημα, να εκτελεί ένα ransomware για να αποκομίσει λύτρα από το θύμα. Εναλλακτικά, αν ο κακόβουλος παράγοντας έχει καταφέρει να αποκτήσει μια υποδομή για να αποστέλλει πλαστά email (τεχνική phishing), θα μπορούσε να επισυνάπτει σε αυτά έναν δούρειο ίππο, που όταν εκτελεστεί σε έναν υπολογιστή, όμοια με πριν να εκτελεί ένα ransomware για να απαιτήσει λύτρα από το θύμα.

4.1.3.6 Botnet, zombies και bots

Ο όρος **botnet** [68] περιγράφει ένα σύνολο από προσβεβλημένα υπολογιστικά συστήματα που βρίσκονται υπό τον έλεγχο ενός ανθρώπινου χειριστή που ονομάζεται *botmaster*. (Rodríguez-Gómez, Maciá-Fernández, & García-Teodoro, 2013) Καθένα από τα προσβεβλημένα συστήματα ονομάζεται *zombie* [69] και έχει περιέλθει υπό τον έλεγχο του botmaster, μετά την εισβολή ενός κακόβουλου **bot** [70]. Ένα bot συνδυάζει χαρακτηριστικά από άλλα είδη κακόβουλου λογισμικού· διαθέτει όμως και ένα ιδιαίτερο χαρακτηριστικό, που είναι το σύστημα **C&C** [71] για την απομακρυσμένη διαχείρισή του από τον botmaster. (Feily, Shahrestani, & Ramadass, 2009)

Η αξιοποίηση ενός μεγάλου πλήθους συστημάτων zombie ενισχύει την επίθεσή του κακόβουλου botmaster, ενώ παράλληλα δυσχεραίνει την αποκάλυψη της πραγματικής ταυτότητάς του. (Bailey, Cooke, Jahanian, Xu, & Karir, 2009) Τυπικά, τα bots ελέγχονται από έναν ή περισσότερους control servers, που διαβιβάζουν τις εντολές του botmaster. Πολλές φορές, ούτε οι control servers επικοινωνούν άμεσα με τον botmaster, αλλά χρησιμοποιείται ένα δίκτυο άλλων προσβεβλημένων συστημάτων που λειτουργούν ως μεσολαβητές τους. (Kamal, Ali, Alani, & Abdulmajed, 2016)

Όσον αφορά τα bots, άμεσα μετά την εισβολή σε ένα νέο σύστημα, προσπαθούν να επικοινωνήσουν με τον control server τους και αναμένουν εντολές. Η επικοινωνία με τον control server γίνεται, είτε με χρήση απευθείας της IP του, είτε με χρήση του domain name του. Αυτά, είτε είναι hardcoded στον κώδικα του bot, είτε παράγονται μέσω αλγορίθμου. (Kamal, Ali, Alani, & Abdulmajed, 2016)

Στον κύκλο ζωής των botnets ξεχωρίζουν τρεις μηχανισμοί:

- Ο *μηχανισμός διάδοσης*: τα bot μπορούν να διαδοθούν μέσω άλλου κακόβουλου λογισμικού που έχει το bot ως φορτίο ή να διαδοθούν ανεξάρτητα, ενσωματώνοντας τεχνικές που εφαρμόζονται σε άλλα είδη κακόβουλου λογισμικού. Σε αυτή την περίπτωση διακρίνονται: η ενεργητική διάδοση, που το bot αυτόνομα, χωρίς καμία παρέμβαση χρήστη, προσπαθεί να εκμεταλλευτεί ευπάθειες (λ.χ. με zero-day exploits) για να εξαπλωθεί· και η παθητική διάδοση, που βασίζεται στη χρήση κοινωνικής μηχανικής (λ.χ. διάδοση μέσω κακόβουλων ιστοτόπων). (Bailey, Cooke, Jahanian, Xu, & Karir, 2009; Kamal, Ali, Alani, & Abdulmajed, 2016)
- Ο *μηχανισμός C&C*: για τον χειρισμό των bot χρησιμοποιούνται διάφορα πρωτόκολλα (IRC, HTTP, P2P) και διάφορα μοντέλα επικοινωνίας (κεντρικοποιημένο, P2P, αδόμητο κ.α.). (Bailey, Cooke, Jahanian, Xu, & Karir, 2009; Li, Jiang, & Zou, 2009)

---

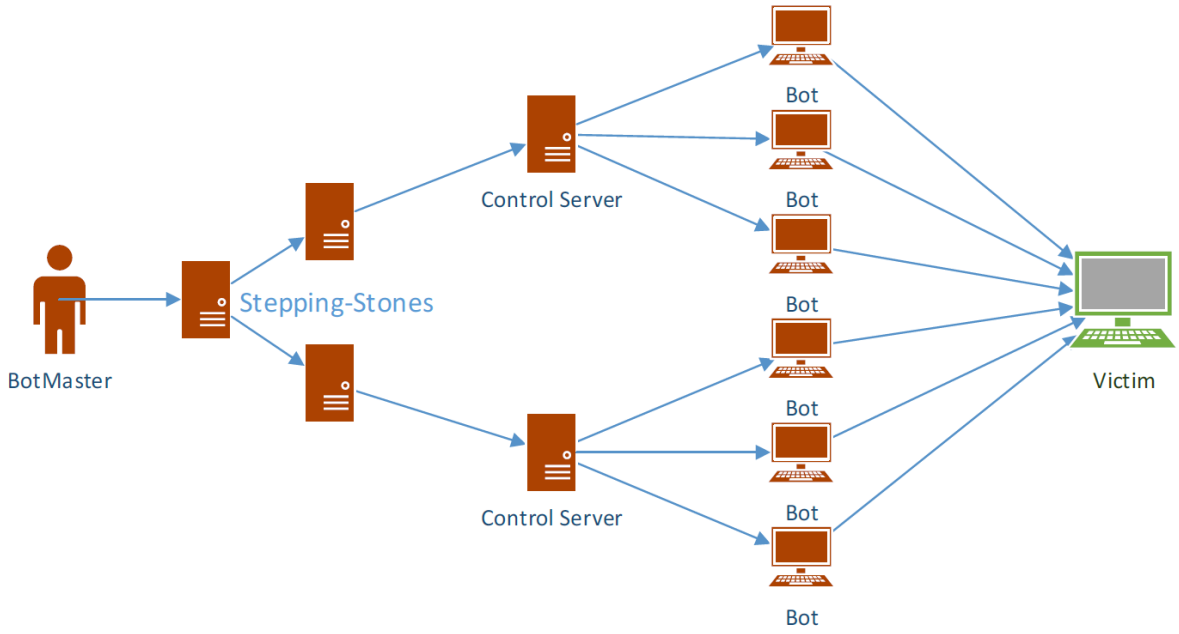
[68] Ο όρος προέρχεται από τη σύνθεση των λέξεων ro**bot** και **net**work. (Lee, 2023)

[69] Ονομάζεται *zombie* παραπέμποντας στον γνωστό μύθο των *ζωντανών – νεκρών*, δηλαδή πλασμάτων που παρότι είναι νεκρά, με μαγεία επανέρχονται στη ζωή, χωρίς όμως να έχουν δική τους βούληση. Έτσι και ο botmaster μπορεί να «ζωντανέψει» τα προσβεβλημένα συστήματα που βρίσκονται σε αδράνεια, όποτε επιθυμεί να εκτελέσει μια κακόβουλη ενέργεια.

[70] Γενικά, *robot* ή **bot** ή *internet bot* ονομάζεται το λογισμικό που χρησιμοποιείται για την εκτέλεση αυτοματοποιημένων εργασιών. Διακρίνονται σε διαφορετικές κατηγορίες ανάλογα με τη λειτουργία που εκτελούν· μία από αυτές είναι τα κακόβουλα bots. (WhatIs.com, 2023)

[71] **C&C**: συντομογραφία της φράσης *Command & Control*, που στα ελληνικά αποδίδεται ως *Διοίκηση & Έλεγχος*.

- Ο *μηχανισμός της κακόβουλης λειτουργίας*: συνήθεις κακόβουλες ενέργειες είναι η αξιοποίηση των πόρων του προσβεβλημένου συστήματος για αποστολή spam ή phishing email, τη φιλοξενία spoofed websites, την υποκλοπή δεδομένων των χρηστών των προσβεβλημένων συστημάτων, την εξόρυξη κρυπτονομισμάτων, την πραγματοποίηση επιθέσεων DDoS κ.α. (Bailey, Cooke, Jahanian, Xu, & Karir, 2009; Li, Jiang, & Zou, 2009)



Εικόνα 5: Τυπική ροή μηνυμάτων σε ένα botnet για την εκτέλεση μιας επίθεσης [πηγή: (Kamal, Ali, Alani, & Abdulmajed, 2016)]

Όσον αφορά την απόκρυψη της παρουσίας τους στα συστήματα, είναι σύνηθες τα bot να αξιοποιούν τον πολυμορφισμό και τις τεχνικές απόκρυψης των rootkit. Επίσης, το botnet μπορεί να επιχειρήσει να αποκρύψει και τις επικοινωνίες του με διάφορες μεθόδους, όπως χρήση κρυπτογράφησης, ανανέωση των IP των control servers κ.α. (Kamal, Ali, Alani, & Abdulmajed, 2016)

Αξίζει να σημειωθεί, πως τα botnet έχουν εξελιχθεί στη μεγαλύτερη σύγχρονη απειλή ασφαλείας, προκαλώντας τεράστιες οικονομικές απώλειες. (Li, Jiang, & Zou, 2009) Ένα από τα χαρακτηριστικότερα παραδείγματα είναι το *botnet Mirai* [72], που «στρατολογούσε» κατά κύριο λόγο συσκευές μικρής υπολογιστικής ισχύος για να πραγματοποιήσει επιθέσεις με τεράστιο αντίκτυπο. (Lee, 2023)

[72] Τον Σεπτέμβριο του 2016 εξαπολύθηκε μια από τις μεγαλύτερες επιθέσεις DDoS. Πίσω από την επίθεση αποκαλύφθηκε πως βρισκόταν το *botnet Mirai*, που αποτελείτο από προσβεβλημένες συσκευές μικρής υπολογιστικής ισχύος. Το botnet λειτουργούσε ως εξής: αρχικά, τα bots σάρωναν το Διαδίκτυο αναζητώντας συσκευές IoT και προσπαθούσαν να αποκτήσουν απομακρυσμένη πρόσβαση σε αυτές, με χρήση των εργοστασιακά προεπιλεγμένων usernames και passwords. Αν το κατάφερναν, απέστελλαν σήμα σε έναν report server, που σε δεύτερο χρόνο αναλάμβανε να μετατρέψει τη συσκευή σε zombie, μεταφορτώνοντας σε αυτήν το bot. Μετά τη μεταφόρτωση και την εγκατάσταση του bot, τα κακόβουλα αρχεία διαγράφονταν και το bot εκτελείτο μόνο ως διεργασία στην κύρια μνήμη, με σκοπό να αποκρυφθεί η παρουσία του. Από αυτό το σημείο και έπειτα, η συσκευή είχε γίνει μέρος του botnet. Όταν ο botmaster επιθυμούσε να πραγματοποιήσει μια επίθεση DDoS, έδινε εντολή στο botnet μέσω του C&C, ώστε οι συσκευές zombie να αρχίσουν να αποστέλλουν διαρκώς αιτήματα σε ένα συγκεκριμένο σύστημα – στόχο, με σκοπό να το καταστήσουν ανίκανο να εξυπηρετήσει αιτήματα από τους νόμιμους χρήστες του. (Lee, 2023)

4.2 Είδη επιθέσεων

Είναι ανάγκη να τονιστεί πως όλα τα συστήματα, όσο μικρά και αν είναι, μπορεί να αποτελέσουν στόχο ενός κακόβουλου παράγοντα που θα θελήσει με κάποιο τρόπο να τα εκμεταλλευτεί [73]. Η πεποίθηση ότι υπάρχει κάποιο υπολογιστικό σύστημα που δεν κινδυνεύει από κυβερνοεπιθέσεις είναι απλά μια ψευδαίσθηση. Συνεπώς, κάθε υπολογιστικό σύστημα πρέπει να λαμβάνει υπόψη μια ενδεχόμενη κυβερνοεπίθεση ως σοβαρό πρόβλημα και να είναι κατάλληλα προετοιμασμένο για να την αντιμετωπίσει επιτυχώς.

Και όπως ισχύει για όλα τα προβλήματα για τα οποία επιζητείται μια λύση, το πρώτο βήμα είναι η κατανόησή τους. Οι Ramsdale, Shiaeles, & Kolokotronis (2020) σημειώνουν πως, η κατανόηση των ίδιων των επιτιθέμενων αν και έχει γίνει πιο πολύπλοκη (σε σχέση με το παρελθόν), είναι παρόλα αυτά ακόμα πιο σημαντική, καθώς αν καταστεί δυνατόν να εξαχθούν χρήσιμες πληροφορίες, αυτές θα επιτρέψουν την προσαρμογή της άμυνας του συστήματος με έναν αυτοματοποιημένο τρόπο που θα βελτιώσει την προστασία του.

Στα πλαίσια της κατανόησης των επιτιθέμενων και των επιθέσεων, οι μελέτες έχουν καταδείξει διαφορετικούς τύπους επιθέσεων, ανάλογα με την οπτική από την οποία εξετάζονται κάθε φορά.

Οι Simmons, Ellis, Shiva, Dasgupta & Wu (2009) επιχειρώντας μια συστηματική ταξινόμηση των επιθέσεων κατέληξαν σε διάφορες μορφές κατηγοριοποίησης που περιλαμβάνουν:

- κατηγοριοποίηση με βάση το διάνυσμα επίθεσης
- κατηγοριοποίηση με βάση τον επιχειρησιακό αντίκτυπο στον οργανισμό
- κατηγοριοποίηση με βάση τον αντίκτυπο στις πληροφορίες
- κατηγοριοποίηση με βάση τον στόχο της επίθεσης
- κατηγοριοποίηση με βάση την άμυνα που μπορεί να αξιοποιηθεί κατά της επίθεσης.

Η πρότασή τους είναι εκτενής και διαχωρίζει τις επιθέσεις σε αρκετά τεχνικό επίπεδο, ξεφεύγοντας από τα όρια της παρούσας εργασίας και για αυτό δεν θα αναλυθούν εξονυχιστικά οι κατηγορίες που προτείνουν. Ωστόσο, το πλήθος των διαφορετικών προτάσεων που παρέχουν σχετικά με την κατηγοριοποίηση των επιθέσεων, αποδεικνύει ότι οι επιθέσεις είναι πολυάριθμες και συνεπώς η αντιμετώπισή τους δεν είναι τετριμμένη υπόθεση. Στη συνέχεια, θα αναλυθούν εν συντομία κάποιες από τις σημαντικότερες επιθέσεις.

[73] Η θέση αυτή επιβεβαιώνεται περίτρανα από το παράδειγμα του botnet Mirai (δείτε υποσημείωση 72).

Ξεκινώντας από ένα πιο γενικό πλαίσιο, ξεχωρίζει από το έργο των Biju, Gopal & Prakash (2019) η πιο στοιχειώδης διάκριση που μπορεί να γίνει για τις επιθέσεις, και αφορά:

- *Στοχευμένες* επιθέσεις, στις οποίες υπάρχει κάποιο ιδιαίτερο ενδιαφέρον που καθορίζει την επιλογή ενός συγκεκριμένου στόχου. Αυτές οι επιθέσεις απαιτούν χρόνο προετοιμασίας, ώστε να βρεθεί ο καλύτερος τρόπος για να πραγματοποιηθούν. [74]
- *Μη στοχευμένες* επιθέσεις, με κάθε επίθεση να εξαπολύεται μαζικά και αδιάκριτα με σκοπό να επηρεάσει όσο περισσότερα συστήματα γίνεται. [75]

Και οι Uma & Padmavathi (2013) ασχολήθηκαν με το ίδιο θέμα και αναφέρουν μια άλλη βασική διάκριση μεταξύ των επιθέσεων, που ανάλογα με τον βαθμό εμπλοκής του επιτιθέμενου διαχωρίζονται σε:

- *Παθητικές* (*passive*) επιθέσεις, στις οποίες ο επιτιθέμενος έχει καταφέρει να παρακολουθεί ή ακόμα και να υποκλέπτει πληροφορίες του συστήματος, χωρίς όμως να παρεμβαίνει σε αυτές.
- *Ενεργητικές* (*active*) επιθέσεις, στις οποίες ο επιτιθέμενος, πέρα από το να παρακολουθεί το σύστημα, μπορεί κατά βούληση να εισάγει ή να αλλοιώνει ή να καταστρέφει τελείως πληροφορίες ή να διακόπτει την πρόσβαση άλλων σε αυτές.

Από τη μελέτη των Uma & Padmavathi (2013) και πάλι μπορούμε να αναφέρουμε ακόμα δύο είδη επιθέσεων, που μπορούν να πραγματοποιηθούν με πολλές διαφορετικές τεχνικές:

- Επιθέσεις *αναγνώρισης* (*reconnaissance*), στις οποίες σκοπός είναι να συγκεντρωθούν χρήσιμες πληροφορίες για τον στόχο της επίθεσης. Η πιο απλή τεχνική που μπορεί να εφαρμοστεί σε αυτή την περίπτωση περιλαμβάνει συνηθισμένα *DNS queries* από τα οποία μπορεί εύκολα να μάθει ο επιτιθέμενος τα domain names και τις αντίστοιχες διευθύνσεις IP που τον ενδιαφέρουν.

  Μια άλλη τεχνική αναγνώρισης είναι το *ping scanning*, δηλαδή η σάρωση του δικτύου για να βρεθούν οι IP των ενεργών κόμβων (αυτή η μέθοδος έχει αξία όταν ο επιτιθέμενος καταφέρει να αποκτήσει πρόσβαση σε ένα εσωτερικό δίκτυο, το οποίο έχει παντελώς άγνωστη δομή σε αυτόν).

  Άλλη μέθοδος είναι το *port scanning*, που επιτρέπει στον επιτιθέμενο να βρει ποιες υπηρεσίες εκτελούνται (και κατ' επέκταση ποιο λογισμικό εκτελείται) σε κάθε κόμβο (είτε του Διαδικτύου, είτε του εσωτερικού δικτύου), ώστε σε δεύτερο βαθμό να προσπαθήσει να εκμεταλλευτεί γνωστές ευπάθειές τους.

[74] Για παράδειγμα, στοχευμένη επίθεση ήταν αυτή που αξιοποιούσε το λογισμικό Stuxnet (δείτε υποσημείωση 46). Άλλο παράδειγμα στοχευμένης επίθεσης είναι μια επίθεση με χρήση λογισμικού RAT.

[75] Για παράδειγμα, μη στοχευμένες επιθέσεις πραγματοποιούνται από ιούς και σκουλήκια, που όπως αναλύθηκε στη σχετική ενότητα, προσπαθούν να εξαπλωθούν αυτόματα σε όσο το δυνατόν περισσότερα συστήματα. Πραγματοποιούνται επίσης και από δούρειους ίππους που διαδίδονται μέσω phishing email ή μέσω κακόβουλων ιστοτόπων.

- Επιθέσεις *πρόσβασης* (*access*), που έχουν σκοπό να επιτρέψουν στον επιτιθέμενο να εισβάλει στο σύστημα. Υπάρχουν πάρα πολλές προσεγγίσεις για να επιτευχθεί αυτός ο στόχος, με πιο απλή στην υλοποίηση να είναι η λεγόμενη *επίθεση brute force*, στην οποία ο επιτιθέμενος δοκιμάζει διαρκώς τυχαίους κωδικούς πρόσβασης μέχρι να καταφέρει να συνδεθεί στο σύστημα (Biju, Gopal, & Prakash, 2019).

  Άλλη προσέγγιση είναι αυτή που χρησιμοποιείται σε μια *επίθεση λεξιλογίου* (*dictionary attack*) στην οποία ο επιτιθέμενος χρησιμοποιεί ευρύτερα γνωστούς συνδυασμούς username και password για να αποκτήσει (απομακρυσμένη) πρόσβαση ως νόμιμος χρήστης του συστήματος (Uma & Padmavathi, 2013) [76].

  Με ίδιο στόχο μπορεί να γίνει μια *επίθεση phishing*, στην οποία, εφαρμόζοντας την ομώνυμη τεχνική που περιγράφηκε σε προηγούμενη ενότητα, ο επιτιθέμενος προσπαθεί να εξαπατήσει έναν ή περισσότερους νόμιμους χρήστες του συστήματος και να τους υποκλέψει το username και το password που θα του δώσουν την επιθυμητή πρόσβαση στο σύστημα.

  Άλλες, πιο προηγμένες προσεγγίσεις αξιοποιούνται όταν ο επιτιθέμενος έχει επιτύχει μια εισβολή στο σύστημα, αλλά χωρίς να έχει πρόσβαση με τα δικαιώματα ή στους υπολογιστικούς πόρους που επιθυμεί. Σε αυτή την περίπτωση, ενδεικτικά, μπορεί να αναζητήσει τα hash των password σε Βάσεις Δεδομένων που αποθηκεύουν στοιχεία σύνδεσης χρηστών ή να εφαρμόσει την τεχνική *sniffing*, δηλαδή να παρακολουθεί τις επικοινωνίες στο εσωτερικό δίκτυο, ελπίζοντας να καταφέρει να βρει στοιχεία σύνδεσης που μεταδίδονται σε μορφή *απλού κειμένου* [77] (Biju, Gopal, & Prakash, 2019). Σε περίπτωση που η επιτυχημένη εισβολή έχει πραγματοποιηθεί με χρήση κακόβουλου λογισμικού που εκτελεί ως φορτίο λειτουργίες κατασκοπίας (δείτε *spyware* και *RAT*), τότε θα μπορέσει να υποκλέψει τα στοιχεία πρόσβασης που τον ενδιαφέρουν από τους χρήστες ή το δίκτυο.

  Τέλος, στην ίδια κατηγορία εντάσσεται και μια *επίθεση port redirection*, στην οποία πρέπει και πάλι ο επιτιθέμενος να έχει καταφέρει να εισβάλει σε έναν υπολογιστικό κόμβο του συστήματος· εκμεταλλευόμενος την εμπιστοσύνη που υπάρχει σε αυτόν τον κόμβο ως συστατικό μέρος του συστήματος, ο επιτιθέμενος μπορεί να διακινεί δεδομένα από και προς το εσωτερικό δίκτυο – με άλλα λόγια, να πραγματοποιεί επιθέσεις (αναγνώρισης, πρόσβασης ή άλλου είδους) σε άλλους κόμβους του συστήματος – επιθέσεις που δεν θα είχε τη δυνατότητα να εκτελέσει ως εξωτερικός χρήστης (Uma & Padmavathi, 2013).

Στις επόμενες ενότητες θα εξεταστούν, με περισσότερες λεπτομέρειες, μερικοί ακόμα βασικοί τύποι επιθέσεων, που έχουν ιδιαίτερα σοβαρό αντίκτυπο στο σύστημα.

[76] Την επίθεση αυτή αξιοποιούσε το botnet Mirai κατά τη φάση διάδοσης (δείτε υποσημείωση 72).

[77] **Απλό κείμενο** (*cleartext*): όρος που χρησιμοποιείται για να περιγράψει δεδομένα σε μη κρυπτογραφημένη μορφή / μορφή πλήρως κατανοητή από ανθρώπους και υπολογιστές. Ως απλό κείμενο αποδίδεται στα ελληνικά και ο όρος *plaintext* που χρησιμοποιείται για τα δεδομένα πριν αυτά κρυπτογραφηθούν ή αφού αποκρυπτογραφηθούν. Συχνά, οι όροι cleartext και plaintext χρησιμοποιούνται ως συνώνυμοι. (NIST, χ.χ.) Σε κάθε περίπτωση, η αποθήκευση, και πολύ περισσότερο, η μετάδοση ευαίσθητων δεδομένων (όπως τα password) σε μορφή απλού κειμένου θεωρούνται σοβαρές ευπάθειες στην Ασφάλεια Πληροφοριών.

4.2.1 Επιθέσεις Buffer Overflow και Buffer Overread

Στις εφαρμογές χρησιμοποιούνται *buffer* (*δομές προσωρινής μνήμης*) για τον χειρισμό των δεδομένων που παρέχουν οι χρήστες· αν δεν γίνεται σωστός έλεγχος των ορίων μιας τέτοιας δομής, τότε είναι πιθανό να προκύψει *buffer overflow* (*υπερχείλιση της προσωρινής μνήμης*). Πρόκειται για ένα προγραμματιστικό σφάλμα που έχει ως αποτέλεσμα τα δεδομένα της προσωρινής μνήμης να «ξεχειλίζουν», δηλαδή να περνούν το επιτρεπτό όριο και να γράφονται πάνω από άλλα δεδομένα του προγράμματος που βρίσκονται σε γειτονικές θέσεις μνήμης.

Χωρίς να εμβαθύνουμε στις τεχνικές λεπτομέρειες, μπορούμε να πούμε πως ένας κακόβουλος χρήστης που θα ανακαλύψει ένα τέτοιο προγραμματιστικό σφάλμα – ελάττωμα του προγράμματος – ευπάθεια, μπορεί να το εκμεταλλευτεί για να κάνει code injection στο πρόγραμμα, που θα του επιτρέψει την εκτέλεση αυθαίρετου κώδικα (δείτε τεχνικές *ACE* και *RCE*) ή/και την κλιμάκωση των προνομίων του.

(Simmons, Ellis, Shiva, Dasgupta, & Wu, 2009; Shahriar & Zulkernine, 2012)

Με παρόμοιο τρόπο μπορεί να επιτευχθεί μια *επίθεση buffer overread*, στην οποία ο κακόβουλος χρήστης καταφέρνει να διαβάσει τιμές γειτονικών δεδομένων στο buffer, που ενδεχομένως να είναι ευαίσθητα και να αφορούν άλλους χρήστες και στα οποία να μην επιτρέπεται να έχει πρόσβαση υπό κανονικές συνθήκες. (Ullah, και συν., 2018) Η πιο γνωστή ευπάθεια buffer overread σχετίζεται με το *σφάλμα Heartbleed* [78], που, όταν αποκαλύφθηκε το 2014 – και ενώ υπήρχε για δύο χρόνια, θεωρήθηκε ως η χειρότερη ευπάθεια από τότε που ξεκίνησε το ηλεκτρονικό εμπόριο στο Διαδίκτυο (Carvalho, DeMott, Ford, & Wheeler, 2014).

[78] Το *σφάλμα Heartbleed* σχετίζεται με τον *μηχανισμό Heartbeat* που προτάθηκε για τον έλεγχο και τη διατήρηση μιας ασφαλούς σύνδεσης για μεγάλο χρονικό διάστημα, χωρίς να απαιτείται η επαναδιαπραγμάτευση μεταξύ των δύο πλευρών. Η λογική του μηχανισμού Heartbeat είναι, πως μέσω του ασφαλούς καναλιού, η μια πλευρά αποστέλλει ένα αυθαίρετο μήνυμα στην άλλη πλευρά, που με τη σειρά της απαντάει με το ίδιο αυθαίρετο μήνυμα, αποδεικνύοντας πως η σύνδεση είναι, και πρέπει να παραμείνει ενεργή. Ένα προγραμματιστικό σφάλμα στην υλοποίηση του μηχανισμού στο OpenSSL, αλλά και το γεγονός πως ο μηχανισμός ήταν ενεργός από προεπιλογή στην έκδοση 1.0.1, έκαναν το OpenSSL ευάλωτο σε μια επίθεση buffer overread, που ονομάστηκε *επίθεση Heartbleed*.

Η επίθεση Heartbleed ήταν σχετικά απλή. Αρχικά, ο επιτιθέμενος έπρεπε να δημιουργήσει μια ασφαλή σύνδεση με έναν τρωτό server. Μετά από αυτό, αρκούσε να αποστείλει παραποιημένα αιτήματα Heartbeat, στα οποία δινόταν ένα αυθαίρετο μήνυμα, αλλά καθοριζόταν ψευδώς ένα πολύ μεγάλο μέγεθος για το μήνυμα που αποστελλόταν. Λόγω της αστοχίας στην υλοποίηση του μηχανισμού στο OpenSSL, δεν γινόταν έλεγχος ώστε το μέγεθος που δηλωνόταν στο αίτημα να ταιριάζει με το πραγματικό μέγεθος του αυθαίρετου μηνύματος που έπρεπε να αναμεταδοθεί. Έτσι, στην απάντηση του Heartbeat περιλαμβανόταν το αυθαίρετο μήνυμα, αλλά και ένα πλήθος δεδομένων που βρίσκονταν αποθηκευμένα «μετά» από αυτό το αυθαίρετο μήνυμα.

Μια τέτοια επίθεση έπληττε την Εμπιστευτικότητα του συστήματος, καθώς τα δεδομένα που διέρρεαν μπορεί να αφορούσαν άλλους χρήστες και να ήταν εξαιρετικά ευαίσθητα (λχ υλικό για κλειδιά κρυπτογράφησης ή προστατευμένα δεδομένα σε μη κρυπτογραφημένη μορφή). Το χειρότερο με αυτή την επίθεση είναι πως δεν άφηνε κανένα ίχνος στα αρχεία καταγραφής.

(Heartbleed Bug, 2020; Carvalho, DeMott, Ford, & Wheeler, 2014)

Εικόνα 6: Λογότυπο που δημιουργήθηκε στα πλαίσια ενημέρωσης για το Heartbleed [πηγή: (Heartbleed Bug, 2020)]

4.2.2 Επίθεση SQL Injection

Οι σύγχρονες εφαρμογές χαρακτηρίζονται από τον όγκο των δεδομένων που διαχειρίζονται· γεγονός που καθιστά επιτακτική την αξιοποίηση μιας Βάσης Δεδομένων για την αποθήκευσή τους. Στην πράξη υπάρχουν τρεις οντότητες που αλληλοεπιδρούν στις σύγχρονες εφαρμογές: ο χρήστης, η εφαρμογή και η Βάση Δεδομένων, με την εφαρμογή να λειτουργεί σε μεγάλο βαθμό ως διεπαφή (interface) που ρυθμίζει την επικοινωνία των άλλων δύο. Οι απλοί χρήστες δεν αλληλοεπιδρούν ποτέ απευθείας με τη Βάση Δεδομένων· μόνο η εφαρμογή επικοινωνεί μαζί της, χρησιμοποιώντας τη γλώσσα SQL για την επικοινωνία. Σε αυτό το μοντέλο οι χρήστες συχνά παρέχουν κάποια δεδομένα που πρέπει η εφαρμογή να διαβιβάσει, αφού πρώτα τα ερμηνεύσει σωστά, στη Βάση Δεδομένων.

Το πρόβλημα σε αυτό το μοντέλο προκύπτει όταν η εφαρμογή δεν ερμηνεύει τα δεδομένα ή δεν επιβάλει περιορισμούς στα δεδομένα που επιτρέπεται να υποβάλει ο χρήστης. Σε μια **επίθεση SQL Injection** ένας κακόβουλος χρήστης επιχειρεί να «μιλήσει» απευθείας με τη Βάση Δεδομένων, γράφοντας σε κάποιο πεδίο εισόδου δεδομένων της εφαρμογής κακόβουλο κώδικα SQL, με σκοπό να διαβιβαστεί μαζί με τον έγκυρο κώδικα SQL που υποβάλει η εφαρμογή· απώτερος στόχος είναι να επιτευχθεί η εκτέλεση του κακόβουλου κώδικα από το σύστημα της Βάσης Δεδομένων.

Η επίθεση αυτή είναι μια εξειδικευμένη περίπτωση της τεχνικής Code Injection και βασίζεται σε μια ευπάθεια της εφαρμογής, καθώς δεν κάνει *έλεγχο της εγκυρότητας* των δεδομένων που δέχεται ως είσοδο από τον χρήστη πριν τα μεταβιβάσει στη Βάση Δεδομένων. Συνήθως, στόχος μιας τέτοιας επίθεσης είναι η αποκάλυψη πληροφοριών στις οποίες δεν έχει πρόσβαση ο κακόβουλος χρήστης, όμως, ανάλογα με το επίπεδο ασφάλειας που τηρείται στη Βάση Δεδομένων, ίσως καταφέρει να αλλοιώσει ή ακόμα και να καταστρέψει δεδομένα (Shahriar & Zulkernine, 2012; Biju, Gopal, & Prakash, 2019). Τέλος, η τεχνική SQL Injection είναι δυνατόν να εφαρμοστεί στα πλαίσια μιας επίθεσης cryptojacking (Yasin, 2019), που θα εξηγηθεί παρακάτω.

4.2.3 Επίθεση Cross Site Scripting (XSS)

Μια **επίθεση Cross Site Scripting** (**XSS** [79]) εφαρμόζει την τεχνική Code Injection για να εισαγάγει κακόβουλο κώδικα [80] σε έμπιστους ιστότοπους, που στη συνέχεια εκτελείται από τους περιηγητές του Παγκόσμιου Ιστού (web browsers) στην πλευρά των θυμάτων.

Διακρίνονται τρεις κύριοι τύποι XSS:

- Στο *stored XSS* ή *persistent XSS* ο επιτιθέμενος έχει καταφέρει να εκμεταλλευτεί μια ευπάθεια στον web server, ώστε να αποθηκεύσει τον κακόβουλο κώδικα απευθείας σε αυτόν. Στον προσβεβλημένο web server, ο κακόβουλος κώδικας εισάγεται (στατικά ή δυναμικά) στις ιστοσελίδες του έμπιστου ιστότοπου, που παραδίδονται στους περιηγητές των θυμάτων, που με τη σειρά τους εκτελούν τον κακόβουλο κώδικα.
- Στο *reflected XSS* η επίθεση εκκινείται από το ίδιο θύμα, περνάει στον έμπιστο ιστότοπο και στη συνέχεια «ανακλάται» ξανά στον περιηγητή του θύματος. Αρχικά πρέπει το θύμα να κάνει κλικ σε έναν *κακόβουλο υπερσύνδεσμο* [81] ενός phishing email ή απλά να περιηγηθεί σε έναν κακόβουλο ιστότοπο [82]. Αυτό έχει ως αποτέλεσμα να δημιουργηθεί ένα αίτημα προς τον έμπιστο ιστότοπο, που θα παραλάβει τον κακόβουλο κώδικα και, λόγω ευπάθειας, θα τον συμπεριλάβει, χωρίς να κάνει έλεγχο εγκυρότητας, στην ιστοσελίδα που θα αποστείλει στο θύμα. Στον περιηγητή του θύματος ο κακόβουλος κώδικας θα εκτελεστεί κανονικά, αφού προέρχεται από έναν έμπιστο ιστότοπο.
- Στο *DOM based XSS* η ευπάθεια εντοπίζεται στον περιηγητή του θύματος και όχι στον server όπως στις προηγούμενες περιπτώσεις. Και σε αυτή την περίπτωση, το θύμα πρέπει να κάνει κλικ σε έναν κακόβουλο υπερσύνδεσμο· ο server αποστέλλει μόνο την ιστοσελίδα, όμως λόγω της ευπάθειας στην πλευρά του περιηγητή, συμπεριλαμβάνεται και εκτελείται και ο κακόβουλος κώδικας.

(OWASP, χ.χ.; Shahriar & Zulkernine, 2012; Biju, Gopal, & Prakash, 2019)

Η επίθεση XSS είναι μια εξαιρετικά διαδεδομένη μέθοδος που χρησιμοποιούν οι επιτιθέμενοι για την αποκάλυψη προσωπικών και ευαίσθητων πληροφοριών (Ullah, και συν., 2018). Για παράδειγμα, ένα session id είναι ευαίσθητη πληροφορία, αφού θα μπορούσε να επιτρέψει στον επιτιθέμενο να «υποδυθεί» το θύμα σε μια άλλη επικοινωνία με τον ίδιο server. Άλλες περιπτώσεις ευαίσθητων πληροφοριών είναι το username, το ΑΦΜ κ.α. που αποθηκεύονται απευθείας μέσα στα cookies.

---

[79] Το *XSS* ως συντομογραφία του *Cross Site Scripting* μπορεί να εξηγηθεί επειδή το γράμμα Χ στην αγγλική γλώσσα παραπέμπει στον σταυρό (cross).

[80] Στις επιθέσεις XSS, ο κακόβουλος κώδικας είναι συνήθως κώδικας JavaScript. Στη μορφή stored XSS ίσως να είναι και κώδικας στη γλώσσα που χρησιμοποιεί ο web server για να δημιουργήσει τις ιστοσελίδες.

[81] Στις επιθέσεις XSS και CSRF, ο όρος **κακόβουλος υπερσύνδεσμος** χρησιμοποιείται για έναν υπερσύνδεσμο που έχει ενσωματωμένο κακόβουλο κώδικα (συνήθως κώδικα JavaScript).

[82] Στην εργασία αυτή, με τον όρο **κακόβουλος ιστότοπος** νοείται είτε ένας ιστότοπος που δημιουργήθηκε εξαρχής με κακόβουλο σκοπό, είτε ένας κατά τα άλλα αγαθός ιστότοπος που προσβλήθηκε με κακόβουλο κώδικα μετά από κάποια επίθεση. Σε κάθε περίπτωση πάντως, ο ιστότοπος αυτός αξιοποιείται για την τέλεση κάποιας επίθεσης στους ανυποψίαστους χρήστες του.

4.2.4 Επίθεση Cross Site Request Forgery (CSRF)

Μια **επίθεση Cross Site Request Forgery *(*CSRF*)*** εφαρμόζει την τεχνική Code Injection για να προκαλέσει μη εξουσιοδοτημένες λειτουργίες σε έναν έμπιστο ιστότοπο, που φαινομενικά προέρχονται από αιτήματα ενός αυθεντικοποιημένου [83] χρήστη. (Shahriar & Zulkernine, 2012; Synopsys, χ.χ.)

Σε αντίθεση με την επίθεση XSS, που εκμεταλλεύεται την εμπιστοσύνη που έχει ένας περιηγητής στον ιστότοπο, η επίθεση CSRF εκμεταλλεύεται την εμπιστοσύνη που έχει ένας ιστότοπος σε έναν αυθεντικοποιημένο χρήστη. (Synopsys, χ.χ.) Για να γίνει κατανοητή η επίθεση CSRF, πρέπει να έχουμε κατά νου πως οι σύγχρονοι ιστότοποι, μετά την αυθεντικοποίηση ενός χρήστη, αποθηκεύουν στα cookies του περιηγητή κάποια μοναδικά δεδομένα συνόδου (λ.χ. session id). Πολλές φορές, αυτά τα δεδομένα συνόδου αποστέλλονται μαζί με άλλα δεδομένα που παρέχει ο χρήστης για να πιστοποιήσουν ότι είναι πράγματι αυτός που αιτείται την εκτέλεση μιας λειτουργίας στον ιστότοπο.

Η επίθεση εξελίσσεται όταν ο χρήστης – θύμα έχει συνδεθεί στον έμπιστο ιστότοπο και ενώ η σύνδεση είναι ακόμα ενεργή, παράλληλα κάνει κλικ σε έναν κακόβουλο υπερσύνδεσμο (που βρίσκεται είτε σε ένα phishing email, είτε σε κάποιον κακόβουλο ιστότοπο)· αυτό έχει ως αποτέλεσμα ο κακόβουλος κώδικας να υποκλέψει τα μοναδικά δεδομένα συνόδου από τα cookies του έμπιστου ιστότοπου και να τα χρησιμοποιήσει για να υποβάλει αυτόματα και χωρίς να το αντιληφθεί ο χρήστης, ένα αίτημα στον έμπιστο ιστότοπο. (Shahriar & Zulkernine, 2012; Synopsys, χ.χ.) [84]

[83] Για τον ορισμό της αυθεντικοποίησης δείτε την ενότητα «*Μηχανισμοί αυθεντικοποίησης*».

[84] Για να γίνει κατανοητή η επίπτωση μιας επίθεσης CSRF, φανταστείτε το εξής σενάριο: είστε συνδεδεμένοι στο web banking σας και ταυτόχρονα περιηγείστε στον Παγκόσμιο Ιστό. Αν περιηγηθείτε σε έναν καλόβουλο και έμπιστο ιστότοπο, που όμως έχει προσβληθεί με κακόβουλο κώδικα CSRF, τότε αυτό είναι αρκετό για να γίνει μεταφορά ενός χρηματικού ποσού από το web banking σε κάποιον λογαριασμό του επιτιθέμενου. (Ευτυχώς, τα σύγχρονα συστήματα web banking έχουν πρόσθετους μηχανισμούς ασφάλειας, οπότε το σενάριο είναι καθαρά υποθετικό).

4.2.5 Επιθέσεις με χρήση κακόβουλου λογισμικού

Όπως προκύπτει από προηγούμενη ενότητα που εξετάστηκε ως διάνυσμα επίθεσης, το κακόβουλο λογισμικό έχει πολλές και διαφορετικές μορφές, που προσφέρουν μια μεγάλη πληθώρα δυνατοτήτων σε έναν επιτιθέμενο. Όντας ένα από τα σημαντικότερα «όπλα» ενός επιτιθέμενου, είναι απολύτως λογικό το κακόβουλο λογισμικό να αξιοποιείται είτε σε στοχευμένες επιθέσεις κατά ενός συγκεκριμένου συστήματος, είτε σε μαζικές, μη στοχευμένες επιθέσεις.

Χωρίς να γίνεται αναφορά σε λεπτομέρειες που έχουν αναλυθεί στην προαναφερόμενη ενότητα, αξίζει να αναφερθούν επιγραμματικά οι τρόποι διάδοσής του και οι κύριοι λόγοι αξιοποίησής του. Ως προς τον τρόπο διάδοσης, κάποια είδη κακόβουλου λογισμικού έχουν τη δυνατότητα να πραγματοποιούν επιθέσεις μέσω του Διαδικτύου με αυτόνομο τρόπο εκμεταλλευόμενα ευπάθειες του λογισμικού του συστήματος. Κάποια μπορεί να χρειάζονται έναν ανθρώπινο παράγοντα – χρήστη του συστήματος που εκούσια ή ακούσια θα τα μεταφέρει με φυσικά μέσα (λ.χ. συσκευές USB) στο σύστημα. Τέλος, κάποια θα περάσουν στο σύστημα ως αποτέλεσμα μιας άλλης επίθεσης (επίθεση phishing, επίθεση drive-by, δημιουργία μιας ευπάθειας από προηγούμενη επίθεση κ.α.).

Ως προς τους απώτερους στόχους μιας επίθεσης με κακόβουλο λογισμικό, αυτοί μπορεί να είναι η αποκάλυψη ή η αλλοίωση πληροφοριών, η καταστροφή πληροφοριών ή πόρων, η παροχή μη εξουσιοδοτημένης πρόσβασης στο σύστημα και η χρήση πόρων για παράνομους σκοπούς.

Αξίζει τέλος να σημειωθεί, πως, στις μέρες μας, μια επίθεση με κακόβουλο λογισμικό είναι ένα μόνο βήμα μιας πιο σύνθετης επίθεσης, που εξελίσσεται σε πολλά στάδια.

4.2.6 Επίθεση cryptojacking

Η **επίθεση cryptojacking** έχει σκοπό τη μη εξουσιοδοτημένη και κρυφή χρήση των υπολογιστικών πόρων ενός συστήματος για την εξόρυξη κρυπτονομισμάτων. Αυτό, όπως εξηγούν οι Tekiner, Acar, Uluagac, Kirda, & Selcuk (2021), μπορεί να επιτευχθεί είτε με τη χρήση εξειδικευμένου κακόβουλου λογισμικού cryptojacking (όπως αναλύθηκε σε προηγούμενη ενότητα), είτε να εκτελεστεί ως κώδικας σε έναν περιηγητή του Παγκόσμιου Ιστού.

Για την εκτέλεση του cryptojacking στους περιηγητές, ένας επιτιθέμενος πρέπει πρώτα διαθέτει έναν κακόβουλο ιστότοπο. Όπως εξηγήθηκε, ο κακόβουλος ιστότοπος δεν είναι απαραίτητο να ανήκει στον επιτιθέμενο· αρκεί να προσβάλει έναν με κακόβουλο κώδικα σε μια προηγούμενη επίθεση. Για αυτό τον σκοπό μπορεί να χρησιμοποιήσει δύο επιθέσεις: ή την επίθεση SQL Injection ή την επίθεση XSS. Και οι δύο θα του επιτρέψουν να εισαγάγει τον κακόβουλο κώδικα cryptojacking μέσα σε κατά τα άλλα αγαθούς ιστότοπους, που στη συνέχεια θα εκτελείται μαζικά στους περιηγητές των ανυποψίαστων χρηστών. (Yasin, 2019)

4.2.7 Επίθεση Drive-By

Η **επίθεση Drive-By** [85] πραγματοποιείται μέσω του Παγκόσμιου Ιστού. Το ιδιαίτερο σε αυτή την επίθεση είναι πως ο χρήστης – θύμα δεν χρειάζεται να κάνει κλικ σε κάποιον κακόβουλο υπερσύνδεσμο, αρκεί να περιηγηθεί σε έναν κακόβουλο ιστότοπο και χωρίς περεταίρω ενέργειά του, το σύστημα προσβάλλεται από κακόβουλο λογισμικό (Shabut, Lwin, & Hossain, 2016) (ενδεικτικά κάποιον δούρειο ίππο, bot ή λογισμικό cryptojacking) ή ο περιηγητής οδηγείται στην εκτέλεση κακόβουλου κώδικα (Biju, Gopal, & Prakash, 2019) στα πλαίσια μιας δευτερεύουσας επίθεσης (ενδεικτικά XSS, CSRF, cryptojacking).

Οι επιθέσεις Drive-By είναι δυνατές εξ αιτίας ευπαθειών στους περιηγητές του Παγκόσμιου Ιστού ή στα plug-in τους ή σε components που λειτουργούν με τους περιηγητές. (Shabut, Lwin, & Hossain, 2016)

[85] Αναφέρεται εναλλακτικά ως επίθεση Drive-By Download ή Drive-By Installation.

4.2.8 Επιθέσεις DoS

Μια **επίθεση DoS** (**Denial of Service – Άρνησης Υπηρεσίας**) έχει σκοπό στερήσει από τους νόμιμους χρήστες του υπολογιστικού συστήματος έναν συγκεκριμένο υπολογιστικό πόρο ή μια συγκεκριμένη υπηρεσία (Simmons, Ellis, Shiva, Dasgupta, & Wu, 2009). Αυτό επιτυγχάνεται είτε προκαλώντας την πλήρη διακοπή της λειτουργίας του συστήματος, είτε επιβραδύνοντας τη λειτουργία του σε τέτοιο βαθμό που να καταστεί άχρηστο (Uma & Padmavathi, 2013). Έχει σκοπό να πλήξει τη διαθεσιμότητά του συστήματος και όχι την εμπιστευτικότητα ή την ακεραιότητά του· συνεπώς, δεν οδηγεί στην αποκάλυψη, στην αλλοίωση ή στην καταστροφή ευαίσθητων δεδομένων, ωστόσο προκαλεί οικονομική ζημία στον οργανισμό στον οποίο ανήκει το υπολογιστικό σύστημα – στόχος (Sinha, Rai, & Bhushan, 2019).

Γενικά, οι επιθέσεις DoS καταναλώνουν το εύρος ζώνης (bandwidth) [86] του δικτύου και χρόνο εκτέλεσης της CPU του συστήματος – στόχου (Vanitha, UMA, & Mahidhar, 2017). Σε αυτές τις επιθέσεις αξιοποιείται το *IP spoofing*, δηλαδή ο επιτιθέμενος αποστέλλει πακέτα με ψευδείς (*spoofed*) διευθύνσεις IP, είτε γιατί απλά επιθυμεί να αποκρύψει τη δική του, είτε γιατί έτσι μπορεί να στοχοποιήσει τον κόμβο – στόχο. Μεγάλο πλεονέκτημα για τους επιτιθέμενους είναι πως είναι τρομερά δύσκολο να διακριθούν τα κακόβουλα αιτήματα από τα νόμιμα αιτήματα προς εξυπηρέτηση και συνεπώς η πρόληψη των επιθέσεων DoS είναι δύσκολή (Biju, Gopal, & Prakash, 2019).

Οι Simmons, Ellis, Shiva, Dasgupta & Wu (2009) διακρίνουν τρία είδη DoS:

- Επιθέσεις *host-based DoS*, στις οποίες στόχος είναι ένας συγκεκριμένος υπολογιστικός κόμβος του συστήματος. Συνήθως, ζητούμενο για τον επιτιθέμενο είναι να καταναλώσει όλους τους πόρους του υπολογιστή αυτού ή να τον θέσει εκτός λειτουργίας [87].
- Επιθέσεις *network-based DoS*, στις οποίες στόχος είναι να καταστεί αδύνατον γίνει χρήση μιας υπηρεσίας του δικτύου· αυτό τυπικά γίνεται «πλημμυρίζοντας» το δίκτυο με πακέτα, γεγονός που καταναλώνει το εύρος ζώνης (bandwidth) του δικτύου και επηρεάζει τη δυνατότητα σύνδεσης των κόμβων στην υπηρεσία.
- Το τρίτο είδος είναι οι επιθέσεις *Distributed DoS*, που, όπως αναφέρουν, αποτελεί την πιο δημοφιλή επιλογή των επιτιθέμενων. Και πράγματι, είναι τέτοια η σημασία τους, που θα αναλυθούν περαιτέρω σε ξεχωριστή ενότητα.

[86] Η κατανάλωση του εύρους ζώνης του δικτύου έχει ως αποτέλεσμα να μειώνεται η αποδοτικότητα όλων των υπηρεσιών και όχι μόνο της υπηρεσίας που αποτελεί στόχο της επίθεσης DoS.

[87] Υπ' αυτή την έννοια, ακόμα και μια εισβολή με κακόβουλο λογισμικό που θα υπερφορτώσει και θα προκαλέσει τον τερματισμό της λειτουργίας του συστήματος, εξελίσσεται σε επίθεση DoS. Ή ακόμα, ένα λογισμικό ransomware που κρυπτογραφεί αρχεία με σημαντικές πληροφορίες έχει ως αποτέλεσμα DoS.

Παρόμοια, σε εφαρμογές του Παγκόσμιου Ιστού θα μπορούσε να αξιοποιηθεί μια επίθεση SQL Injection για να επιτευχθεί το DoS. Για παράδειγμα, οι κακόβουλες εντολές SQL θα μπορούσαν να ορίζουν την καταστροφή ενός σημαντικού πίνακα της Βάσης Δεδομένων της εφαρμογής, με αποτέλεσμα η εφαρμογή δυσλειτουργεί και παύει να είναι διαθέσιμη προς χρήση.

4.2.8.1 Επιθέσεις DDoS

Οι **επιθέσεις *DDoS*** (**Distributed Denial of Service – Κατανεμημένης Άρνησης Υπηρεσίας**) αποτελούν την πιο προηγμένη μορφή επιθέσεων DoS. Σε αυτή τη μορφή, πολλά διαφορετικά συστήματα συνεργάζονται, ώστε να πραγματοποιήσουν συγχρονισμένα μια επίθεση DoS σε έναν συγκεκριμένο κόμβο – στόχο (Sinha, Rai, & Bhushan, 2019) [88]. Ο Lee (2023) εξηγεί με παραστατικό τρόπο το αποτέλεσμα της επίθεσης, παρομοιάζοντάς την σαν ένα πλήθος ατόμων που δεν είναι πελάτες και συνωστίζονται κλείνοντας την είσοδο του καταστήματος, διαταράσσοντας έτσι την ορθή εμπορική δραστηριότητά του. Και επισημαίνει πως επειδή η επίθεση προέρχεται από πολλές διαφορετικές πηγές είναι αδύνατον να περιοριστεί με το μπλοκάρισμα μίας από αυτές [89].

Γενικά, όπως αναφέρει ο Medlock (2023) οι επιθέσεις DDoS μπορούν να χωριστούν σε τρεις κατηγορίες:

- Τις *volume based* επιθέσεις, δηλαδή αυτές που απλά προσπαθούν να κατακλύσουν τον στόχο με ένα τεράστιο όγκο ανεπιθύμητης [90] εισερχόμενης κυκλοφορίας. Οι επιθέσεις αυτές καταναλώνουν τόσο εύρος ζώνης εισερχόμενης κυκλοφορίας, όσο και εύρος ζώνης εξερχόμενης κυκλοφορίας (Vanitha, UMA, & Mahidhar, 2017) και μετριούνται σε Gigabits per second (Gbps) (Imperva, 2023). Από αυτή την κατηγορία ξεχωρίζουν:
    - Η επίθεση *ICMP flood* ή επίθεση *ping*, στην οποία ο επιτιθέμενος αποστέλλει διαρκώς αιτήματα ICMP echo (ping) [91] στον κόμβο – στόχο, αναγκάζοντάς τον να δεσμεύσει πόρους για να αποστείλει μια απάντηση. (Vanitha, UMA, & Mahidhar, 2017)
    - Η επίθεση *UDP flood*, στην οποία αποστέλλονται πακέτα UDP σε διάφορα ports του κόμβου – στόχου, αναγκάζοντάς τον να εξετάσει αν υπάρχει κάποια υπηρεσία σε αυτό το port και να στείλει ένα πακέτο ICMP Destination Unreachable. (Vanitha, UMA, & Mahidhar, 2017)

[88] Με άλλα λόγια, οποιαδήποτε επίθεση DoS κατά ενός δικτύου, αν εκτελεστεί ταυτόχρονα από πολλά συστήματα, μετατρέπεται σε DDoS. Για να γίνει κατανοητό πως μπορεί να λειτουργήσει ο συντονισμός, δείτε την ενότητα «*Botnet, zombies και bots*» και την υποσημείωση 72 για το botnet Mirai.

[89] Το μπλοκάρισμα της εισερχόμενης κυκλοφορίας από μια συγκεκριμένη πηγή είναι ένα αντίμετρο που θα μπορούσε να εφαρμοστεί αποτελεσματικά για την αντιμετώπιση μιας απλής επίθεση DoS.

[90] Ανεπιθύμητη γιατί δεν πρόκειται για αιτήματα νόμιμων χρηστών, αλλά για πλαστά αιτήματα που μόνο σκοπό έχουν να καταναλώσουν το εύρος ζώνης (bandwidth) του δικτύου και πόρους του συστήματος.

[91] Η αποστολή ενός αιτήματος και η αναμονή μίας απάντησης είναι ο κανονικός τρόπος λειτουργίας, που σκοπό έχει να ελεγχθεί αν υπάρχει συνδεσιμότητα μεταξύ των δύο πλευρών.

- Τις *protocol based* επιθέσεις, δηλαδή αυτές που εκμεταλλεύονται εγγενείς αδυναμίες των πρωτοκόλλων επικοινωνίας των δικτύων. Οι επιθέσεις αυτές μετριούνται σε packets per second (pps) (Imperva, 2023). Από αυτή την κατηγορία ξεχωρίζουν:
    - Η επίθεση *SYN flood*, στην οποία ο επιτιθέμενος εκμεταλλεύεται τον τρόπο λειτουργίας του μηχανισμού 3-way-handshake του πρωτοκόλλου TCP. Η επίθεση πραγματοποιείται αποστέλλοντας διαρκώς πακέτα SYN στον κόμβο – στόχο για να αιτηθεί το άνοιγμα νέων συνδέσεων TCP, χωρίς να ολοκληρώνει ποτέ κάποια σύνδεση. Αποτέλεσμα είναι πως ο κόμβος – στόχος δεσμεύει διαρκώς πόρους, για συνδέσεις που δεν ολοκληρώνονται, έως ότου να εξαντληθούν οι διαθέσιμοι πόροι του και να μην μπορεί να εξυπηρετήσει νόμιμες συνδέσεις. (Sinha, Rai, & Bhushan, 2019)
    - Η επίθεση *Ping of Death*, στην οποία ο επιτιθέμενος αποστέλλει παραποιημένα πακέτα ICMP, με μέγεθος μεγαλύτερο του επιτρεπτού. Αστοχίες στον σχεδιασμό κάποιων συστημάτων που δεν μπορούν να χειριστούν τέτοια πακέτα δημιουργούν μια ευπάθεια, στην οποία το σύστημα δυσλειτουργεί καθώς αναζητά δεδομένα που ξεπερνούν το μέγιστο επιτρεπτό όριο του buffer. (Mallikarjunan, K.Muthupriya, & Shalinie, 2016)
- Τις *application based* επιθέσεις ή επιθέσεις *DDoS επιπέδου 7*, δηλαδή αυτές που τυπικά προσπαθούν να εξαντλήσουν τους πόρους (μνήμη, χρόνο επεξεργαστή κτλ.) των πρωτοκόλλων του Επιπέδου Εφαρμογών του μοντέλου OSI. Γενικά. καταναλώνουν λιγότερο εύρος ζώνης (bandwidth) σε σχέση με τα άλλα είδη επιθέσεων (Vanitha, UMA, & Mahidhar, 2017). Οι επιθέσεις αυτές μετριούνται σε requests per second (rps) (Imperva, 2023). Από αυτή την κατηγορία ξεχωρίζουν:
    - Η επίθεση *HTTP flood*, που είναι η πιο κοινή επίθεση που μπορεί να πραγματοποιηθεί για το πρωτόκολλο HTTP. Εκτελείται πολύ εύκολα με τη διαρκή αποστολή αιτημάτων για ιστοσελίδες ενός ιστότοπου. Συνήθης στόχος είναι η αρχική σελίδα του ιστότοπου, όμως κάτι τέτοιο είναι σχετικά εύκολο να αναγνωριστεί και να μπλοκαριστεί. Έτσι, σε μια συχνή παραλλαγή χρησιμοποιούνται τυχαίες διευθύνσεις του ιστότοπου. (Praseed & Thilagam, 2019) Αποτέλεσμά της είναι να εξαντλούνται οι διαθέσιμες συνδέσεις του ιστότοπου και να μην μπορεί να εξυπηρετήσει νόμιμους χρήστες.

- Η επίθεση *Slowloris* [92] είναι το πιο γνωστό παράδειγμα των λεγόμενων *αργών επιθέσεων DDoS*. Αυτή η επίθεση πραγματοποιείται με το άνοιγμα HTTP συνδέσεων και την αποστολή αιτημάτων GET σε πολύ μικρά κομμάτια (fragments), που συχνά αποτελούνται από μία επικεφαλίδα HTTP τη φορά. Κάθε φορά που αποστέλλεται ένα κομμάτι, η αποστολή σταματάει. Και λίγο πριν λήξει το χρονικό όριο στο οποίο ο web server κλείνει τη σύνδεση, γίνεται αποστολή ενός νέου κομματιού. Αυτό έχει ως αποτέλεσμα ο web server να συνεχίσει να δεσμεύει πόρους για να κρατάει ανοικτές αυτές τις συνδέσεις, πράγμα που, αργά αλλά σταθερά, οδηγεί στην εξάντληση των διαθέσιμων συνδέσεων και το αναπόφευκτο DoS. Παραλλαγή της επίθεσης Slowloris είναι η επίθεση *R.U.D.Y.* (*aRe yoU Dead Yet?*) η οποία χρησιμοποιεί αιτήματα POST για να πετύχει το ίδιο αποτέλεσμα. (Praseed & Thilagam, 2019)
- Η επίθεση *DNS amplification*, η οποία εκμεταλλεύεται συγκεκριμένα χαρακτηριστικά του συστήματος DNS για να εκτελέσει παράλληλα μια επίθεση *ανάκλασης* (*reflection)* [93] και μια επίθεση *ενίσχυσης* (*amplification)* [94] ώστε να μεγιστοποιήσει το «χτύπημα» στον κόμβο – στόχο. Η επίθεση ξεκινά με τον επιτιθέμενο να αποστέλλει διαρκώς ειδικά διαμορφωμένα [95] ερωτήματα DNS με πλαστή IP πηγής (αυτή του κόμβου – στόχου), σε πολλούς ανοικτούς DNS server. Ανταποκρινόμενος στο ερώτημα που έλαβε, καθένας από αυτούς τους server παράγει μια «ενισχυμένη» απάντησή την οποία «ανακλά» στον κόμβο – στόχο (Imperva, 2023). Έτσι, τόσο ο κόμβος όσο και το δίκτυο κατακλύζονται από ένα τεράστιο πλήθος μεγάλων σε μέγεθος πακέτων· γεγονός που τελικά οδηγεί σε DoS.

[92] Το όνομα της επίθεσης αυτής προέρχεται από το είδος *slow loris* (*αργός λόρις*), ενός πρωτεύοντος γνωστού για τη βραδύτητά του.

[93] Σε μια **επίθεση ανάκλασης (reflection)**, ο επιτιθέμενος παραπλανά έναν αγαθό τρίτο υπολογιστικό κόμβο, ώστε να αποστείλει πακέτα στον κόμβο – στόχο· πακέτα που ενώ προέρχονται από έναν αγαθό κόμβο, παραμένουν ανεπιθύμητα. Η επίθεση πραγματοποιείται με τον επιτιθέμενο να αποστέλλει πλαστά αιτήματα με ψευδή (spoofed) IP πηγής (αυτή του κόμβου – στόχου) στο τρίτο μέρος, το οποίο μη γνωρίζοντας την απάτη, αποστέλλει («*ανακλά*») τις απαντήσεις στον κόμβο – στόχο. (Vanitha, UMA, & Mahidhar, 2017)

[94] Σε μια **επίθεση ενίσχυσης (amplification)**, ο επιτιθέμενος καταφέρνει, από ένα δικό του πακέτο μικρού μεγέθους, να αποστείλει ένα πακέτο με σημαντικά μεγαλύτερο μέγεθος στον στόχο του ή από ένα δικό του πακέτο να παράγονται και να παραδίδονται πολλά πακέτα στον στόχο του. (Vanitha, UMA, & Mahidhar, 2017)

[95] Στο σύστημα DNS είναι δυνατόν ένα απλό ερώτημα να παραγάγει μια δυσανάλογα μεγάλη απάντηση· ιδιότητα που μπορεί να εκμεταλλευτεί ο επιτιθέμενος για να επιτύχει την *ενίσχυση*.

4.2.8.2 Σύγκριση DoS / DDoS με άλλες επιθέσεις

Η Ασφάλεια Πληροφοριών εξετάζει πολλά και διαφορετικά είδη επιθέσεων. Οι επιθέσεις DoS / DDoS έχουν αναδειχθεί στη μεγαλύτερη απειλή κατά της ασφάλειας (Vanitha, UMA, & Mahidhar, 2017) και επιφέρουν μεγάλο οικονομικό κόστος στους οργανισμούς που γίνονται στόχος τους (Sinha, Rai, & Bhushan, 2019). Αυτού του είδους οι επιθέσεις είναι πάντοτε στοχευμένες και σε αντίθεση με άλλα είδη στοχευμένων επιθέσεων δεν προϋποθέτουν μια επιτυχημένη εισβολή στο σύστημα.

Το γεγονός αυτό κάνει τη δουλειά των επιτιθέμενων πολύ πιο απλή. Δεν χρειάζεται να εξαπατήσουν χρήστες του συστήματος με μεθόδους κοινωνικής μηχανικής, που απαιτούν υπομονή και αρκετή επιμονή πολλές φορές, ενώ το αποτέλεσμα δεν είναι ποτέ εγγυημένο. Επίσης, οι επιθέσεις αυτές δεν απαιτούν από τους επιτιθέμενους να αναζητούν διαρκώς νέες ευπάθειες λογισμικού για να εκμεταλλευτούν. Οποιοδήποτε σύστημα έχει σύνδεση στο Διαδίκτυο, είναι ένας υποψήφιος στόχος για μια επίθεση DoS / DDoS, αν το επιθυμήσει ένας επιτιθέμενος. Η ύπαρξη εργαλείων αυτοματοποίησης των επιθέσεων, όπως τα HOIC (High Orbit Ion Cannon), LOIC (Low Orbit Ion Cannon), RUDY, κ.α. που είναι διαθέσιμα για οποιονδήποτε στον Παγκόσμιο Ιστό (Vanitha, UMA, & Mahidhar, 2017), κάνει τα πράγματα πολύ πιο απλά και ταυτόχρονα επικίνδυνα.

Αξιοσημείωτο είναι πως πλέον οι επιθέσεις DDoS προσφέρονται ως υπηρεσία προς πληρωμή. Κακόβουλοι παράγοντες χτίζουν έναν «στρατό» συστημάτων zombie, και στο τέλος διαθέτουν το botnet που δημιούργησαν σε οποιονδήποτε (άτομο, ομάδα ή ακόμα και κράτος) καταβάλει το ανάλογο αντίτιμο για να πραγματοποιήσει μια επίθεση σε έναν στόχο που θα του υποδειχθεί.

4.2.9 Επιθέσεις MitM

Μια **επίθεση MitM** (**Man-in-the-Middle**) πραγματοποιείται όταν ένας επιτιθέμενος καταφέρει να βρεθεί ως ενδιάμεσος στην επικοινωνία μεταξύ δύο νόμιμων πλευρών (Sinha, Rai, & Bhushan, 2019) (χωρίς, φυσικά, να το γνωρίζουν οι δύο πλευρές). Από τεχνικής απόψεως, για αυτή την επίθεση απαιτείται ο επιτιθέμενος να δημιουργεί μια ανεξάρτητη σύνδεση με κάθε πλευρά και να αναμεταδίδει τα μηνύματα που ανταλλάσσουν μεταξύ τους, κάνοντάς τους να πιστεύουν πως επικοινωνούν υπό μυστικότητα. (Uma & Padmavathi, 2013)

Η επίθεση προσφέρει δύο εναλλακτικές στον επιτιθέμενο: σε μια παθητική επίθεση, ο επιτιθέμενος λειτουργεί απλά ως *ωτακουστής*· παραλαμβάνει και καταγράφει τα μηνύματα που διακινούνται εκατέρωθεν, αλλά τα αναμεταδίδει αυτούσια. Σε μια ενεργητική επίθεση, ο επιτιθέμενος δεν παρακολουθεί απλά την επικοινωνία των δύο μερών, αλλά *υποδύεται* τον αντίθετο ρόλο σε κάθε πλευρά· αυτό του επιτρέπει να αλλοιώνει την επικοινωνία μεταξύ τους, τροποποιώντας τα μηνύματα πριν τα μεταδώσει ή και καταστρέφοντάς τα. (Sinha, Rai, & Bhushan, 2019) Στην εκδοχή της παθητικής επίθεσης, η επίθεση MitM πλήττει μόνο την εμπιστευτικότητα των πληροφοριών, αλλά στην ενεργητική της εκδοχή πλήττει επιπλέον και την ακεραιότητα και τη διαθεσιμότητά τους. (Conti, Dragoni, & Lesyk, 2016)

Οι Conti, Dragoni & Lesyk (2016), εξηγούν ότι μια επίθεση MitM δεν περιορίζεται σε περιπτώσεις όπου ο επιτιθέμενος βρίσκεται στο ίδιο δίκτυο με κάποιο από τα θύματα, αφού υπάρχουν μορφές της επίθεσης όπου ο επιτιθέμενος μπορεί να δράσει πλήρως απομακρυσμένα. Αξίζει τέλος να αναφερθεί, πως η πιο διαδεδομένη μορφή επίθεσης MitM υλοποιείται με τη μορφή κακόβουλου λογισμικού [96] που προσβάλλει τους περιηγητές του Παγκόσμιου Ιστού και παρεμβαίνει στην επικοινωνία τους με ιστότοπους (εστιάζοντας σε αυτούς που έχουν ευαίσθητο περιεχόμενο λ.χ. ιστότοπους web banking). (Yasar, 2024)

[96] Τη μορφή αυτή την ονομάζει *Man-in-the-Browser* (*MitB*) (Yasar, 2024).

4.3 Μοντέλα μελέτης επιθέσεων

Αν αναλογιστούμε όσα αναφέρθηκαν στην προηγούμενη ενότητα, προκύπτει ότι υπάρχουν πάρα πολλές μορφές κυβερνοεπιθέσεων, άλλες απλές και άλλες πιο σύνθετες· με τις τελευταίες να έχουν συχνά ως προαπαιτούμενο το αποτέλεσμα κάποιων προηγούμενων επιθέσεων για να μπορέσουν να εξελιχθούν. Στο κομμάτι αυτό λοιπόν, η έρευνα των μελετητών μας δείχνει πως οι επιθέσεις πραγματοποιούνται σταδιακά. Και μάλιστα είναι δυνατόν να αναγνωριστούν κάποιες φάσεις που μπορούν να θεωρηθούν κοινές για πολλές κυβερνοεπιθέσεις.

Για παράδειγμα, οι Ullah και συν. (2018) αναγνωρίζουν σε επιθέσεις που στοχεύουν στην αποκάλυψη πληροφοριών τα εξής στάδια:

- *Έρευνα* (*research*), όπου ο επιτιθέμενος αναζητά αδυναμίες που μπορεί να εκμεταλλευτεί για να πετύχει τους σκοπούς του.
- *Επίθεση* (*attack*), όπου ο επιτιθέμενος αξιοποιεί την τεχνική που έχει αναγνωρίσει ως καταλληλότερη για να εκμεταλλευτεί την αδυναμία του συστήματος και να αποκτήσει πρόσβαση.
- *Λαθραία εξαγωγή* (*exfiltration*), όπου ο επιτιθέμενος έχει αποκτήσει πρόσβαση στις επιθυμητές πληροφορίες και τις εξάγει από το σύστημα.

Παρομοίως, το Βρετανικό Εθνικό Κέντρο για την Κυβερνοασφάλεια (NCSC, 2015) αναγνωρίζει και αναφέρει τέσσερα κοινά στάδια [97] σε πολλές επιθέσεις:

- *Επόπτευση* (*survey*), στο οποίο ο επιτιθέμενος αξιοποιεί όλα τα διαθέσιμα μέσα για να συγκεντρώσει και στη συνέχεια να αναλύσει πληροφορίες που θα του επιτρέψουν να αναγνωρίσει ευπάθειες στο σύστημα.
- *Παράδοση* (*delivery*), στο οποίο ο επιτιθέμενος προσπαθεί να εκμεταλλευτεί μια ή περισσότερες ευπάθειες που έχει αναγνωρίσει στο προηγούμενο στάδιο.
- *Παραβίαση* (*breach*), στο οποίο ο επιτιθέμενος αξιοποιεί έναν μηχανισμό εκμετάλλευσης μιας ευπάθειας για να αποκτήσει πρόσβαση στο σύστημα.
- *Επίδραση* (*affect*), στο οποίο ο επιτιθέμενος, αφού έχει διεισδύσει στο σύστημα, πραγματοποιεί τις ενέργειες που ήθελε εξ αρχής να εκτελέσει.

[97] Οι ελληνικοί όροι αποδίδονται με ελεύθερη μετάφραση των αντίστοιχων αγγλικών όρων.

Το 2011 η Lockheed Martin εξέδωσε ένα έγγραφο σχετικά με την άμυνα των Πληροφοριακών Συστημάτων, όπου παρουσίασε το λεγόμενο **μοντέλο Cyber Kill Chain** (Hutchins, Cloppert, & Amin, 2011) [98]. Το μοντέλο αυτό αποτελεί ένα από τα βασικότερα έργα για την κατανόηση του τρόπου δράσης των επιτιθέμενων καθώς και των σταδίων διεξαγωγής μιας σύνθετης επίθεσης. Βασική ιδέα σε αυτό το μοντέλο είναι πως η επίθεση εξελίσσεται σειριακά και σε φάσεις που αλληλεξαρτώνται, όπως οι κρίκοι σε μια αλυσίδα. Συνεπώς, αν ο αμυνόμενος καταφέρει να «*σπάσει την αλυσίδα*» (δηλαδή να εξουδετερώσει τη δράση του επιτιθέμενου) στα πρώτα στάδια, τότε η επίθεση αποτυγχάνει. (Xcitium, 2024)

Οι επτά αλληλένδετες φάσεις μιας σύνθετης επίθεσης στο μοντέλο Cyber Kill Chain είναι οι εξής [99]:

- *Αναγνώριση* (*reconnaissance*), όπου ο επιτιθέμενος πραγματοποιεί μια έρευνα για την αναγνώριση και την επιλογή των πιθανών στόχων του.
- *Μετατροπή σε όπλο* (*weaponization*), όπου επιτιθέμενος δημιουργεί κακόβουλο λογισμικό ή κακόβουλο κώδικα για να εκμεταλλευτεί τις πιθανές ευπάθειες ασφαλείας που αναγνώρισε.
- *Παράδοση* (*delivery*), όπου ο επιτιθέμενος βρίσκει έναν τρόπο για να μεταδώσει το «όπλο» του στο σύστημα – στόχο.
- *Εκμετάλλευση* (*exploitation*), όπου το «όπλο» ενεργοποιείται, αφού έχει πλέον εισβάλει στο σύστημα.
- *Εγκατάσταση* (*installation*), όπου ο επιτιθέμενος προσπαθεί να αποκτήσει μια μόνιμη παρουσία μέσα στο σύστημα.
- *Διοίκηση και Έλεγχος* (*Command and Control*), όπου το προσβεβλημένο σύστημα, τυπικά ειδοποιεί έναν controller server, ώστε να εγκαθιδρυθεί ένα κανάλι επικοινωνίας με τον επιτιθέμενο.
- *Δράσεις επί των Σκοπών* (*actions on objectives*), όπου ο επιτιθέμενος έχει εξασφαλίσει την παρουσία του στο υπολογιστικό σύστημα και μπορεί να εκτελέσει ενέργειες σχετικές με τους πραγματικούς του σκοπούς. Αυτές οι ενέργειες μπορεί να πλήττουν την εμπιστευτικότητα (λ.χ. λαθραία εξαγωγή πληροφοριών), την ακεραιότητα (λ.χ. αλλοίωση δεδομένων) ή τη διαθεσιμότητα (λ.χ. κρυπτογράφηση αρχείων) του συστήματος. Εναλλακτικά, ίσως το συγκεκριμένο υπολογιστικό σύστημα να είναι το εφαλτήριο για να *κινηθεί παράπλευρα* [100] σε άλλους πιο σημαντικούς κόμβους του δικτύου, στους οποίους δεν μπορούσε να αποκτήσει απευθείας πρόσβαση.

(Hutchins, Cloppert, & Amin, 2011; Bahrami, και συν., 2019)

[98] Η εταιρεία Lockheed Martin ειδικεύεται στον τομέα των οπλικών συστημάτων και εισήγαγε την έννοια *kill chain* (σε ελεύθερη μετάφραση: *φονική αλυσίδα*) των στρατιωτικών επιθέσεων στις κυβερνοεπιθέσεις.

[99] Κάποιες περιγραφές είναι ελαφρώς παραλλαγμένες σε σχέση με το αρχικό έγγραφο, για να περιλαμβάνουν και πιο σύγχρονες περιπτώσεις, με βάση το (Bahrami, και συν., 2019).

[100] **Παράπλευρη κίνηση** (**lateral movement**) ονομάζεται μια στρατηγική, όπου ο επιτιθέμενος αποκτά πρόσβαση σε έναν κόμβο, επειδή σε δεύτερο βαθμό θα του επιτρέψει να αποκτήσει πρόσβαση σε κάποιον άλλο κόμβο του δικτύου με μεγαλύτερη «αξία».

Ένα άλλο πολύ σημαντικό έργο για την κατανόηση των κυβερνοεπιθέσεων είναι το **μοντέλο ATT&CK** [101] του οργανισμού MITRE. Ο ίδιος ο οργανισμός περιγράφει το μοντέλο αυτό ως μια παγκόσμια προσβάσιμη βάση γνώσης (knowledge base) για τις τακτικές, τις τεχνικές και τις διαδικασίες (*Tactics, Techniques & Procedures – TTPs*) που χρησιμοποιούν οι αντίπαλοι, βασιζόμενο σε πραγματικές παρατηρήσεις. Σκοπός του είναι να λειτουργήσει ως θεμέλιο για την ανάπτυξη μοντέλων απειλών και μεθοδολογιών, ενώ το ίδιο το μοντέλο είναι ανοικτό και διαθέσιμο προς όλους χωρίς χρέωση. (MITRE, 2024)

Το μοντέλο ATT&CK οργανώνεται καταρχάς σε τρεις πίνακες (matrices) – *Enterprise*, *Mobile* και *ICS* – ανάλογα με το περιβάλλον εξετάζεται. Καθένας από τους πίνακες αποτελείται από τακτικές επίθεσης [102], που εξηγούν γενικά τους πιθανούς στόχους που θέλει να επιτύχει ένας αντίπαλος. Παράλληλα, καθεμιά από αυτές τις τακτικές περιλαμβάνει ένα πλήθος τεχνικών, που σκοπό έχουν να αναλύσουν εκτενέστερα τον τρόπο δράσης ενός αντιπάλου, σε σχέση πάντοτε με τον στόχο που επιθυμεί να επιτύχει [103]. Επειδή το πλήθος των τεχνικών είναι πραγματικά μεγάλο και εν πολλοίς ξεφεύγει από τα όρια της παρούσας εργασίας [104], παρακάτω αναφέρονται μόνο οι τακτικές που εντάσσονται στο μοντέλο. Αυτές είναι:

- Η τακτική της *αναγνώρισης* (*reconnaissance*), που περιλαμβάνει τεχνικές συγκέντρωσης πληροφοριών για την προετοιμασία μιας επίθεσης κατά ενός στόχου.
- Η τακτική της *ανάπτυξης πόρων (resource development*), που περιλαμβάνει τεχνικές που θα επιτρέψουν σε έναν αντίπαλο να δημιουργήσει, να αγοράσει ή ακόμα και να κλέψει πόρους που θα τον βοηθήσουν στην επίθεση.
- Η τακτική της *αρχικής πρόσβασης* (*initial access*), που περιλαμβάνει τεχνικές για την αξιοποίηση διάφορων διανυσμάτων επίθεσης, οι οποίες θα επιτρέψουν στον αντίπαλο να αποκτήσει πρόσβαση στο σύστημα.
- Η τακτική της *εκτέλεσης* (*execution*), που περιλαμβάνει τεχνικές που θα επιτρέψουν σε έναν αντίπαλο να εκτελέσει τον κακόβουλο κώδικά του σε ένα τοπικό ή ένα απομακρυσμένο σύστημα.
- Η τακτική της *επιμονής* (*persistence*), που περιλαμβάνει τεχνικές που θα επιτρέψουν σε έναν αντίπαλο να διατηρήσει την πρόσβασή του στο σύστημα, ακόμα και αν γίνουν γεγονότα όπως αλλαγή κωδικών πρόσβασης, επανεκκίνηση του συστήματος κτλ.

---

[101] ATT&CK: αρκτικόλεξο της φράσης *Adversarial Tactics, Techniques and Common Knowledge*, που στα ελληνικά μπορεί να αποδοθεί ως *Αντίπαλες Τακτικές, Τεχνικές και Κοινή Γνώση*.

Ο οργανισμός MITRE χρησιμοποιεί τον όρο **αντίπαλος** (**adversary**) για να αναφερθεί σε κάθε είδους κακόβουλο παράγοντα, εξωτερικό ή εσωτερικό, επιτιθέμενο ή εισβολέα.

[102] Ο πίνακας Enterprise περιέχει και τις δεκατέσσερις τακτικές, ενώ οι πίνακες Mobile και ICS περιέχουν από δώδεκα τακτικές· δεν περιλαμβάνουν τις τακτικές της αναγνώρισης και της ανάπτυξης πόρων.

[103] Ή, όπως αναφέρει ο ίδιος ο οργανισμός MITRE, οι τακτικές αναπαριστούν το «*γιατί*», ενώ οι τεχνικές το «*πώς*» στη δράση ενός αντιπάλου. (MITRE, 2024)

[104] Διαβάζοντας για τις τακτικές του μοντέλου ATT&CK, ο αναγνώστης θα συνειδητοποιήσει ότι σε έναν μεγάλο βαθμό πολλές τεχνικές που απαρτίζουν μια τακτική έχουν ήδη εξηγηθεί σε προηγούμενες ενότητες και δεν υπάρχει λόγος να αναλυθούν εκ νέου.

- Η τακτική της *κλιμάκωσης προνομιών* (*privilege escalation*), που περιλαμβάνει τεχνικές ώστε ένας αντίπαλος να αποκτήσει δικαιώματα υψηλότερου επιπέδου σε ένα σύστημα. Συχνά, είναι ευκολότερο για τους αντιπάλους να αποκτήσουν πρόσβαση με χαμηλό επίπεδο δικαιωμάτων σε ένα σύστημα· αυτό τους επιτρέπει να το εξερευνούν, όμως για να επιτύχουν τους απώτερους σκοπούς τους χρειάζονται υψηλότερο επίπεδο δικαιωμάτων.
- Η τακτική της *αποφυγής της άμυνας* (*defense evasion*), που περιλαμβάνει τεχνικές που θα αποτρέψουν την ανίχνευση της δράσης ενός αντιπάλου ή ακόμα και μιας επιτυχημένης εισβολής.
- Η τακτική της *πρόσβασης σε διαπιστευτήρια* (*credential access*), που περιλαμβάνει τεχνικές που αξιοποιούν οι αντίπαλοι για να κλέψουν διαπιστευτήρια, όπως στοιχεία σύνδεσης σε λογαριασμούς, usernames και passwords.
- Η τακτική της *ανακάλυψης* (*discovery*), που περιλαμβάνει τεχνικές για την εξερεύνηση και τελικά την κατανόηση της εσωτερικής δομής ενός συστήματος.
- Η τακτική της *παράπλευρης κίνησης* (*lateral movement*), που θα επέτρεπε σε έναν αντίπαλο να μετακινείται από τον έναν υπολογιστικό κόμβο ή λογαριασμό χρήστη στον επόμενο μέχρι να αποκτήσει πρόσβαση στον επιθυμητό πόρο.
- Η τακτική της *συλλογής* (*collection*), που επιτρέπει σε έναν αντίπαλο να συγκεντρώσει πληροφορίες, ώστε στη συνέχεια να κλέψει δεδομένα.
- Η τακτική της *διοίκησης και ελέγχου* (*command and control*), που περιλαμβάνει τεχνικές με τις οποίες ένας αντίπαλος θα μπορούσε να επικοινωνεί με ένα σύστημα και να το ελέγχει απομακρυσμένα.
- Η τακτική της *λαθραίας εξαγωγής* (*exfiltration*), που περιλαμβάνει τεχνικές για την κλοπή δεδομένων από ένα σύστημα. Μετά τη συλλογή των δεδομένων, οι αντίπαλοι εφαρμόζουν διάφορες τεχνικές για να τα βγάλουν από το σύστημα χωρίς να γίνει αντιληπτή η δράση αυτή.
- Η τακτική του *αντικτύπου* (*impact*), που περιλαμβάνει διάφορες τεχνικές κατά της διαθεσιμότητας ή της ακεραιότητας των πληροφοριών (λ.χ. καταστροφή ή αλλοίωση δεδομένων) του συστήματος. Επίσης, είναι δυνατόν να εφαρμοστούν τέτοιες τεχνικές ως κάλυψη για μια παραβίαση της εμπιστευτικότητας.

Το μεγάλο πλεονέκτημα του μοντέλου ATT&CK είναι πως εμβαθύνει στον τρόπο δράσης των αντιπάλων, κάτι που δεν κάνει το μοντέλο Cyber Kill Chain. Επίσης, δεν ακολουθεί το κεντρικό αξίωμα του μοντέλου Cyber Kill Chain, που θεωρεί πως η αντιμετώπιση μιας από τις πρώτες φάσεις οδηγεί συνολικά σε αποτυχία της επίθεσης. Αντιθέτως, το μοντέλο ATT&CK παρουσιάζει τις τακτικές μόνο σε λογική σειρά, ενώ αναλύει τους στόχους που προσπαθούν να επιτύχουν οι αντίπαλοι (Xcitium, 2024), γεγονός που επιτρέπει την προσαρμογή σε περισσότερα σενάρια χρήσης.

5

Κυβερνοάμυνα

5.1 Βασικές έννοιες και αρχές

Στο προηγούμενο κεφάλαιο εξετάστηκαν οι σημαντικότερες πτυχές των κυβερνοεπιθέσεων· τα διανύσματα επίθεσης, τα διαφορετικά είδη τους, αλλά και τα κυριότερα μοντέλα μελέτης τους· όλα αυτά που μας προσφέρουν μια καλή κατανόηση των τεχνικών και τακτικών που εφαρμόζουν οι επιτιθέμενοι για να επιτύχουν τους σκοπούς τους. Έχοντας κατανοήσει σε μεγάλο βαθμό το *modus operandi* των επιτιθέμενων, ακολουθεί το επόμενο βήμα σε αυτή την εργασία, που είναι η μελέτη των μέτρων που μπορεί να ληφθούν για την προστασία ενός συστήματος από τις κυβερνοεπιθέσεις.

Αρχικά, πρέπει να τονιστεί ότι το ζήτημα της προστασίας κάθε είδους αγαθού στον κυβερνοχώρο βρίσκεται στο επίκεντρο της Ασφάλειας Πληροφοριών και συχνά αναφέρεται ως **κυβερνοάμυνα** (**cyber-defense**). Η Rouse (2024) περιγράφει την κυβερνοάμυνα ως έναν μηχανισμό για την προστασία κρίσιμων υποδομών του συστήματος, τη διασφάλιση των πληροφοριών, καθώς και την απόκριση σε διάφορες ενέργειες (που αποτελούν απειλή για το σύστημα). Και συνεχίζει εξηγώντας πως εστιάζει στην *πρόληψη*, την *ανίχνευση* και την έγκαιρη *απόκριση* σε επιθέσεις ή απειλές ώστε να μην θιγούν υποδομές και πληροφορίες.

Η Denning (2014) παρουσιάζει διάφορες οπτικές της κυβερνοάμυνας αντλώντας έμπνευση από στρατιωτικά αμυντικά συστήματα. Έτσι λοιπόν, αναφέρει ότι η **ενεργητική κυβερνοάμυνα** είναι «*μια άμεση αμυντική δράση για την καταστροφή, την εξουδετέρωση ή τη μείωση της αποτελεσματικότητας των απειλών κατά φίλιων δυνάμεων και πόρων*». Και η **παθητική κυβερνοάμυνα** είναι «*κάθε μέτρο, που δεν ανήκει στην ενεργητική κυβερνοάμυνα, και λαμβάνεται με σκοπό να μειώσει την αποτελεσματικότητα των απειλών κατά φίλιων δυνάμεων και πόρων*».

Στη συνέχεια, διευκρινίζει ότι η ενεργητική κυβερνοάμυνα αφορά άμεσες δράσεις κατά συγκεκριμένων απειλών, ενώ η παθητική κυβερνοάμυνα προσπαθεί να καταστήσει τους πόρους του συστήματος πιο ανθεκτικούς σε επιθέσεις. Στα μέτρα ενεργητικής κυβερνοάμυνας περιλαμβάνει: τους μηχανισμούς *αυθεντικοποίησης*, τους μηχανισμούς *ελέγχου πρόσβασης*, τη χρήση συστημάτων *honeypot* [105], *αντεπιθέσεις* [106], τη χρήση συστημάτων *antimalware*, *firewall*, *IDS* και *IPS*. Και στα μέτρα παθητικής κυβερνοάμυνας αναφέρει: την αξιοποίηση *κρυπτογραφίας* [107] και *στεγανογραφίας* [108], τη δημιουργία εφεδρικών αντιγράφων (backup) για την επαναφορά (recovery) δεδομένων σε περίπτωση απώλειας, τον ασφαλή σχεδιασμό και την επαλήθευση ασφάλειας, την παρακολούθηση και την διαχείριση των ρυθμίσεων του συστήματος, την *εκτίμηση κινδύνου* [109] για το σύστημα και φυσικά την εκπαίδευση των χρηστών σε θέματα κυβερνοασφάλειας. (Denning, 2014)

Συνεχίζοντας, η ίδια συγγραφέας παρουσιάζει τη διάκριση της ενεργητικής κυβερνοάμυνας σε *αυτόματη* (*automatic*) και *μη αυτόματη* (*manual*). Στην αυτόματη ενεργητική κυβερνοάμυνα δεν υπάρχει ανάγκη ανθρώπινης παρέμβασης, ενώ στη μη αυτόματη ενεργητική κυβερνοάμυνα απαιτείται, είτε η άμεση εκκίνησή της από κάποιον ανθρώπινο παράγοντα ή έστω μια ενέργεια επιβεβαίωσής πριν την ενεργοποίησή της. (Denning, 2014)

Τέλος, τονίζει πως κάποια μέτρα ενεργητικής κυβερνοάμυνας εγείρουν ηθικά και νομικά ζητήματα, οπότε θα πρέπει να ικανοποιούνται πρόσθετα κάποιες αρχές για την εφαρμογή τους. Καταρχάς, θα πρέπει να εξετάζεται αν υπάρχει η απαραίτητη εξουσιοδότηση εφαρμογής, να λαμβάνεται υπόψη η επίδραση σε αμέτοχα τρίτα μέρη, να πληρείται το κριτήριο της αναγκαιότητας χρήσης και το κριτήριο της αναλογικότητας, να μην παραβιάζονται τα δικαιώματα και οι ελευθερίες των πολιτών, ενώ σε κάποιο βαθμό θα πρέπει να αναμειγνύονται άνθρωποι στη λήψη αποφάσεων [110]. (Denning, 2014)

---

[105] *Honeypot* ονομάζονται απομονωμένα συστήματα που έχουν σκοπό να παρασύρουν τους επιτιθέμενους μακριά από τα παραγωγικά συστήματα, για να τους θέσουν υπό παρακολούθηση. (Denning, 2014)

[106] Οι *αντεπιθέσεις στον κυβερνοχώρο* (*counter cyber-attacks*) είναι επιθετικές ενέργειες που λαμβάνονται από τα μέρη που επηρεάζονται από μια επίθεση, ώστε ο επιτιθέμενος να αναγκαστεί να πάρει αμυντική στάση και να σταματήσει την επίθεση. (Ferdous, 2022)

[107] Η *κρυπτογραφία* είναι ένα σημαντικό κεφάλαιο στην Ασφάλεια Πληροφοριών, όμως η εκτενής ανάλυσή της είναι εκτός των ορίων της παρούσας εργασίας. Οπότε, περιοριζόμενοι σε μια απλή περιγραφή θα λέγαμε τα εξής: Ο όρος κρυπτογραφία αναφέρεται σε μεθόδους για τη μετατροπή κατανοητών πληροφοριών σε μη κατανοητές. Η μετατροπή των μη κατανοητών πληροφοριών ξανά στην κατανοητή τους μορφή είναι δυνατή μόνο με τη χρήση ενός μυστικού κλειδιού.

[108] Η *στεγανογραφία* αναφέρεται σε μεθόδους που σκοπό έχουν να αποκρύψουν πληροφορίες μέσα σε άλλες, μη κρυφές πληροφορίες.

[109] Η *εκτίμηση κινδύνου* (*risk assessment*) έχει σκοπό να αναγνωρίσει πιθανές απειλές κατά του συστήματος και να αποτιμήσει τον αντίκτυπό τους.

[110] Τις περισσότερες φορές, αυτές οι αρχές θεωρούνται θέματα με τετριμμένες απαντήσεις, όταν τα μέτρα εφαρμόζονται στο εσωτερικό του συστήματος. Για παράδειγμα, η συγγραφέας αναφέρει πως η εξουσιοδότηση πηγάζει από τον ίδιο τον οργανισμό και δεν αποτελεί πρόβλημα. Το πρόβλημα προκύπτει για μέτρα που έχουν επίδραση στο εξωτερικό του συστήματος. Χαρακτηριστικότερο παράδειγμα που πρέπει όλες αυτές οι αρχές να ληφθούν υπόψη, είναι το μέτρο των αντεπιθέσεων στον κυβερνοχώρο.

Οι Kimura, DeVries & Gordon (2017) προσεγγίζουν την κυβερνοάμυνα από διαφορετική σκοπιά. Στο δικό τους έργο, αναφέρουν ότι η κυβερνοάμυνα δομείται πάνω σε τρεις βασικές συνιστώσες:

- Την *πρόληψη* (*prevention*) [111], που χρησιμοποιεί αυτοματοποιημένους μηχανισμούς που λειτουργούν σε πραγματικό χρόνο για να εμποδίσουν απειλές να αποκτήσουν πρόσβαση στο σύστημα. Είναι το πιο γρήγορο και πιο φθηνό από τα τρία στοιχεία.
- Την *ανίχνευση* (*detection*), που εστιάζει στην αναγνώριση απειλών ώστε να αντιμετωπιστούν είτε με πρόληψη, είτε με απόκριση. Η ανίχνευση με στόχο την πρόληψη βασίζεται στις *υπογραφές* (signature based) [112] των απειλών κατά κύριο λόγο. Οι συγγραφείς επισημαίνουν, πως εξαιτίας της ταχείας εξέλιξης των απειλών, απαιτούνται προηγμένες λειτουργίες, όπως η *συμπεριφοριστική ανάλυση* (*behavioral analytics*), για την παραγωγή επίκαιρων ενδείξεων (indicators) για τους μηχανισμούς πρόληψης.
- Και την *απόκριση* (*response*), που επικεντρώνεται σε απειλές που έχουν καταφέρει να παρακάμψουν τους μηχανισμούς πρόληψης και να εισβάλουν στο σύστημα. Η απόκριση σε περιστατικά (incident response) είναι ζωτικής σημασίας για τον τερματισμό της δραστηριότητας των επιτιθέμενων, την απομείωση (mitigation) και την αποκατάσταση (remediation) της ζημίας και την πρόληψη παρόμοιων μελλοντικών περιστατικών.

Και για τα τρία στοιχεία της κυβερνοάμυνας απαραίτητη είναι η αξιοποίηση **αντίμετρων** (**countermeasures**) [113] (Kimura, DeVries, & Gordon, 2017), δηλαδή χρήση συσκευών, διαδικασιών ή τεχνικών που σκοπό έχουν την ελάττωση της ευπάθειας του συστήματος (NIST, χ.χ.).

[111] Η λέξη *prevention* αποδίδεται στα ελληνικά είτε ως *πρόληψη*, είτε ως *αποτροπή*. Στα πλαίσια της παρούσας εργασίας οι δύο λέξεις και κάθε παράγωγό τους θα χρησιμοποιούνται ως συνώνυμες.

[112] Στην ενότητα για το λογισμικό antimalware θα εξηγηθεί περεταίρω η έννοια της υπογραφής μιας απειλής.

[113] Δεδομένου του πεδίου εφαρμογής τους, ονομάζονται και *αντίμετρα στον κυβερνοχώρο* (*cyber countermeasures*). Μέσα στο γενικότερο πλαίσιο της Ασφάλειας Πληροφοριών αποκαλούνται και *μέτρα κυβερνοασφάλειας*.

5.2 Εργαλεία οργάνωσης της κυβερνοάμυνας

Όπως αναφέρθηκε στο κεφάλαιο για τις κυβερνοεπιθέσεις, ένα από τα διανύσματα που αξιοποιούν οι επιτιθέμενοι είναι η εκμετάλλευση ευπαθειών. Λογικό λοιπόν είναι ένας από τους πρώτους στόχους της κυβερνοάμυνας να είναι η εξουδετέρωσή τους. Δεδομένου όμως ότι οι περισσότερες ευπάθειες πηγάζουν από τον σχεδιασμό και την υλοποίηση των συστημάτων, ο εντοπισμός και η εξάλειψή τους δεν είναι απλή υπόθεση. Στα πλαίσια αυτά, ο οργανισμός MITRE διατηρεί τη *λίστα CWE* [114], που καταδεικνύει μια σειρά από *αδυναμίες* [115] στο υλικό (hardware) και το λογισμικό, που θα μπορούσαν να μετατραπούν σε ευπάθειες. (MITRE, 2023) Η κατανόηση των αδυναμιών [116] από τις ομάδες σχεδιασμού και ανάπτυξης συστημάτων βοηθάει στην αξιοποίηση των κατάλληλων πρακτικών που θα τις εξαλείψουν.

Επιπρόσθετα, ο οργανισμός MITRE αναπτύσσει και τη *λίστα CVE* [117], στην οποία καταγράφει ευπάθειες που έχουν γνωστοποιηθεί δημόσια. Οι δυο τους διαφέρουν, γιατί η λίστα CWE οργανώνει και κατηγοριοποιεί αδυναμίες που δυνητικά οδηγούν σε ευπάθειες, ενώ η λίστα CVE περιέχει υπαρκτές ευπάθειες για συγκεκριμένα προϊόντα (εφαρμογές, πλατφόρμες, συστήματα κτλ.). Υπάρχει πολύ ισχυρή σχέση μεταξύ των δύο, καθώς κάποια αδυναμία στη λίστα CWE είναι το αίτιο που έχει ως αποτέλεσμα μια συγκεκριμένη ευπάθεια στη λίστα CVE.

Ένα άλλο έργο, η βάση δεδομένων *NVD* [118] του οργανισμού NIST, αναλαμβάνει να εμπλουτίσει κάθε εγγραφή CVE με επιπλέον πληροφορίες. Μία πληροφορία είναι η συσχέτιση της εγγραφής CVE με έναν τύπο CWE· αυτό υποδεικνύει στην εκάστοτε ομάδα ανάπτυξης του συστήματος, ποια ήταν η αστοχία στη σχεδίαση ή την υλοποίηση, ώστε να τη διορθώσει (πιθανότατα με τη μορφή μιας ενημέρωσης ασφαλείας). Μια δεύτερη, πολύ σημαντική πληροφορία είναι η κρισιμότητα της ευπάθειας με βάση την κλίμακα βαθμονόμησης *CVSS* [119]. Αυτή η κρισιμότητα καθορίζεται από τον αντίκτυπο που θα έχει η ευπάθεια στο σύστημα και επισημαίνει πόσο σημαντική είναι η επιδιόρθωσή της. (MITRE, n.d.)

Με τον τρόπο αυτό, πηγές πληροφόρησης, όπως η βάση δεδομένων NVD και η λίστα CVE, μπορούν να αξιοποιηθούν για την ενίσχυση της άμυνας ενός συστήματος. Ωστόσο, περιορίζονται μόνο σε γνωστές απειλές που προέρχονται από ευπάθειες, ενώ δεν μπορούν να προσφέρουν επαρκή προστασία έναντι μιας άγνωστης ευπάθειας (zero-day) ή χρήσης κάποιου άλλου διανύσματος από έναν επιτιθέμενο. Για τον λόγο αυτό είναι απαραίτητη η εξερεύνηση περεταίρω αντιμέτρων για κάθε τύπο επίθεσης.

---

[114] *CWE*: αρκτικόλεξο του όρου Common Weakness Enumeration.

[115] Ο MITRE ορίζει την *αδυναμία* ως μια κατάσταση στο υλικό (hardware), το firmware, το λογισμικό ή ένα συστατικό μέρος μιας υπηρεσίας που θα μπορούσε να μετατραπεί σε ευπάθεια. (MITRE, 2023)

[116] Χαρακτηριστικό παράδειγμα αδυναμίας που καταγράφεται στη λίστα είναι η CWE-20: Improper Input Validation, που μπορεί να γίνει η αιτία για κάποια επίθεση buffer overflow, injection ή XSS. Άλλο παράδειγμα είναι η CWE-352: Cross-Site Request Forgery, που μπορεί να οδηγήσει σε μια επίθεση CSRF.

[117] *CVE*: αρκτικόλεξο του όρου Common Vulnerabilities and Exposures.

[118] *NVD*: αρκτικόλεξο του όρου National Vulnerability Database.

[119] *CVSS*: αρκτικόλεξο του όρου Common Vulnerability Scoring System.

Αναγνωρίζοντας την αξία που έχουν τα αντίμετρα κατά των κυβερνοεπιθέσεων, ο οργανισμός MITRE ξεκίνησε την ανάπτυξη του **πλαισίου D3FEND** [120]. Στην απλούστερη μορφή του, λειτουργεί ως μια βάση γνώσης (knowledge base), που καθορίζει έναν κατάλογο αμυντικών αντίμετρων για την αντιμετώπιση γνωστών επιθετικών τεχνικών. Όμως, σε αυτό το μοντέλο, σκοπός δεν είναι να παρουσιάζεται απλά το «*τι*» υποτίθεται ότι επιλύει κάθε προτεινόμενο αντίμετρο, αλλά επιπλέον να καθορίζεται το «*πώς*» διευθετούνται οι απειλές από άποψη μηχανικής, αλλά και υπό ποιες προϋποθέσεις αναμένεται να λειτουργήσει σωστά. (MITRE, 2023)

Το έργο αυτό έχει παρόμοια οργάνωση [121] με αυτή του μοντέλου ATT&CK, αλλά από την αμυντική οπτική των πραγμάτων. Αναπτύσσεται ως συμπληρωματικό του ATT&CK, καθώς χαρτογραφεί σχέσεις μεταξύ των τακτικών, των τεχνικών και των διαδικασιών των αντιπάλων, με αμυντικά αντίμετρα, επιτρέποντας, σε έναν βαθμό, την ανάπτυξη μιας αμυντικής στρατηγικής που σχετίζεται άμεσα με τον γνωστό τρόπο δράσης των αντιπάλων. Και ενώ το ATT&CK περιέχει κάποιες στοιχειώδεις συμβουλές για την ελάττωση του αντικτύπου των επιθέσεων, το D3FEND παρέχει μια πιο οργανωμένη και τυποποιημένη προσέγγιση, που παράλληλα επιτρέπει τον σχεδιασμό μιας μακροχρόνιας στρατηγικής για την παρακολούθηση, την ανίχνευση και την απόκριση σε κυβερνοεπιθέσεις. (BlackBerry, χ.χ.) [122]

Όπως ακριβώς και το μοντέλο ATT&CK, το πλαίσιο D3FEND οργανώνεται ιεραρχικά σε τακτικές και τεχνικές, που περιγράφουν μεθόδους για την επίτευξη των επιθυμητών στόχων της άμυνας, παρέχοντας άμεσες αναφορές σε σχετικά πρότυπα, εργαλεία ή ακόμα και πατέντες του τομέα των Πληροφοριακών Συστημάτων.

Στην τελευταία του έκδοση, το πλαίσιο αποτελείται από επτά τακτικές:

- Η *τακτική της μοντελοποίησης* (*model*) χρησιμοποιείται για να εφαρμοστεί η μηχανική ασφαλείας, η ανάλυση ευπαθειών, η ανάλυση απειλών και η ανάλυση κινδύνου σε ένα σύστημα. Αυτά επιτυγχάνονται μέσω της κατανόησης του συστήματος, των λειτουργιών του, των παραγόντων που δρουν σε αυτό, καθώς και των σχέσεων και των αλληλεπιδράσεων μεταξύ όλων αυτών. Χωρίζεται σε τέσσερις κατηγορίες τεχνικών: την *καταγραφή αγαθών* (*asset inventory*), τη *χαρτογράφηση του δικτύου* (*network mapping*), τη *χαρτογράφηση της λειτουργικής δραστηριότητας* (*operational activity mapping*) και τη *χαρτογράφηση του συστήματος* (*system mapping*).

[120] D3FEND: αρκτικόλεξο της φράσης *Detection, Denial and Disruption Framework Empowering Network Defense*, που στα ελληνικά σημαίνει *Πλαίσιο Ανίχνευσης, Άρνησης και Διακοπής που Ενδυναμώνει την Άμυνα του Δικτύου*.

[121] Το D3FEND οργανώνεται σε έναν μοναδικό πίνακα, σε αντίθεση με το ATT&CK που οργανώνεται σε τρεις.

[122] Αξίζει να επισημανθεί πως το πλαίσιο D3FEND είναι σχετικά νέο – η δοκιμαστική του έκδοση έγινε διαθέσιμη τον Ιούλιο του 2021 και τη στιγμή της συγγραφής της παρούσας εργασίας βρίσκεται στην έκδοση 0.14.0. Συνεπώς, το έργο είναι ακόμα σε φάση ωρίμανσης και έχει ακόμα κενά.

Καθώς το πλαίσιο D3FEND εξελίσσεται και τα κομμάτια του συσχετίζονται με αυτά του ATT&CK, μετατρέπεται σε έναν γράφο γνώσης (knowledge graph), όπως το αποκαλεί ο MITRE, που επιτρέπει σε έναν αμυνόμενο την υποβολή ερωτημάτων για την εύρεση και την παρουσίαση των αντίμετρων που συσχετίζονται με γνωστές τακτικές, τεχνικές και διαδικασίες των αντίπαλων. (MITRE, 2023).

- Η *τακτική σκλήρυνσης* (*harden*) χρησιμοποιείται για να κάνει πιο δύσκολη την εκμετάλλευση του συστήματος και γενικά πραγματοποιείται πριν το σύστημα να τεθεί σε επιχειρησιακή λειτουργία. Περιλαμβάνει τις τεχνικές που εντάσσονται στις ακόλουθες κατηγορίες: τη *σκλήρυνση εφαρμογών* (*application hardening*), τη *σκλήρυνση διαπιστευτηρίων* (*credential hardening*), τη *σκλήρυνση μηνυμάτων* (*message hardening*) και τη *σκλήρυνση πλατφόρμας* (*platform hardening*).
- Η *τακτική ανίχνευσης* (*detect*) χρησιμοποιείται για να αναγνωρίσει μια πιθανή πρόσβαση ενός αντίπαλου ή γενικά μη εξουσιοδοτημένη δραστηριότητα σε ένα δίκτυο. Περιλαμβάνει τις τεχνικές: της *ανάλυσης αρχείων* (*file analysis*), της *ανάλυσης αναγνωριστικών* (*identifier analysis*), της *ανάλυσης μηνυμάτων* (*message analysis*), της *ανάλυσης της κυκλοφορίας δικτύου* (*network traffic analysis*), της *παρακολούθησης της πλατφόρμας* (*platform monitoring*), της *ανάλυσης των διεργασιών* (*process analysis*), και της *ανάλυσης της συμπεριφοράς των χρηστών* (*user behavior analysis*).
- Η *τακτική απομόνωσης* (*isolate*) χρησιμοποιείται για να δημιουργήσει φυσικά ή λογικά εμπόδια που μειώνουν τη δυνατότητα των αντιπάλων να αποκτήσουν περεταίρω πρόσβαση στο σύστημα. Σε αυτή την τακτική υπάρχουν δύο κατηγορίες τεχνικών: η *απομόνωση εκτέλεσης* (*execution isolation*) και η *απομόνωση δικτύου* (*network isolation*).
- Η *τακτική εξαπάτησης* (*deceive*) χρησιμοποιείται για να δελεάσει αντίπαλους και να τους παρασύρει σε ένα περιβάλλον, όπου η δράση τους μπορεί να ελεγχθεί ή να παρακολουθηθεί. Και σε αυτή την τακτική αναφέρονται δύο κατηγορίες τεχνικών: η τεχνική του *περιβάλλοντος δολώματος* (*decoy environment*) και η τεχνική του *αντικειμένου δολώματος* (*decoy object*).
- Η *τακτική έξωσης* (*evict*) χρησιμοποιείται για να απομακρύνει έναν αντίπαλο από το σύστημα. Οι τεχνικές είναι: η *έξωση διαπιστευτηρίων* (*credential eviction*), η *έξωση αρχείων* (*file eviction*) και η *έξωση διεργασιών* (*process eviction*).
- Η *τακτική επαναφοράς* (*restore*) χρησιμοποιείται για να επιστρέψει το σύστημα σε μια καλύτερη κατάσταση. Αυτή η τακτική περιλαμβάνει: την *επαναφορά της πρόσβασης* (*restore access*) και την *επαναφορά ενός αντικειμένου* (*restore object*).

5.3 Βασικοί μηχανισμοί και συστήματα κυβερνοάμυνας

Έχοντας μελετήσει τις βάσεις πάνω στις οποίες δομείται ο σχεδιασμός της κυβερνοάμυνας, καλό θα ήταν να εξεταστούν με λίγες λεπτομέρειες και κάποιοι από τους πιο βασικούς μηχανισμούς και συστήματα της κυβερνοάμυνας· χωρίς τους οποίους δεν υφίσταται η Ασφάλεια Πληροφοριών.

5.3.1 Μηχανισμοί αυθεντικοποίησης

Ο όρος **αυθεντικοποίηση** (**authentication**) [123] αναφέρεται σε μια διαδικασία που επιβεβαιώνει πως ένας χρήστης του συστήματος είναι πράγματι αυτός που υποστηρίζει ότι είναι. (Okta, 2023)

Οι Cárdenas, Roosta, Taban, & Sastry (2008) αναφέρουν ότι οι μηχανισμοί αυθεντικοποίησης χρηστών μπορούν να χωριστούν σε τρεις κατηγορίες:

- Την *αυθεντικοποίηση με βάση τη γνώση* (*knowledge based authentication*), που χαρακτηρίζεται από τη φράση «*τι γνωρίζεις*» (*what you know*) και βασίζεται στη μυστικότητα. Σε έναν τέτοιο μηχανισμό, ο χρήστης καλείται να επιβεβαιώσει ποιος είναι δείχνοντας ότι γνωρίζει ένα μυστικό, όπως ένα password, την απάντηση σε μια ερώτηση ασφαλείας, ένα PIN κ.α..
- Την *αυθεντικοποίηση με βάση ένα αντικείμενο* (*object based authentication*) [124], που χαρακτηρίζεται από τη φράση «*τι έχεις*» (*what you have*) και βασίζεται στην κατοχή ενός μοναδικού αντικειμένου. Σε έναν τέτοιο μηχανισμό, ο χρήστης καλείται να επιβεβαιώσει ποιος είναι χρησιμοποιώντας μια προσωπική συσκευή, όπως ένα USB token, ένα smartphone κ.α.
- Την *αυθεντικοποίηση με βάση την ταυτότητα* (*ID based authentication*), που χαρακτηρίζεται από τη φράση «*ποιος είσαι*» (*who you are*) και βασίζεται στη μοναδικότητα ενός ατόμου. Σε έναν τέτοιο μηχανισμό, ο χρήστης καλείται να επιβεβαιώσει ποιος είναι μέσω κάποιων μοναδικών του χαρακτηριστικών, όπως τα βιομετρικά του στοιχεία, όπως τα δακτυλικά αποτυπώματα ή η ίριδα των ματιών.

Γενικά, ο απλούστερος και ο πιο κοινός μηχανισμός αυθεντικοποίησης είναι αυτός που βασίζεται στη χρήση password, όμως, όπως αναφέρθηκε ήδη, υπάρχουν οι λεγόμενες *επιθέσεις πρόσβασης* που σκοπό έχουν να επιτρέψουν στους επιτιθέμενους να ξεπεράσουν τον σκόπελο της αυθεντικοποίησης και να εισβάλουν στο σύστημα. Αυτές, είτε στοχεύουν τα password των χρηστών (π.χ. με μια *επίθεση λεξιλογίου*), είτε τους ίδιους τους χρήστες (π.χ. με κάποια *επίθεση phishing* με σκοπό την αποκάλυψη του password). Υπάρχουν μέτρα που αξιοποιούνται για την ενίσχυση της ασφάλειας (όπως πολιτικές για το μήκος του password ή την ανανέωσή του κ.α.), όμως ο μηχανισμός αυθεντικοποίησης με password παραμένει ο ευκολότερος στόχος για τους επιτιθέμενους [125].

[123] Εναλλακτικά, ο όρος αποδίδεται στα ελληνικά και ως *πιστοποίηση* ή *ταυτοποίηση*.

[124] Άλλοι συγγραφείς αναφέρονται στον ίδιο μηχανισμό με τον όρο *token authentication*.

[125] Διότι αυτά τα μέτρα δεν προσφέρουν καμία προστασία απέναντι σε επιθέσεις phishing.

Μια άλλη τεχνική που αξίζει να αναλυθεί περεταίρω είναι η χρήση του μηχανισμού *2FA* / *MFA* [126]. Πρόκειται για έναν μηχανισμό με σχετικά εύκολη υλοποίηση, που ενισχύει σημαντικά την ασφάλεια της αυθεντικοποίησης. Σε αυτόν τον μηχανισμό απαιτείται από έναν χρήστη να αποδείξει την ταυτότητά του με τουλάχιστον δύο τρόπους. Ο πρώτος τρόπος είναι η χρήση του σωστού password, ενώ ως δεύτερος τρόπος μπορεί να χρησιμοποιηθεί οποιαδήποτε άλλη μέθοδος αυθεντικοποίησης. Στην πράξη ο δεύτερος τρόπος συνήθως είναι η αποστολή ενός κωδικού μιας χρήσης μέσω email ή άλλου μηνύματος. Αυτή η τεχνική προσφέρει σημαντική ενίσχυση της ασφάλειας απέναντι σε *επιθέσεις MitM*, καθώς ένας επιτιθέμενος θα πρέπει να καταφέρει να αποκτήσει περισσότερα διαπιστευτήρια του χρήστη για να αυθεντικοποιηθεί επιτυχώς. (Johnson, 2023)

Πέρα από την αυθεντικοποίηση των χρηστών, είναι πολύ σημαντικό να υπάρχει και αυθεντικοποίηση συσκευών ή ακόμα και διεργασιών που αποστέλλουν διάφορα μηνύματα σε ένα σύστημα. Η αυθεντικοποίηση των μηνυμάτων από τέτοιες οντότητες αξιοποιεί μεθόδους πρόκλησης – απόκρισης (challenge – response), χρονοσφραγίδες μηνυμάτων, καθώς και nonces [127]. (Cárdenas, Roosta, Taban, & Sastry, 2008)

Η ανάγκη εφαρμογής μηχανισμών αυθεντικοποίησης των χρηστών, των διάφορων οντοτήτων του συστήματος, αλλά και των μηνυμάτων που ανταλλάσσονται σε αυτό, αποδεικνύουν ότι η αυθεντικοποίηση έχει μεγάλη σημασία για την Ασφάλεια Πληροφοριών· τόσο μεγάλη που πολλοί ερευνητές θεωρούν ότι έχει μια θέση δίπλα στην τριάδα CIA (δείτε σχετικά την υποσημείωση 18).

[126] Το 2FA προέρχεται από τον όρο two factor authentication (αυθεντικοποίηση δύο παραγόντων), ενώ το MFA είναι αρκτικόλεξο του όρου multi factor authentication (αυθεντικοποίηση πολλαπλών παραγόντων).

[127] Τα *nonces* είναι κρυπτογραφικά ασφαλή, τυχαία αναγνωριστικά που χρησιμοποιούνται σε μια επικοινωνία για να εξασφαλίσουν ότι κάθε μήνυμα είναι μοναδικό και προέρχεται από την αυθεντική πηγή.

5.3.2 Μηχανισμοί εξουσιοδότησης

Ο όρος **εξουσιοδότηση** (**authorization**) αναφέρεται σε μια διαδικασία με την οποία δίνεται η άδεια σε έναν χρήστη του συστήματος να αποκτήσει πρόσβαση σε κάποιον *πόρο* (*resource*) του. (Okta, 2023)

Είναι σημαντικό να σημειωθεί ότι η εξουσιοδότηση είναι άρρηκτα δεμένη με την αυθεντικοποίηση, αφού ένας χρήστης πρέπει πρώτα να αυθεντικοποιηθεί ώστε στη συνέχεια να προχωρήσει η εξουσιοδότησή του. Στο πλαίσιο αυτό είναι συνηθισμένο να εφαρμόζεται ένας ενιαίος μηχανισμός *Διαχείρισης Ταυτότητας και Πρόσβασης* (*Identity and Access Management – IAM*) [128], που σκοπό έχει να ορίσει ποιοι χρήστες έχουν πρόσβαση σε ποιους πόρους. Για το κομμάτι της εξουσιοδότησης είναι απαραίτητο να υλοποιείται ένας μηχανισμός *Ελέγχου Πρόσβασης* (*Access Control*). Τα κυριότερα είδη τέτοιων μηχανισμών είναι:

- Ο *Διακριτικός Έλεγχος Πρόσβασης* (*Discretionary Access Control*), που ακολουθεί την εξής λογική: κάθε πόρος ανήκει σε έναν χρήστη του συστήματος και κάθε χρήστης έχει τον πλήρη έλεγχο των πόρων που του ανήκουν. Ο χρήστης – ιδιοκτήτης ενός πόρου έχει τη δυνατότητα αξιοποιώντας μια *ACL* [129] να αποδίδει συγκεκριμένα δικαιώματα πρόσβασης σε συγκεκριμένους λογαριασμούς χρηστών.
- Ο *Υποχρεωτικός Έλεγχος Πρόσβασης* (*Mandatory Access Control*), που ακολουθεί μια αυστηρή πολιτική ασφαλείας για τον καθορισμό των δικαιωμάτων πρόσβασης στους πόρους. Η πολιτική ορίζεται από τον διαχειριστή του συστήματος και οι χρήστες έχουν μόνο όσα δικαιώματα απορρέουν από αυτή την πολιτική ασφάλειας.
- Ο *Έλεγχος Πρόσβασης Βάσει Ρόλων* (*Role-Based Access Control – RBAC*), που χρησιμοποιείται για να παραχωρηθεί πρόσβαση με βάση κάποιους ρόλους και όχι παραχωρώντας δικαιώματα απευθείας σε λογαριασμούς χρηστών. Σε αυτή τη μέθοδο ορίζονται ρόλοι με προκαθορισμένα δικαιώματα πρόσβασης σε πόρους και σε κάθε χρήστη του συστήματος ανατίθεται ένας ρόλος.
- Ο *Έλεγχος Πρόσβασης Βάσει Χαρακτηριστικών* (*Attribute-Based Access Control – ABAC*) είναι μια πολύπλοκη μέθοδος που αποδίδει τόσο στους χρήστες, όσο και στους πόρους μια πληθώρα χαρακτηριστικών (attributes). Η μέθοδος αυτή ορίζει πως ένας χρήστης έχει πρόσβαση μόνο σε πόρους για τους οποίους έχει τα αντίστοιχα χαρακτηριστικά. Αυτό το είδος ελέγχου πρόσβασης είναι το πιο πολύπλοκο, αλλά παράλληλα επιτρέπει στους διαχειριστές να υλοποιήσουν τους πιο ευέλικτους κανόνες πρόσβασης.

  (Brown, 2023; Cloudflare, χ.χ.)

[128] Ένα σύστημα IAM παρέχει μια σειρά από λειτουργίες που ξεκινούν από τη διαχείριση των λογαριασμών των χρηστών (δημιουργία, ενεργοποίηση, απενεργοποίηση) και φτάνουν ως τον καθορισμό του μηχανισμού αυθεντικοποίησης και του μηχανισμού εξουσιοδότησης.

[129] *ACL*: αρκτικόλεξο του όρου *Access Control List* που στα ελληνικά αποδίδεται ως *Λίστα Ελέγχου Πρόσβασης*.

5.3.3 Συστήματα Firewall

Όπως αναφέρουν οι Kimura, DeVries & Gordon (2017), ο όρος **firewall**, που στα ελληνικά αποδίδεται ως *τείχος προστασίας*, προέρχεται από τα πρώτα ατμοκίνητα οχήματα και αναφερόταν στο τοίχωμα που προστάτευε τον οδηγό από τη φωτιά του θαλάμου καύσης. Ο όρος συνέχισε να χρησιμοποιείται στα οχήματα, αλλά και στα κτίρια, όπου αναφέρεται κυριολεκτικά σε έναν τοίχο που αποτρέπει τη διάδοση της φωτιάς μέσα στο κτίριο. Βάσει της ιστορικής προέλευσης του όρου υπονοείται ξεκάθαρα ότι το firewall έχει σκοπό να προστατεύει τους χρήστες και γενικά το σύστημα από κάτι επικίνδυνο.

Ο τρόπος που ένα firewall προσφέρει προστασία είναι σχετικά απλός: δρώντας σαν θυρωρός εποπτεύει την κυκλοφορία του δικτύου, εισερχόμενη και εξερχόμενη, και φιλτράρει ότι δεν είναι σύμφωνο με την πολιτική ασφαλείας. Για να το επιτύχει αυτό τοποθετείται στην περίμετρο του συστήματος, ως ένας κόμβος που βρίσκεται ανάμεσα στο εσωτερικό δίκτυο και το Διαδίκτυο. (Sinha, Rai, & Bhushan, 2019)

Υλοποιείται είτε με τη μορφή υλικού, είτε με τη μορφή λογισμικού και ακολουθεί τρεις διαφορετικές προσεγγίσεις για το φιλτράρισμα της κυκλοφορίας:

- Η πρώτη εφαρμόζει τη λογική του *φιλτραρίσματος πακέτων* (*packet filtering*). Το firewall λειτουργεί στο επίπεδο Δικτύου και εξετάζει τις διευθύνσεις IP πηγής και προορισμού, καθώς και τα αντίστοιχα port για να αποφασίσει αν κάθε πακέτο είναι σύμφωνο με τους με τους κανόνες που έχουν οριστεί. Αν ακολουθεί τους κανόνες επιτρέπεται να μεταδοθεί, διαφορετικά απορρίπτεται.
- Η δεύτερη εφαρμόζει τη λογική του *φιλτραρίσματος πακέτων με επίγνωση της κατάστασης* (*stateful packet filtering*). Το firewall λειτουργεί στο επίπεδο Μεταφοράς και απαιτεί τη διατήρηση ενός ιστορικού των συνδέσεων. Αντί να εξετάζει κάθε πακέτο με βάση τα ατομικά του στοιχεία, καταγράφει κάποια σημαντικά πεδία του και αποφασίζει για το φιλτράρισμα κρίνοντας αν οι τιμές των πεδίων εξελίσσονται όπως πρέπει στην πάροδο του χρόνου.
- Η τρίτη εφαρμόζει τη λογική του *πληρεξούσιου εφαρμογής* (*application proxy*). Το firewall λειτουργεί στο επίπεδο Εφαρμογής και εξετάζει τα πακέτα από το κατώτερο ως το ανώτερο επίπεδο για να εξασφαλίσει ότι είναι ασφαλή ως προς την προέλευση και ως προς το περιεχόμενο. Το firewall λειτουργεί ως ένας πληρεξούσιος μεσάζων για κάθε σύνδεση ανάμεσα στο εσωτερικό δίκτυο και το Διαδίκτυο και χρησιμοποιεί πολύ προηγμένες τεχνικές εξέτασης των πακέτων για να αποφανθεί αν πρέπει να τα επιτρέψει ή να τα απορρίψει.

(Cárdenas, Roosta, Taban, & Sastry, 2008)

Αυτό που πρέπει να επισημανθεί είναι ότι, παρότι τα firewall παρέχουν προστασία απέναντι σε αρκετά είδη επιθέσεων (λ.χ. αναγνώρισης, DoS, διάδοση σκουληκιών του Διαδικτύου), η τοποθέτησή τους στην περίμετρο του δικτύου τα κάνει ανίκανα να προστατέψουν το σύστημα από εσωτερικές επιθέσεις. (Sinha, Rai, & Bhushan, 2019)

5.3.4 Συστήματα Antimalware

Το **antimalware** [130] είναι λογισμικό που σχεδιάζεται με σκοπό να σαρώνει, να αναγνωρίζει και να εξουδετερώνει το κακόβουλο λογισμικό που έχει προσβάλει ένα σύστημα (Xcitium, 2022). Ιστορικά, ως το πρώτο antimalware θα μπορούσε να θεωρηθεί το *Reaper*, ένα πρόγραμμα που είχε δημιουργηθεί με μοναδικό σκοπό να εντοπίζει και να καταστρέφει το πρόγραμμα *Creeper* [131] (Ask, 2006). Αξίζει επίσης να σημειωθεί ότι τα πρώτα προγράμματα αυτής της κατηγορίας έγιναν γνωστά στο ευρύ κοινό ως *αντι-ιικά προγράμματα* (*antivirus programs*), επειδή την εποχή που πρωτοεμφανίστηκαν στο εμπόριο, οι ιοί ήταν το κυρίαρχο είδος κακόβουλου λογισμικού [132].

Όπως αναλύθηκε στη σχετική ενότητα, στην κατηγορία του κακόβουλου λογισμικού περιλαμβάνεται μια πληθώρα προγραμμάτων με μεγάλη διαφοροποίηση στον τρόπο λειτουργίας, στον τρόπο διάδοσης και φυσικά στον τελικό σκοπό που θέλει να επιτύχουν. Έτσι, και το σύγχρονο λογισμικό antimalware σχεδιάζεται για την αντιμετώπιση πολλών διαφορετικών ειδών κακόβουλου λογισμικού. Δεν αρκείται στην υλοποίηση μίας μόνο τεχνικής ανίχνευσης, καθώς δεν υπάρχει κάποια που να μπορεί να εφαρμοστεί για όλα τα είδη (Ask, 2006).

Η πιο παλιά τεχνική για την ανίχνευση του κακόβουλου λογισμικού είναι η λεγόμενη **ανίχνευση με βάση τις υπογραφές** (**signature based detection**). Η τεχνική αυτή προέρχεται από τα πρώτα αντι-ιικά προγράμματα και περιλαμβάνει τη χρήση των *υπογραφών*. Αρχικά, η υπογραφή ήταν μια συγκεκριμένη ακολουθία δυαδικού κώδικα που είχε ταυτιστεί με έναν ιό, και την οποία το λογισμικό προστασίας αναζητούσε μέσα σε ύποπτα αρχεία για να διαπιστώσει αν είναι μολυσμένα (Kamal, Ali, Alani, & Abdulmajed, 2016). Η τεχνική αυτή αποδείχτηκε αποδοτική, οπότε εφαρμόστηκε και για άλλα είδη κακόβολου λογισμικού. Καθώς το κακόβουλο λογισμικό εξελισσόταν, έτσι εξελίχθηκε και ο μηχανισμός για τον ορισμό των υπογραφών [133], ώστε να λαμβάνει υπόψη επιπλέον χαρακτηριστικά, όπως το όνομα, το μέγεθος, διάφορες συμβολοσειρές που περιέχονται στα αρχεία κ.α. (LinkedIn, 2024)

---

[130] Ο όρος *antimalware* σπάνια αποδίδεται στα ελληνικά· τις λιγοστές φορές που γίνεται κάτι τέτοιο, αποδίδεται μόνο περιγραφικά ως *πρόγραμμα προστασίας από κακόβουλο λογισμικό*.

[131] Με τα σημερινά δεδομένα, το πρόγραμμα Creeper θα ανήκε στην κατηγορία του κακόβουλου λογισμικού, παρότι δεν εκτελούσε κάποια καταστρεπτική ενέργεια· περνούσε στο σύστημα, τύπωνε το μήνυμα «I'M THE CREEPER: CATCH ME IF YOU CAN», μεταδιδόταν στον επόμενο υπολογιστή και διέγραφε τον εαυτό του από τον προηγούμενο (Ask, 2006). Όμως, δεν περνούσε στον υπολογιστή από επιλογή κάποιου χρήστη, αλλά διαδιδόταν και εκτελείτο αυτόνομα, εκμεταλλευόμενο την υλοποίηση του συστήματος (συνεπώς θα κατηγοριοποιούνταν ως σκουλήκι).

[132] Ακόμα και σήμερα, το ευρύ κοινό αποκαλεί «ιό» κάθε πρόγραμμα ή λειτουργία που εκλαμβάνει ως κακόβουλη.

[133] Η δημιουργία μιας υπογραφής ακούγεται απλή υπόθεση, αλλά στην πραγματικότητα δεν είναι. Αρχικά, πρέπει να αποκαλυφθεί ότι ένα λογισμικό είναι κακόβουλο. Τότε μια ομάδα ειδικών ασφαλείας αναλαμβάνει να μελετήσει ένα αντίγραφό του σε ένα ασφαλές σύστημα χρησιμοποιώντας διάφορες τεχνικές reverse engineering. Όταν καταφέρουν να κατανοήσουν πως ακριβώς λειτουργεί, μπορούν να καταλήξουν στην υπογραφή του· εξασφαλίζοντας όμως ότι η υπογραφή θα μπορεί να ταυτοποιεί το κακόβουλο λογισμικό και όχι κάποιο αθώο λογισμικό. Συνθέτοντας αυτή την υπογραφή με άλλα στοιχεία δημιουργούν έναν *ορισμό* (*definition*), που αξιοποιείται από το λογισμικό antimalware για την ανίχνευση του κακόβουλου λογισμικού σε άλλα συστήματα. (LinkedIn, 2024)

Virus Name	*String Pattern (Signature)*
Accom.1280	89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D500 BF50 04CD
Die.448	B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD
Xany.979	8B96 0906 B000 E85C FF8B D5B9 D303 E864 FFC6 8602 0401 F8C3

Εικόνα 7: Δείγματα υπογραφών ιών
[πηγή: (Rad, Masrom, & Ibrahim, 2011)]

Η τρομακτική αύξηση του πλήθους του κακόβουλου λογισμικού και η ταχεία διάδοσή του μέσω του Διαδικτύου έχει προκαλέσει επιπλέον προκλήσεις για τα προγράμματα antimalware. Η πρώτη πρόκληση αφορά τη διαχείριση των υπογραφών. Καθώς νέο κακόβουλο λογισμικό αποκαλύπτεται συνεχώς και ο κίνδυνος προσβολής ενός συστήματος είναι διαρκής, είναι απαραίτητο το antimalware να διαθέτει μια επικαιροποιημένη Βάση Δεδομένων υπογραφών. Αυτό επιτυγχάνεται με συγχρονισμό της τοπικής Βάσης Δεδομένων με μια online Βάση Δεδομένων υπογραφών, στην οποία οι ομάδες ασφαλείας προσθέτουν τις υπογραφές του νέου κακόβουλου λογισμικού που έχουν καταφέρει να ανιχνεύσουν. (Rad, Masrom, & Ibrahim, 2011; Saeed, Selamat, & Abuagoub, 2013) Η δεύτερη πρόκληση έχει να κάνει με την ταχύτητα λειτουργίας του antimalware. Καθώς η Βάση Δεδομένων των υπογραφών μεγαλώνει διαρκώς, η εξέταση αν το σύστημα έχει προσβληθεί από κάποιο γνωστό κακόβουλο λογισμικό γίνεται συνεχώς πιο χρονοβόρα. Σε αυτή την κατεύθυνση αναπτύσσονται και αξιοποιούνται διάφορες μέθοδοι, ώστε η εξέταση ταύτισης με κάθε υπογραφή να γίνει πιο γρήγορη, αλλά να παραμείνει αξιόπιστη. (Rad, Masrom, & Ibrahim, 2011)

Όσον αφορά τον χρόνο κατά τον οποίο εκτελείται η ανίχνευση εφαρμόζονται δύο τακτικές:

- η *κατ' απαίτηση* (*on-demand*) ανίχνευση, όπου η σάρωση εκκινείται από τον ίδιο τον χρήστη στον χρόνο που εκείνος θα επιλέξει
- και η *κατά την πρόσβαση* (*on-access*) ανίχνευση, όπου υπάρχει μια διεργασία στην κύρια μνήμη που εκτελεί αυτόματα τη σάρωση όταν ένα αρχείο ανοίγει, δημιουργείται ή κλείνει.

(Ask, 2006)

Μια άλλη διάκριση στον τρόπο λειτουργίας αφορά την αξιοποίηση *στατικών μεθόδων*, όπου το ύποπτο λογισμικό εξετάζεται όταν βρίσκεται σε μια μονάδα αποθήκευσης (λ.χ. σκληρό δίσκο) ή την αξιοποίηση *δυναμικών μεθόδων*, όπου το ύποπτο λογισμικό εκτελείται στην κύρια μνήμη και εξετάζονται χαρακτηριστικά του κατά τον χρόνο εκτέλεσης. (Saeed, Selamat, & Abuagoub, 2013)

Παρότι η ανίχνευση με βάση τις υπογραφές είναι εξαιρετικά αποδοτική, προσφέροντας πολύ μικρά ποσοστά ψευδών ανιχνεύσεων [134] (Griffin, Schneider, Hu, & Chiueh, 2009), έχει ένα μεγάλο μειονέκτημα· δεν μπορεί να προσφέρει προστασία απέναντι σε νέο κακόβουλο λογισμικό, που δεν έχει μελετηθεί ώστε να έχει δημιουργηθεί μια υπογραφή για αυτό (Saeed, Selamat, & Abuagoub, 2013). Προσπαθώντας να εκμεταλλευτούν αυτό το γεγονός, οι δημιουργοί του κακόβουλου λογισμικού ανέπτυξαν νέες τεχνικές, όπως ο *πολυμορφισμός* ή ο *μεταμορφισμός* [135], που σκοπό έχουν την απόκρυψη ή ακόμα και την αλλοίωση της υπογραφής του λογισμικού.

Μια νεότερη προσέγγιση για την ανίχνευση του κακόβουλου λογισμικού είναι η **ανίχνευση με ευρετική ανάλυση** (**heuristic analysis**). Αυτό το είδος δεν βασίζεται σε υπογραφές και ως εκ τούτου επιτρέπει την ανίχνευση νέου, άγνωστου κακόβουλου λογισμικού. Για να καθοριστεί αν ένα πρόγραμμα είναι κακόβουλο ή μη εξετάζονται διάφορα στατικά χαρακτηριστικά του, καθώς και η δομή του κώδικά του (Rad, Masrom, & Ibrahim, 2011). Μια άλλη προσέγγιση που είτε εφαρμόζεται ανεξάρτητα (Ask, 2006), αλλά πολλές φορές και από κοινού με τεχνικές ευρετικής ανάλυσης (Rad, Masrom, & Ibrahim, 2011) είναι η **συμπεριφοριστική ανάλυση** (**behavioral analysis**). Σε αυτή τη μέθοδο ένα νέο, άγνωστο πρόγραμμα παρακολουθείται σε πραγματικό χρόνο για εξακριβωθεί αν εκτελεί κακόβουλες λειτουργίες (π.χ. διαγραφή ή αλλοίωση σημαντικών αρχείων του συστήματος, άνοιγμα δικτυακών συνδέσεων κ.α.). (Ask, 2006; Rad, Masrom, & Ibrahim, 2011) Επειδή η εκτέλεση ακόμα και με αυτό τον τρόπο περιέχει αρκετό ρίσκο (μπορεί το άγνωστο πρόγραμμα να προκαλέσει ζημιά πριν εξακριβωθεί η κακόβουλη λειτουργία του πάνω σε σημαντικούς πόρους του συστήματος), η συμπεριφοριστική ανάλυση προτιμάται να γίνεται σε μια εικονική μηχανή (Virtual Machine – VM) (Saeed, Selamat, & Abuagoub, 2013), που προσφέρει έναν απομονωμένο χώρο εκτέλεσης, χωρίς να υπάρχει πραγματικός κίνδυνος για το σύστημα. Το μεγάλο μειονέκτημα τόσο της ανίχνευσης με ευρετική ανάλυση, όσο και της ανίχνευσης με συμπεριφοριστική ανάλυση είναι ότι έχουν ένα πολύ σημαντικό ποσοστό ψευδώς θετικών ανιχνεύσεων (Ask, 2006; Rad, Masrom, & Ibrahim, 2011).

[134] Η έννοια των ψευδών ανιχνεύσεων ή *ψευδώς αληθών ανιχνεύσεων* αναλύεται στην ενότητα 6.1.

[135] Για τις έννοιες αυτές δείτε στο κεφάλαιο 4 την ενότητα «Ιός».

5.3.5 Αδυναμίες των βασικών μηχανισμών και συστημάτων κυβερνοάμυνας

Στις προηγούμενες ενότητες αναφέρθηκαν οι βασικότεροι μηχανισμοί και συστήματα κυβερνοάμυνας και αναλύθηκε ο τρόπος που ο καθένας από αυτούς συνεισφέρει στην αύξηση της ασφάλειας του συστήματος. Όμως, ενώ παραμένουν απαραίτητοι – αφού δεν μπορεί να υπάρξει Ασφάλεια Πληροφοριών χωρίς αυτούς – παράλληλα αποδεικνύονται ανεπαρκείς, καθώς στο σύγχρονο περιβάλλον του κυβερνοχώρου, ο όγκος των απειλών διαρκώς αυξάνεται.

Ξεκινώντας από τους μηχανισμούς της αυθεντικοποίησης και της εξουσιοδότησης, αξίζει να επαναλάβουμε πως στα σύγχρονα συστήματα υλοποιούνται μέσα από ένα ενιαίο σύστημα IAM, που επιτρέπει αυξημένο έλεγχο πάνω στη διαχείριση των χρηστών και των πόρων. Όμως, όπως είδαμε στο σχετικό κεφάλαιο, υπάρχει μια πληθώρα κυβερνοεπιθέσεων που θα μπορούσαν να αξιοποιήσουν διάφοροι κακόβουλοι παράγοντες για να «προσπεράσουν» τα εμπόδια που τους θέτουν αυτοί οι μηχανισμοί.

Γενικά, θεωρείται πως ένας κακόβουλος παράγοντας που διαθέτει τις κατάλληλες γνώσεις, τα μέσα, αλλά και την ανάλογη επιμονή, είναι πολύ πιθανόν να καταφέρει να βρει – ή να δημιουργήσει – ένα ρήγμα στην ασφάλεια του συστήματος. Για παράδειγμα, θα μπορούσε να ξεκινήσει με απλές επιθέσεις πρόσβασης, όπως μια επίθεση brute force ή μια επίθεση λεξιλογίου για να «σπάσει» ένα αδύναμο password. Είναι βέβαιο πως σε ένα σύστημα με αδύναμες πολιτικές ασφαλείας που αφορούν τα password, αυτές οι απλές επιθέσεις πρόσβασης θα ήταν αποτελεσματικές.

Αν, ωστόσο, διαπίστωνε πως δεν μπορεί να έχει το επιθυμητό αποτέλεσμα με αυτή την προσέγγιση (γιατί ακολουθούνται οι δέουσες πολιτικές ασφαλείας), τότε θα μπορούσε να επιστρατεύσει επιπλέον πόρους και μέσα, για να εφαρμόσει μια άλλη τακτική που στοχεύσει απευθείας στον «πιο αδύναμο κρίκο» της ασφάλειας ενός συστήματος· στους ανθρώπινους χειριστές του. Σε αυτή την περίπτωση θα μπορούσε να προχωρήσει σε μια επίθεση τύπου phishing με στόχο να «αποσπάσει» τα διαπιστευτήρια ενός νόμιμου χρήστη.

Εναλλακτικά, θα μπορούσε να αξιοποιήσει μια επίθεση με κακόβουλο λογισμικό (που δεν χρειάζεται να δημιουργήσει μόνος του, αρκεί να διαθέτει το ανάλογο χρηματικό ποσό για να το προμηθευτεί), που θα στόχευε σε μια στιγμή απροσεξίας ενός χρήστη για να εισβάλει στο σύστημα. Έτσι, το κακόβουλο λογισμικό, λειτουργώντας με αυτόματο τρόπο μέσα στο προσβεβλημένο σύστημα, θα μπορούσε, ανάλογα με το είδος του, να προσφέρει διάφορες εναλλακτικές για τον κακόβουλο ιδιοκτήτη του. Μια δυνατότητα θα ήταν είναι να παγιδεύσει άμεσα τα αρχεία και να ζητήσει λύτρα για την επαναφορά τους (δείτε ransomware) ή να αρχίσει να καταχράται την επεξεργαστική ισχύ του συστήματος για ίδιον όφελος (δείτε cryptojacking).

Άλλη δυνατότητα είναι να αρχίσει να «κατασκοπεύει» τον χρήστη και να αποστέλνει όσα δεδομένα υποκλέπτει στον ιδιοκτήτη του, με σκοπό να καταφέρει εκείνος να εξακριβώσει πληροφορίες που θα του είναι χρήσιμες (π.χ. διαπιστευτήρια πρόσβασης σε πόρους ή υπηρεσίες – δείτε spyware).

Άλλη δυνατότητα είναι να ανοίξει ένα κανάλι επικοινωνίας για τον απομακρυσμένο έλεγχο του προσβεβλημένου συστήματος, που θα επιτρέψει στον κακόβουλο παράγοντα να αποκτήσει πρόσβαση σε τους τοπικούς ή/και δικτυακούς πόρους του συστήματος (δείτε backdoor και RAT).

Με αξιοποίηση του κατάλληλου κακόβουλου λογισμικού, αντί να στοχεύσει στην απροσεξία ενός χρήστη, θα μπορούσε να στοχεύσει σε ευπάθειες του λογισμικού ή αστοχίες στην παραμετροποίηση του ίδιου του συστήματος για να εισβάλει στο σύστημα. Για αυτό θα έπρεπε να επιστρατεύσει κάποιο κακόβουλο λογισμικό που διαδίδεται αυτόνομα (λ.χ. σκουλήκι ή bot) και προσφέρει στον κακόβουλο ιδιοκτήτη του δυνατότητες όπως αυτές που αναφέρθηκαν στις προηγούμενες παραγράφους (εκβιασμός για αποκόμιση λύτρων, κατάχρηση επεξεργαστικής ισχύς, κατασκοπία χρηστών, παροχή μη εξουσιοδοτημένης απομακρυσμένης πρόσβασης στο σύστημα).

Γίνεται λοιπόν κατανοητό πως, σε περίπτωση επίθεσης με κακόβουλο λογισμικό, οι μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης μόνοι τους δεν μπορούν να προσφέρουν επαρκή προστασία. Συνεπώς, η αποτελεσματική άμυνα πρέπει να σχεδιάζεται σε βάθος, με πρόσθετα μέτρα που εφαρμόζονται σε πολλαπλά επίπεδα.

Ένα τέτοιο πρόσθετο μέτρο είναι η χρήση ενός συστήματος firewall για το φιλτράρισμα της εισερχόμενης και εξερχόμενης κυκλοφορίας του δικτύου. Και όπως αναφέρθηκε και στη σχετική ενότητα, τα συστήματα firewall μπορούν να προσφέρουν προστασία έναντι κακόβουλου λογισμικού που διαδίδεται αυτόνομα μέσω του δικτύου (όπως τα σκουλήκια και τα bot), αλλά και έναντι κάποιων επιθέσεων (όπως αναγνώρισης και DoS). Όμως, ο τρόπος λειτουργίας τους επιβάλει να βρίσκονται στην περίμετρο του συστήματος, οπότε η κυκλοφορία στο εσωτερικό δίκτυο είναι τις περισσότερες φορές εκτός του ελέγχου τους.

Ακόμα και αν ένα firewall κατάφερνε να εξουδετερώσει το 100% του κακόβουλου λογισμικού που διαδίδεται αυτόνομα από το Διαδίκτυο προς το εσωτερικό του δικτύου, οι υπολογιστικοί κόμβοι δεν θα μπορούσαν να θεωρηθούν ασφαλείς από τα άλλα είδη κακόβουλου λογισμικού. Όπως αναφέρθηκε και παραπάνω, μια απροσεξία ενός χρήστη (ή μια πολύ επιδέξια απάτη ενός κακόβουλου παράγοντα) είναι ικανή να προκαλέσει ρήγμα στην άμυνα και υπάρξει μια εισβολή κακόβουλου λογισμικού.

Το επόμενο μέτρο για να αντικρουστούν τέτοια φαινόμενα είναι η χρήση λογισμικού antimalware με σκοπό τον εντοπισμό και την εξουδετέρωση του κακόβουλου λογισμικού σε έναν υπολογιστικό κόμβο. Και πράγματι, όπως ειπώθηκε και στη σχετική ενότητα, τα προγράμματα antimalware είναι ο πιο αποτελεσματικός τρόπος αντιμετώπισης του κακόβουλου λογισμικού. Ωστόσο, υπάρχει ένα πολύ βασικό πρόβλημα· η πιο αξιόπιστη λειτουργία τους βασίζεται σε υπογραφές που έχουν παραχθεί μόνο για γνωστό κακόβουλο λογισμικό. Και στη διαρκή μάχη μεταξύ των επιτιθέμενων και των αμυνόμενων, οι επιτιθέμενοι έχουν βρει τρόπους να εκμεταλλευτούν αυτή την αδυναμία του antimalware· και με την εφαρμογή των τεχνικών του πολυμορφισμού και του μεταμορφισμού να δημιουργούν μια νέα εκδοχή του λογισμικού τους, με άλλη υπογραφή, που τα προγράμματα antimalware δεν γνωρίζουν.

Από την πλευρά τους, οι αμυνόμενοι επιστρατεύουν πιο προηγμένα antimalware που εφαρμόζουν την ευρετική και τη συμπεριφοριστική ανάλυση για να εξακριβώσουν αν ένα νέο, άγνωστο λογισμικό έχει κακόβουλες λειτουργίες. Αυτό το είδος ανίχνευσης όμως δεν είναι τόσο αξιόπιστο όσο η ανίχνευση με βάση τις υπογραφές. Επιπλέον, θα πρέπει να ληφθεί υπόψη ότι τα πιο προηγμένα κακόβουλα προγράμματα λαμβάνουν υπόψη τους την παρουσία των προγραμμάτων antimalware και εφαρμόζουν τεχνικές rootkit για την απόκρυψή τους (των διεργασιών τους στη μνήμη ή/και των αρχείων τους στον δίσκο). Κάποια άλλα έχουν τη δυνατότητα να ανιχνεύουν πότε βρίσκονται υπό ανάλυση από ένα antimalware και να εκτελούν διαφορετικές, μη κακόβουλες λειτουργίες με σκοπό να το εξαπατήσουν και να χαρακτηριστούν ως ασφαλή.

Αν αναλογιστούμε όσα αναφέρθηκαν παραπάνω, προκύπτει ότι οι βασικοί μηχανισμοί άμυνας δεν μπορούν να προσφέρουν επαρκή προστασία σε ένα υπολογιστικό σύστημα που θα μπει στο στόχαστρο επιδέξιων κακόβουλων παραγόντων. Υπάρχουν αδυναμίες όσον αφορά την αντιμετώπιση του κακόβουλου λογισμικού, αλλά και μια σειρά από επιθέσεις (που εξετάστηκαν σε προηγούμενο κεφάλαιο) για τις οποίες οι κλασικοί μηχανισμοί δεν είναι επαρκείς. Την απάντηση σε αυτά τα ζητήματα είναι που προσπαθούν να δώσουν τα *Συστήματα Ανίχνευσης και Πρόληψης Εισβολών* που θα μελετηθούν στο επόμενο κεφάλαιο.

6

Συστήματα Ανίχνευσης και Πρόληψης Εισβολής

6.1 Εισαγωγή

Όπως έγινε προφανές στα προηγούμενα κεφάλαια της εργασίας, η Ασφάλεια Πληροφοριών στον κυβερνοχώρο εν πολλοίς ασχολείται με δύο αντίπαλες ομάδες σε μια διαρκή σύγκρουση. Τους κακόβουλους παράγοντες, που εξαπολύουν διάφορων ειδών κυβερνοεπιθέσεις κατά Πληροφοριακών Συστημάτων και τους υπεύθυνους ασφάλειας, που οργανώνουν την κυβερνοάμυνα των συστημάτων, με σκοπό, αφενός να συνεχιστεί η ομαλή λειτουργία τους και αφετέρου να μην υπάρξει καμία ζημία για τα ίδια τα συστήματα, αλλά και τους οργανισμούς στους οποίους ανήκουν.

Παρουσιάζοντας αυτήν ακριβώς τη σύγκρουση, η προηγούμενη ενότητα ανέδειξε πως ένα σύστημα που αξιοποιεί τους βασικούς μηχανισμούς και συστήματα άμυνας, δεν προστατεύεται επαρκώς έναντι του μεγάλου «οπλοστασίου» που έχουν στη διάθεσή τους οι επιτιθέμενοι. Έτσι λοιπόν, έρχεται ως φυσικό ακόλουθο η ανάγκη για πρόσθετους μηχανισμούς άμυνας που θα εξισορροπήσουν την κατάσταση για τους αμυνόμενους. Αυτοί οι μηχανισμοί είναι τα **Συστήματα Ανίχνευσης Εισβολής** (**Intrusion Detection Systems – IDS**) και τα **Συστήματα Πρόληψης Εισβολής** (**Intrusion Prevention Systems – IPS**).

Όπως αναφέρουν οι Othman, Alsohybe, Ba-Alwi & Zahary (2018), ένα IDS είναι ένα σύστημα που υλοποιείται είτε με τη μορφή λογισμικού, είτε με τη μορφή υλικού (hardware) και έχει σχεδιαστεί για την ανίχνευση επιθέσεων ή γενικά κακόβουλων δραστηριοτήτων σε ένα υπολογιστικό σύστημα ή ένα δίκτυο. Χαρακτηριστικό του είναι πως μετά την αναγνώριση μιας πιθανής παραβίασης της ασφάλειας, παράγει μια *ειδοποίηση* (*alert*) προς τον διαχειριστή [136], για να αντιμετωπίσει την εισβολή.

[136] Για λόγους απλότητας, στην παρούσα εργασία γίνεται συχνά αναφορά σε κάποιον *διαχειριστή*. Στην πράξη υπολογιστικά συστήματα όπως τα IDPS, έχουν τέτοιο μέγεθος και πολυπλοκότητα, που απαιτείται μια ομάδα εξειδικευμένου προσωπικού που ασχολείται με την ασφάλεια του συστήματος και όχι απλά ένα άτομο. Στο υπόλοιπο της εργασίας, για τα άτομα που ανήκουν στην ομάδα αυτή θα χρησιμοποιείται ο όρος **αναλυτής ασφαλείας** (**security analyst**) ή ακόμα και ο όρος **διαχειριστής ασφαλείας** (**security administrator**).

Παρομοίως, ένα IPS είναι ένα σύστημα που προσπαθεί να ανιχνεύσει επιθέσεις και κακόβουλες δραστηριότητες, αλλά επιπλέον περιλαμβάνει και λειτουργίες απόκρισης για την αντιμετώπιση αυτών των απειλών. Όπως εύστοχα παρατηρούν οι Ashoor & Gore (2011), ένα IDS λειτουργεί σαν μια περίπολος, ενώ ένα IPS λειτουργεί σαν ένας φρουρός πύλης [137].

Πριν γίνει περεταίρω ανάλυση των συστημάτων αυτών, είναι σημαντικό να σταθούμε σε μερικά σημεία: το πρώτο είναι πως στον πυρήνα και των δύο ειδών συστημάτων βρίσκεται η διεργασία της ανίχνευσης μιας εισβολής, οπότε είναι απαραίτητο να κατανοήσουμε πρώτα αυτή την έννοια. Οι Scarfone & Mell (2007) σε μια ειδική έκδοση του οργανισμού NIST εξηγούν πως η **ανίχνευση εισβολής** είναι η παρακολούθηση των **γεγονότων (events)** που παρατηρούνται σε ένα υπολογιστικό σύστημα ή σε ένα δίκτυο και η ανάλυσή τους για σημάδια πιθανών **περιστατικών** (**incidents**).

Τα περιστατικά είναι παραβιάσεις ή απειλές για μια επικείμενη παραβίαση των πολιτικών ασφαλείας, των αποδεκτών πολιτικών χρήσης ή των συνήθων πρακτικών ασφάλειας. Έχουν πολλές αιτίες, όπως η εκτέλεση κακόβουλου λογισμικού, η μη εξουσιοδοτημένη πρόσβαση από επιτιθέμενους ή ακόμα και νόμιμους χρήστες που δεν κάνουν σωστή χρήση των προνομίων τους ή αποπειρώνται να αποκτήσουν επιπλέον προνόμια για τα οποία δεν έχουν εξουσιοδότηση. Αξίζει να τονιστεί ιδιαίτερα, πως δεν είναι όλα τα περιστατικά κακόβουλα από τη φύση τους, καθώς κάποια μπορεί να προέρχονται από αθώα λάθη χρηστών του συστήματος. (Scarfone & Mell, 2007)

Το δεύτερο σημείο που πρέπει να αναφερθεί είναι ουσιαστικά απόρροια του πρώτου. Καθώς τα IDS και τα IPS βασίζονται στις ίδιες αρχές λειτουργίας, είναι σύνηθες να εξετάζονται από την ερευνητική κοινότητα από κοινού ως **Συστήματα Ανίχνευσης και Πρόληψης Εισβολής** (**Intrusion Detection and Prevention Systems – IDPS**). Και στην πράξη όμως, είναι σύνηθες να δημιουργείται ένα προϊόν, που ενσωματώνει τις λειτουργίες και από τις δύο προσεγγίσεις και ο διαχειριστής του συστήματος να απενεργοποιεί τις λειτουργίες που αφορούν την πρόληψη των εισβολών, ώστε το προϊόν να μετατραπεί σε ένα αμιγές IDS (Scarfone & Mell, 2007).

Ένα τρίτο σημείο είναι πως τα συστήματα IDPS αδυνατούν να προσφέρουν απόλυτη ακρίβεια στις ανιχνεύσεις τους. Για αυτόν τον λόγο είναι απαραίτητο να έχουμε κατά νου δύο έννοιες: μιλάμε για ένα **ψευδώς αληθές** (**false positive**) αποτέλεσμα, όταν μία φυσιολογική δραστηριότητα αναγνωρίζεται από το σύστημα ως κακόβουλη και για ένα **ψευδώς αρνητικό** (**false negative**) αποτέλεσμα, όταν μία κακόβουλη δραστηριότητα θεωρείται φυσιολογική και δεν ανιχνεύεται. (Scarfone & Mell, 2007; Saeed, Selamat, & Abuagoub, 2013)

[137] Αυτή η παρομοίωση αξίζει να αναλυθεί περεταίρω. Γενικά, μια περίπολος έχει σκοπό να εντοπίσει τον εχθρό και να το αναφέρει στις δυνάμεις που φυλούν το κάστρο, χωρίς να εμπλακεί σε μάχη. Αντιθέτως, ο φρουρός της πύλης εμποδίζει και συγκρούεται με όποιον δεν πρέπει να εισέλθει στο κάστρο.

6.2 Τεχνικές ανίχνευσης

Ο McHugh (2001) μας προσφέρει μια ιστορική αναδρομή και περιγράφει πως τα συστήματα που ασχολούνταν με την ανίχνευση μιας εισβολής, αρχικά βασίζονταν στη λογική του *ελέγχου* (*audit*) των δεδομένων και πως αυτά τα δεδομένα συλλέγονταν από τον έναν και μοναδικό υπολογιστή που προστάτευαν. Η πρώτη πηγή των *δεδομένων ελέγχου* (*audit data*) ήταν οι συμπληρωματικές πληροφορίες που αποθήκευε το ίδιο το σύστημα στο πλαίσιο της διαχείρισης των λογαριασμών των χρηστών. Στη συνέχεια, όπως εξηγεί, η προσέγγιση αυτή γενικεύτηκε στη συλλογή δεδομένων από κάθε πηγή που θεωρείτο πως έχει σχέση με την ασφάλεια του συστήματος. Η ανάπτυξη συστημάτων που συλλέγουν και αναλύουν δεδομένα από το δίκτυο είναι μια ιδέα που υλοποιήθηκε σε μεταγενέστερο χρόνο. Ωστόσο, αυτή η ιδέα επικράτησε και, στα σύγχρονα IDPS, τα δεδομένα δικτύου είναι η κύρια – και σε πολλές περιπτώσεις η μοναδική – πηγή δεδομένων προς έλεγχο. (McHugh, 2001)

Όσον αφορά τη σύγχρονη προσέγγιση για την ανίχνευση, οι ερευνητές ξεχωρίζουν τρεις τεχνικές [138], που εφαρμόζοντα είτε μεμονωμένα, είτε σε συνδυασμό:

- Η πρώτη είναι η **ανίχνευση με βάση την υπογραφή** (**signature-based detection**) [139]. Αποτελεί την πιο απλή μορφή ανίχνευσης και βασίζεται στη σύγκριση των δεδομένων ελέγχου με μια λίστα υπογραφών γνωστών απειλών [140]. Ως παραδείγματα δεδομένων ελέγχου σε αυτή την περίπτωση μπορούμε να αναφέρουμε τμήματα από ένα πακέτο δικτύου ή κάποια εγγραφή σε ένα *αρχείο καταγραφής* (*log file*). Είναι η πιο αποτελεσματική μέθοδος απέναντι σε γνωστές απειλές, όμως είναι εντελώς αναποτελεσματική έναντι απειλών για τις οποίες δεν υπάρχουν υπογραφές. (Scarfone & Mell, 2007)
    - Η δεύτερη είναι η **ανίχνευση με βάση μια ανωμαλία** (**anomaly-based detection**). Ανήκει στην κατηγορία των μεθόδων συμπεριφοριστικής ανάλυσης (Azeez, et al., 2020) και βασίζεται στη σύγκριση δραστηριοτήτων που θεωρούνται φυσιολογικές με τα γεγονότα που παρατηρούνται στο σύστημα, ώστε να βρεθούν σοβαρές αποκλίσεις (Scarfone & Mell, 2007). Απαιτεί τη δημιουργία *προφίλ* (*profile*) φυσιολογικής δραστηριότητας για τους χρήστες, τους υπολογιστικούς κόμβους, τις εφαρμογές ή τις δικτυακές συνδέσεις (Azeez, et al., 2020). Μερικά μόνο παραδείγματα των χαρακτηριστικών (attributes) που χρησιμοποιούνται για τη δημιουργία των προφίλ είναι το πλήθος των email που αποστέλλει ένας χρήστης ημερησίως, το πλήθος των ανεπιτυχών προσπαθειών σύνδεσης σε έναν υπολογιστή, το επίπεδο χρήσης του επεξεργαστή ενός κόμβου σε ένα συγκεκριμένο χρονικό διάστημα. (Scarfone & Mell, 2007)

[138] Ένα πολύ ενδιαφέρον σημείο είναι πως αυτές οι μέθοδοι έχουν παρόμοια λογική με τις μεθόδους που εφαρμόζονται στα συστήματα antimalware και firewall (δείτε σχετικά τις αντίστοιχες ενότητες). Επίσης, να σημειωθεί ότι σε πολλές υλοποιήσεις εμπορικών συστημάτων, οι μέθοδοι αναμειγνύονται σε τέτοιο βαθμό, που κατάταξή τους σε κάποια κατηγορία δεν είναι εύκολη.

[139] Όπως σημειώνουν οι Scarfone & Mell (2007), κάποιοι χρησιμοποιούν τον όρο *misuse detection* ως συνώνυμο. Επίσης, αναφέρουν ότι πολλές φορές θεωρείται πως η τεχνική της ανίχνευσης με βάση τις υπογραφές περιλαμβάνει και αυτή της της ανίχνευσης με επίγνωση της κατάστασης του πρωτοκόλλου, που θα αναλυθεί παρακάτω· στα πλαίσια της παρούσας εργασίας εξετάζονται και θεωρούνται ξεχωριστές τεχνικές.

[140] Πολλά συστήματα αντί του όρου υπογραφές, χρησιμοποιούν τον όρο **κανόνες** (**rules**).

Για κάθε χαρακτηριστικό που χρησιμοποιείται σε ένα προφίλ, ορίζεται κάποιο *όριο* (*threshold*)· στη φάση της ανίχνευσης το χαρακτηριστικό αυτό παρακολουθείται για κάποιο χρονικό διάστημα και αν η τιμή του ξεπεράσει αυτό το όριο, τότε κρίνεται ως ανωμαλία (δηλαδή πιθανή απειλή για το σύστημα). (Scarfone & Mell, 2007) Η δημιουργία των προφίλ, δηλαδή ο καθορισμός των ορίων των χαρακτηριστικών, απαιτεί μια περίοδο εκπαίδευσης πριν τεθεί σε λειτουργία η διεργασία της ανίχνευσης (McHugh, 2001; Scarfone & Mell, 2007).

Τα προφίλ μπορεί να είναι είτε *στατικά* ή *δυναμικά*. Η διαφορά τους είναι πως τα πρώτα παραμένουν αμετάβλητα στην πάροδο του χρόνου και σταδιακά γίνονται λιγότερο ακριβή, αφού και η φυσιολογική συμπεριφορά δεν είναι σταθερή, αλλά μεταβάλλεται· συνεπώς, πρέπει να αντικαθίστανται περιοδικά. Τα δεύτερα αναπροσαρμόζονται διαρκώς καθώς καταγράφονται νέα γεγονότα. Όμως, ούτε και αυτά είναι άψογα, καθώς οι επιτιθέμενοι μπορεί να ξεκινήσουν με «μικρής έντασης» κακόβουλη δραστηριότητα και σιγά σιγά να αυξάνουν αυτή την «ένταση». Με αυτό τον τρόπο το σύστημα μπορεί να εκλάβει την κακόβουλη δραστηριότητα ως φυσιολογική και να τη συμπεριλάβει στο προφίλ. (Scarfone & Mell, 2007)

Η μέθοδος της ανίχνευσης με βάση μια ανωμαλία έχει διαφορετικές προσεγγίσεις για να τον καθορισμό των ορίων· από την αξιοποίηση στατιστικών τεχνικών ή νευρωνικών δικτύων, μέχρι τη χρήση τεχνικών Τεχνητής Νοημοσύνης εμπνευσμένων από το ανθρώπινο ανοσοποιητικό σύστημα (McHugh, 2001). Ισχυρότερο πλεονέκτημα της μεθόδου είναι η δυνατότητά της να ανιχνεύσει νέες, παντελώς άγνωστες απειλές. Το μειονέκτημα όμως είναι εξίσου σοβαρό· η μέθοδος είναι δυνατόν να παραγάγει ένα υψηλό πλήθος ψευδώς αληθών ανιχνεύσεων. (Scarfone & Mell, 2007)

- Η τρίτη είναι η **ανάλυση με επίγνωση της κατάστασης του πρωτοκόλλου** (**stateful protocol analysis**) [141]. Και αυτή ανήκει στην κατηγορία των μεθόδων συμπεριφοριστικής ανάλυσης (Scarfone & Mell, 2007). Σε αντίθεση με την ανίχνευση με βάση τις ανωμαλίες, που βασίζεται σε προφίλ που παράγονται για συγκεκριμένους υπολογιστές ή για συγκεκριμένα δίκτυα, η μέθοδος αυτή βασίζεται σε καθολικά αποδεκτά προφίλ που καθορίζουν πως συγκεκριμένα πρωτόκολλα θα έπρεπε ή δεν θα έπρεπε να συμπεριφέρονται ή ακόμα και αποδεκτές ή μη αποδεκτές αλληλουχίες εκτέλεσης εντολών (Azeez, et al., 2020; Scarfone & Mell, 2007). Στα μειονεκτήματα της μεθόδου αναφέρεται πως είναι συχνά δύσκολο ή και αδύνατο να παραχθούν απολύτως ακριβή μοντέλα κάποιων πρωτοκόλλων, ότι αδυνατεί να εντοπίσει επιθέσεις που ακολουθούν την αποδεκτή συμπεριφορά ενός πρωτοκόλλου και ότι είναι αρκετά απαιτητική σε υπολογιστικούς πόρους. (Scarfone & Mell, 2007)

Επικουρικά προς την ανάλυση υπογραφής και την ανάλυση με επίγνωση της κατάστασης του πρωτοκόλλου, κάποια IDPS αξιοποιούν μια *μαύρη λίστα* (*black list*) για να μπλοκάρουν με δυναμικό τρόπο – και ενδεχομένως προσωρινά – συγκεκριμένες οντότητες (IP, URL, εφαρμογές, τύπους αρχείων, υπολογιστικούς κόμβους κ.α.) που έχουν συσχετιστεί με κακόβουλη δραστηριότητα. Παρομοίως, συχνά χρησιμοποιείται και κάποια *λευκή λίστα* (*white list*) ως αντίμετρο στις επιβεβαιωμένες ψευδώς θετικές ανιχνεύσεις. (Scarfone & Mell, 2007)

[141] Όπως σημειώνουν οι Scarfone & Mell (2007), πολλά εμπορικά συστήματα που συνδυάζουν κάποια μορφή της τεχνικής αυτής με τη λειτουργικότητα ενός firewall για την εξέταση των πακέτων του δικτύου, συχνά χρησιμοποιούν τον όρο *deep packet inspection*.

6.3 Πηγές δεδομένων ελέγχου και τύποι συστημάτων

Όσον αφορά την πηγή των δεδομένων ελέγχου, υπάρχουν δύο εναλλακτικές. Η πρώτη είναι η άμεση λήψη των δεδομένων ελέγχου από το λειτουργικό σύστημα ή τις εφαρμογές ενός υπολογιστικού συστήματος. Η δεύτερη είναι η παρατήρηση των πακέτων που κυκλοφορούν στο δίκτυο. (McHugh, 2001) Με βάση αυτές τις δύο πηγές, τα συστήματα διαχωρίζονται σε τρεις βασικούς τύπους:

- Τα *Host-based IDPS* (*HIDS / HIPS*), που ασχολούνται με την ανίχνευση απειλών σε έναν συγκεκριμένο υπολογιστικό κόμβο. Ένα τέτοιο σύστημα έχει τη μορφή λογισμικού ή συσκευής [142] που εγκαθίσταται στον κόμβο που πρέπει να προστατεύει – το συστατικό μέρος που εκτελεί την ανίχνευση σε ένα host-based IDPS ονομάζεται *agent*. Για να επιτύχει την προστασία του κόμβου του, ένας agent παρακολουθεί τα χαρακτηριστικά του κόμβου, καθώς και κάθε γεγονός που τον αφορά. Οι κυριότερες πηγές από τις οποίες αντλεί τα δεδομένα ελέγχου είναι: τα αρχεία καταγραφής του λειτουργικού συστήματος, τα αρχεία ρυθμίσεων των εφαρμογών και του λειτουργικού συστήματος, οι λειτουργίες που εκτελούν οι εφαρμογές και οι εκτελούμενες διεργασίες, οι αλλαγές και οι τροποποιήσεις αρχείων, αλλά και η εισερχόμενη και η εξερχόμενη κυκλοφορία δικτύου [143]. (Scarfone & Mell, 2007)

  Το κυριότερο πλεονέκτημα των host-based IDPS είναι η ικανότητά τους να αναλύουν κρυπτογραφημένη δραστηριότητα [144]. Και αντιστοίχως, στα μειονεκτήματά τους ξεχωρίζουν η αδυναμία προστασίας από εξωτερικές απειλές (λ.χ. επιθέσεις αναγνώρισης, DoS) και, κυρίως, πως καταναλώνουν ένα σημαντικό μέρος των υπολογιστικών πόρων του κόμβου που εποπτεύουν. (Othman, T. Alsohybe, Ba-Alwi, & Zahary, 2018) Σαν γενικό συμπέρασμα [145], θα λέγαμε πως τα συστήματα αυτά προσφέρουν αυξημένη προστασία στο εσωτερικό σύστημα, αλλά είναι πιο επιρρεπή σε εξωτερικές επιθέσεις. (Saeed, Selamat, & Abuagoub, 2013)

[142] Σε αυτή τη μορφή, είναι η συσκευή που περιέχει το λογισμικό agent, αντί αυτό να εγκατασταθεί απευθείας στον υπολογιστικό κόμβο. Συνδέεται αποκλειστικά με έναν κόμβο και δέχεται πακέτα από το δίκτυο όπως ένα network-based IDPS, ωστόσο εκτελεί επιπλέον τις ίδιες ή παρόμοιες λειτουργίες με ένα λογισμικό agent και για αυτό κατατάσσεται στα host-based συστήματα. (Scarfone & Mell, 2007)

[143] Είναι σημαντικό να παρατηρήσουμε πως παρακολουθείται μόνο η κυκλοφορία δικτύου που «περνάει» από τον συγκεκριμένο κόμβο. Οπότε, κακόβουλη κυκλοφορία που αφορά άλλον κόμβο και δεν «περνάει» ποτέ από αυτόν, δεν μπορεί να ανιχνευθεί από τον συγκεκριμένο agent.

[144] Είτε ως αποστολέας, είτε ως παραλήπτης ενός κρυπτογραφημένου πακέτου, ο υπολογιστικός κόμβος έχει πρόσβαση πάντοτε στο απλό κείμενο, οπότε ο agent μπορεί να το αναλύσει.

[145] Με μια προσεκτική εξέταση στον τρόπο λειτουργίας τους, θα διαπιστώσει κανείς ότι τα host-based συστήματα γενικεύουν και επεκτείνουν τον τρόπο λειτουργίας του λογισμικού antimalware. Το λογισμικό antimalware ανιχνεύει και εξουδετερώνει κακόβουλα προγράμματα, ενώ τα host-based IDPS γενικά επεκτείνουν τη διεργασία ελέγχου πέρα από τις εφαρμογές και στους χρήστες, τις συνδέσεις και την κυκλοφορία του δικτύου. Η διαφορά είναι πως στα host-based IDPS η διαδικασία της εξουδετέρωσης έχει διαφορετική αντιμετώπιση ανάλογα με τον τύπο του. Μόνο στα HIPS υπάρχουν αυτοματοποιημένες λειτουργίες απόκρισης για την εξουδετέρωση της απειλής. Αυτό δεν ισχύει στα HIDS, που η λειτουργία τους περιορίζεται στην ειδοποίηση των υπεύθυνων ασφάλειας που θα αναλάβουν την απόκριση στο περιστατικό με μη αυτοματοποιημένο τρόπο.

- Τα *Network-based IDPS* (*NIDS / NIPS*), τα οποία ασχολούνται με την ανίχνευση απειλών σε ένα δίκτυο κόμβων. Ένα τέτοιο σύστημα έχει τη μορφή ενός ανεξάρτητου κόμβου – συσκευής, που ονομάζεται *sensor* και ο οποίος παρακολουθεί και αναλύει τα πακέτα που κυκλοφορούν σε ένα συγκεκριμένο τμήμα του δικτύου. Η ανάλυση γίνεται στα επίπεδα πρωτοκόλλου δικτύου, μεταφοράς και εφαρμογής. (Scarfone & Mell, 2007)

  Το κύριο πλεονέκτημα των network-based IDPS είναι πως ένας μόνο sensor, αν τοποθετηθεί στο σωστό σημείο του δικτύου, είναι αρκετός για προσφέρει προστασία σε πολλούς υπολογιστικούς κόμβους (McHugh, 2001). Αυτό προσφέρει ένα πρόσθετο πλεονέκτημα όσον αφορά το συνολικό κόστος για την επίτευξη προστασίας. Επίσης, ένας sensor δεν επιβαρύνει την υπολογιστική απόδοση των κόμβων που παρακολουθεί. Τέλος, ένα network-based IDPS λειτουργεί πιο γρήγορα καθώς εποπτεύει την κίνηση σε πραγματικό χρόνο ή σχεδόν πραγματικό χρόνο. (Othman, T. Alsohybe, Ba-Alwi, & Zahary, 2018)

  Από την άλλη, υπάρχει μια σειρά από περιορισμούς. Καταρχάς, η ανάλυση κυκλοφορίας που γίνεται με κρυπτογραφημένα πακέτα είναι εξαιρετικά δύσκολη [146]. (Scarfone & Mell, 2007) Δεύτερον, η ανίχνευση γίνεται αναποτελεσματική αν υπάρχουν και άλλα μονοπάτια, εκτός της εποπτείας του sensor, για την επικοινωνία με έναν κόμβο. (McHugh, 2001) Τρίτον, υπάρχει μεγάλη δυσκολία στη διαχείριση ενός δικτύου με υψηλή κυκλοφορία [147]. (Scarfone & Mell, 2007) Αξιοσημείωτο είναι το γεγονός πως ένας sensor μπορεί να γίνει και ο ίδιος στόχος επίθεσης, καθώς ένας εισβολέας μπορεί να αποφασίσει να τον θέσει εκτός λειτουργίας προτού επιτεθεί σε κάποιον άλλο κόμβο του δικτύου. (McHugh, 2001)

  Οι Saeed, Selamat & Abuagoub (2013) επισημαίνουν πως τα network-based IDPS έχουν τα αντίθετα χαρακτηριστικά από τα host-based IDPS· προσφέρουν καλή προστασία σε εξωτερικές επιθέσεις, αλλά είναι πιο επιρρεπή σε επιθέσεις που πηγάζουν από έναν εσωτερικό κόμβο [148]. Οι Othman, T. Alsohybe, Ba-Alwi & Zahary (2018) παρατηρούν πως τα network-based IDPS επικεντρώνονται στην ανίχνευση της εκμετάλλευσης ευπαθειών, σε αντίθεση με τα host-based IDPS που επικεντρώνονται στην ανίχνευση της εκμετάλλευσης προνομίων.

[146] Αν για παράδειγμα, η κακόβουλη ενέργεια βρίσκεται στο πρωτόκολλο εφαρμογής και χρησιμοποιείται κρυπτογράφηση, τότε ο sensor δεν μπορεί να εκτελέσει ανάλυση σε αυτό το επίπεδο, αλλά μόνο στα χαμηλότερα – δηλαδή μπορεί να αναζητήσει απειλές μόνο με βάση τις επικεφαλίδες των πακέτων των πρωτοκόλλων δικτύου και μεταφοράς. Ακόμα χειρότερα, αν η κρυπτογράφηση εφαρμόζεται στο πρωτόκολλο δικτύου (με IPsec), τότε ο sensor δεν θα μπορεί να ελέγξει ούτε αυτά.

[147] Ο sensor ίσως να μην προλαβαίνει να αναλύσει όλα τα πακέτα που καταφτάνουν στο δίκτυο. Αν το σύστημα είναι ένα NIDS, τότε αυτό σημαίνει πως θα υπάρχουν πακέτα που δεν θα ελεγχθούν – συνεπώς απειλές μπορεί να περάσουν απαρατήρητες. Στην περίπτωση που το σύστημα είναι NIPS, υπάρχει η πιθανότητα να αρχίσει να απορρίπτει αυτόματα όλα τα πακέτα που δεν προλαβαίνει να ελέγξει και να προκαλέσει ένα DoS σε νόμιμα αιτήματα.

[148] Εύκολα διαπιστώνει κανείς ότι τα network-based IDPS και τα firewall λειτουργούν με παρόμοιο τρόπο (ειδικά τα NIPS που έχουν τη δυνατότητα να μπλοκάρουν πακέτα). Η διαφορά είναι πως τα firewall ασχολούνται με την κυκλοφορία από το Διαδίκτυο προς το εσωτερικό δίκτυο και αντίστροφα, ενώ τα network-based IDPS στοχεύουν στην παρακολούθηση της κυκλοφορίας στο εσωτερικό δίκτυο. Στην πράξη, αυτές οι δύο προσεγγίσεις δρουν συμπληρωματικά. Είτε ως υλοποίηση σε έναν κόμβο (οι προηγμένες μορφές firewall ουσιαστικά ενσωματώνουν λειτουργικότητα NIPS), είτε με τη μορφή δύο κόμβων που εργάζονται ο ένας στα πακέτα που παραδίδει ο άλλος.

- Τα *hybrid IDPS* (*hybrid IDS / hybrid IPS*), στα οποία γίνεται ανάμειξη των δυνατοτήτων των host-based IDPS και των network-based IDPS. Σε αυτή την προσέγγιση, το IDPS αποτελείται από πολλαπλά υποσυστήματα που βρίσκονται σε διαφορετικούς κόμβους του δικτύου. Πιο συγκεκριμένα, ένας IDPS agent εγκαθίσταται και λειτουργεί σε κάθε σημαντικό [149] κόμβο, ενώ υπάρχει και κάποιος IDPS sensor για να εποπτεύει την κυκλοφορία στο δίκτυο. Όλα αυτά τα υποσυστήματα συλλέγουν δεδομένα από αυτούς τους κόμβους και το δίκτυο και τα αποστέλλουν σε έναν κόμβο κεντρικής διαχείρισης (management) του IDPS για ανάλυση. (Saeed, Selamat, & Abuagoub, 2013; Othman, T. Alsohybe, Ba-Alwi, & Zahary, 2018).

  Ο συνδυασμός των δύο προσεγγίσεων προσφέρει μια λύση που έχει τα πλεονεκτήματα και των δύο. Όμως, αυτή η προσέγγιση έχει επιπρόσθετη επιβάρυνση στο σύστημα, τόσο σε κυκλοφορία δικτύου, όσο και σε υπολογιστική ισχύ. Επιπλέον, πρέπει να σημειωθεί ότι στα υβριδικά συστήματα η ανάλυση είναι γενικά πιο αργή. (Othman, T. Alsohybe, Ba-Alwi, & Zahary, 2018)

Πέρα από αυτούς τους βασικούς τύπους υπάρχουν και κάποιοι άλλοι, που εφαρμόζονται σε ειδικότερες περιπτώσεις. Για παράδειγμα, ο τύπος *Wireless IDPS* αποτελεί μια παραλλαγή του network-based IDPS, με εφαρμογή σε ασύρματα δίκτυα. Σε αυτά τα δίκτυα υπάρχουν διάφορες προκλήσεις· το μέσο μετάδοσης (ο αέρας) είναι γενικά διαθέσιμο στους πάντες, τα συστήματα έχουν περιορισμένη υπολογιστική ισχύ λόγω εξάρτησης από μια μπαταρία κ.α. Η διαφορετική τους φύση απαιτεί την αξιοποίηση πρωτοκόλλων με διαφορετική λογική. Παράλληλα όμως, τα καθιστά πολύ πιο ευάλωτα σε κάποια είδη επιθέσεων (όπως MitM και DoS). Συνεπώς, η προσέγγιση των κλασικών network-based IDPS δεν επαρκεί και πρέπει να ακολουθηθεί μια διαφορετική προσέγγιση με ανίχνευση στο επίπεδο ζεύξης δεδομένων [150]. (Scarfone & Mell, 2007; Azeez, et al., 2020)

Παραλλαγή των network-based IDPS είναι και τα συστήματα *Network Behavior Analysis* (*NBA*). Αυτά διαφέρουν γιατί αξιοποιούν δεδομένα (είτε επιπρόσθετα, είτε αποκλειστικά) από άλλες συσκευές, όπως switch και δρομολογητές, ενώ, συνήθως, παραλαμβάνουν αυτά τα δεδομένα σε δέσμες, πράγμα που σημαίνει πως η ανίχνευση σε πραγματικό χρόνο είναι δύσκολη. Σκοπός σε κάθε περίπτωση είναι να αναγνωρίσουν μη συνηθισμένη ροή κυκλοφορίας στο δίκτυο με σκοπό να ανιχνεύσουν διάφορες επιθέσεις όπως DDoS, επιθέσεις από συγκεκριμένους τύπους κακόβουλου λογισμικού (όπως σκουλήκια ή backdoor) κ.α. (Scarfone & Mell, 2007)

Τέλος, η εξάπλωση του Υπολογιστικού Νέφους (Cloud Computing) γέννησε την ανάγκη για έναν νέο τύπο IDPS. Παρότι οι κλασικοί τύποι μπορούν να λειτουργήσουν μέσα σε μια Εικονική Μηχανή (Virtual Machine – VM), η προστασία δεν είναι επαρκής, καθώς αναπτύχθηκαν επιθέσεις κατά των ίδιων των VM και του Hypervisor που τις ελέγχει. Τα *Hypervisor-based IDPS* ακολουθούν μια πιο ολιστική προσέγγιση που επιτρέπει την εποπτεία και την ανάλυση των δεδομένων που διακινούνται μέσα στο εικονικό δίκτυο μεταξύ των VM, καθώς και μεταξύ των VM και του Hypervisor, με σκοπό να ανιχνευθούν οι απειλές. (Othman, T. Alsohybe, Ba-Alwi, & Zahary, 2018)

[149] Συνήθως, για λόγους αποδοτικότητας δεν επιβαρύνονται όλοι οι κόμβοι με έναν agent, αλλά μόνο οι πιο σημαντικοί. Για τους υπόλοιπους παρέχεται η βασική προστασία από κάποιον sensor.

[150] Να αναφερθεί ως υπενθύμιση ότι στα κλασικά network-based IDPS η ανίχνευση γίνεται στα ανώτερα επίπεδα και δεν ασχολούνται με το επίπεδο ζεύξης δεδομένων.

6.4 Τυπική δομή

Παρότι τα IDPS μπορεί να έχουν διαφορετικές μορφές, μπορεί να θεωρηθεί πως όλα αποτελούνται από κάποια βασικά συστατικά μέρη:

- Έναν (τουλάχιστον) *sensor* ή *agent* που, όπως αναφέρθηκε ήδη, είναι το κομμάτι του IDPS που ασχολείται με την παρακολούθηση και την ανάλυση των δεδομένων ελέγχου σε ένα network-based IDPS ή σε ένα host-based IDPS αντίστοιχα.
- Προαιρετικά, έναν *management server* που αναλαμβάνει να αναλύσει και να συσχετίσει δεδομένα που συγκεντρώνει από μεμονωμένους sensor ή agent, με σκοπό να ανιχνεύσει απειλές που δεν έχουν καταφέρει να ανιχνεύσουν αυτοί [151]. Μπορεί να είναι είτε ανεξάρτητη συσκευή, είτε λογισμικό. Στις πιο απλές υλοποιήσεις IDPS αυτή η λειτουργικότητα μπορεί να μην παρέχεται.
- Έναν *database server*, που αποθηκεύει τα γεγονότα που έχουν καταγράψει οι agent, οι sensor ή οι management server.
- Την *κονσόλα*, που αποτελεί τη διεπαφή για τη διαχείριση του IDPS από έναν διαχειριστή. Ενδεχομένως, ανάλογα με το IDPS, να επιτρέπει επιπλέον σε έναν διαχειριστή να παραμετροποιήσει τους μεμονωμένους sensor ή agent.

(Scarfone & Mell, 2007)

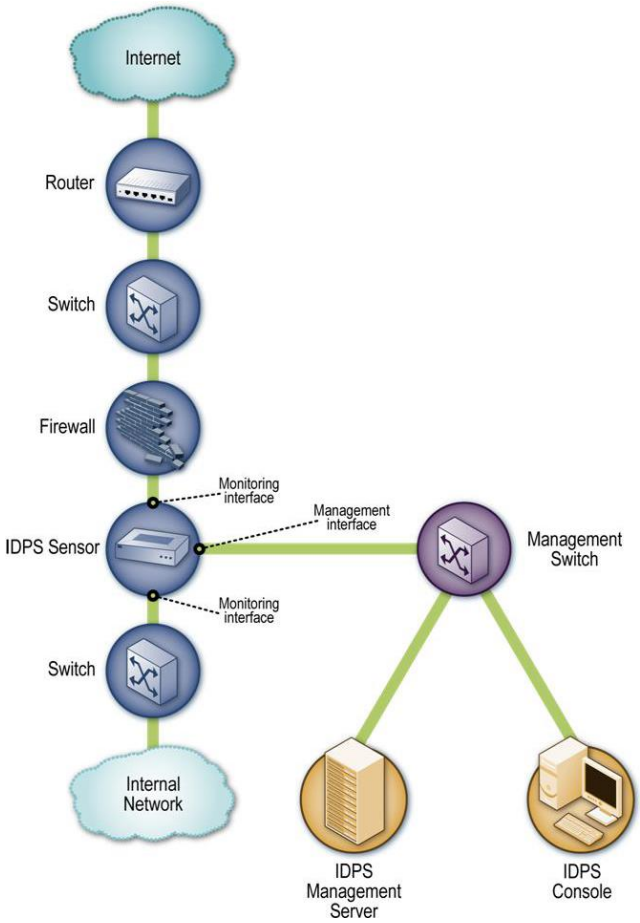
Αυτό που πρέπει να τονιστεί είναι πως σε κάποιες υλοποιήσεις αυτά τα συστατικά μέρη μπορεί να υπάρχουν ως ξεχωριστές συσκευές, ενώ σε άλλες, πιο απλές υλοποιήσεις, να υπάρχουν μόνο ως λογικά τμήματα ενός ενιαίου λογισμικού.

[151] Ουσιαστικά, αυτή είναι η λειτουργικότητα είναι που μετατρέπει ένα σύστημα σε hybrid IDPS.

6.5 *Καταστάσεις λειτουργίας*

Από τις προηγούμενες ενότητες γίνεται ξεκάθαρο πως ένας agent εγκαθίσταται στον εκάστοτε κόμβο που πρέπει να προστατέψει και ελέγχει τη (δικτυακή και μη) δραστηριότητα μόνο αυτού του κόμβου. Ένας sensor από την άλλη, πρέπει να προστατέψει πολλούς κόμβους ταυτόχρονα, ελέγχοντας μόνο τη δικτυακή δραστηριότητά τους. Αυτό έχει ως αποτέλεσμα η θέση του σε σχέση με τους άλλους κόμβους να αποκτά ιδιαίτερη σημασία. Σε σχέση με αυτή τη θέση, διακρίνονται δύο καταστάσεις λειτουργίας:

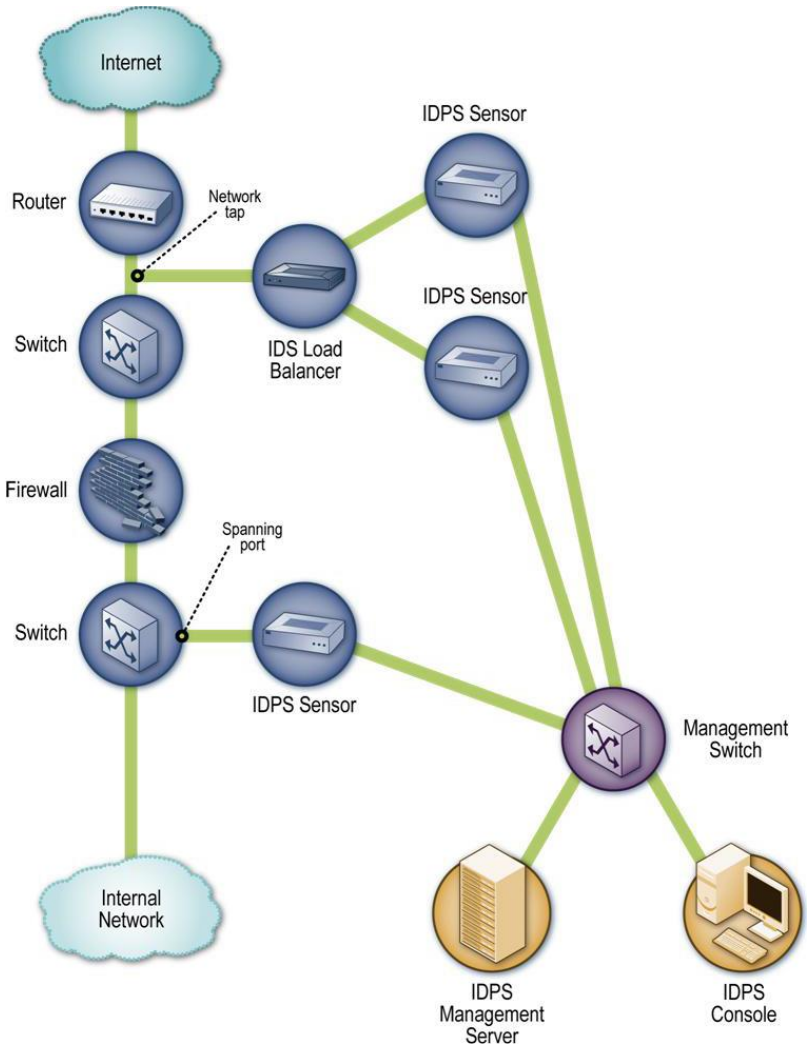
- Σε κατάσταση *inline* κάθε πακέτο που κυκλοφορεί στο δίκτυο πρέπει να περάσει από τον sensor για να ελεγχθεί πριν συνεχίσει προς τον τελικό προορισμό του.



Εικόνα 8: IDPS σε κατάσταση inline
[πηγή: (Scarfone & Mell, 2007)]

Στην παραπάνω εικόνα από το (Scarfone & Mell, 2007) παρουσιάζεται η τυπική εγκατάσταση ενός IDPS σε κατάσταση λειτουργίας inline. Ο sensor παρεμβάλλεται στην κυκλοφορία των πακέτων στο δίκτυο μέσω δύο διεπαφών (monitoring interface) που εξασφαλίζουν πως όλα τα εισερχόμενα και τα εξερχόμενα πακέτα θα περάσουν από αυτόν. Επιπλέον, διαθέτει μια ξεχωριστή διεπαφή διαχείρισης (management interface) για τη ροή των πληροφοριών από και προς τα άλλα συστατικά μέρη του IDPS.

- Σε κατάσταση *passive* ο sensor λαμβάνει και ελέγχει ένα αντίγραφο κάθε πακέτου που διακινείται στο δίκτυο, χωρίς να παρεμβάλλεται στην κυκλοφορία του.



Εικόνα 9: IDPS σε κατάσταση passive
[πηγή: (Scarfone & Mell, 2007)]

Στην παραπάνω εικόνα από το (Scarfone & Mell, 2007) παρουσιάζεται η εγκατάσταση ενός IDPS σε κατάσταση λειτουργίας passive. Σε αυτή την περίπτωση το switch που βρίσκεται μετά το firewall, μέσω ειδικής θύρας, κάνει broadcast στον sensor όλα πακέτα που φτάνουν σε αυτό. Και σε αυτή την περίπτωση ο sensor είναι μια ξεχωριστή συσκευή που επικοινωνεί με τα υπόλοιπα συστατικά μέρη του IDPS. Στη συγκεκριμένη διαμόρφωση αξίζει να παρατηρηθεί πως αξιοποιούνται επιπρόσθετοι sensor «πίσω» από έναν IDPS load balancer. Η τοποθέτηση που balancer «πριν» το firewall έχει ως αποτέλεσμα αυτοί οι sensor να αναλύουν όλα τα πακέτα που διακινούνται από και προς το Διαδίκτυο. Με αυτή τη διαμόρφωση, το IDPS έχει πλήρη γνώση των πακέτων πριν και μετά το φιλτράρισμα του firewall, συνεπώς μπορεί να αναγνωρίσει ποιες απειλές αντιμετωπίζονται επιτυχώς και ποιες όχι.

Και οι δύο καταστάσεις λειτουργίας εμφανίζουν την ίδια μεγάλη αδυναμία: τη διαχείριση ενός δικτύου με υψηλή κυκλοφορία. Ένας inline sensor είναι επιρρεπής στην πρόκληση ενός bottleneck effect· η αυξημένη κίνηση μετατρέπει τον κόμβο σε σημείο συμφόρησης, που πασχίζει να ελέγξει και να αναλύσει τα πακέτα ταχύτερα από ότι αυτά καταφτάνουν. Σε ήπια μορφή κάτι τέτοιο μπορεί να επιφέρει μια μείωση της απόδοσης, με επιβράδυνση της εξυπηρέτησης από τους κόμβους που βρίσκονται «πίσω» από τον sensor· στη χειρότερη μορφή μπορεί να εξελιχθεί σε DoS. Από την άλλη, ένας passive sensor δεν εμποδίζει τη ροή των πακέτων προς τους τελικούς τους προορισμούς. Ωστόσο, για να ανταποκριθεί στην αυξημένη κυκλοφορία πακέτων, ίσως χρειαστεί να απορρίψει μερικά πακέτα· αυτό όμως μπορεί να προκαλέσει τη μη ανίχνευση ενός περιστατικού – αυτό είναι ένα ιδιαίτερο πρόβλημα για τον μηχανισμό ανίχνευσης που βασίζεται στην ανάλυση πρωτοκόλλων.

Στην πράξη, inline τοποθετούνται οι sensor των NIPS, ώστε να είναι σε θέση να μπλοκάρουν την κακόβουλη κυκλοφορία που θα εντοπίσουν [152]. Οι sensor των NIDS, από την άλλη, τοποθετούνται σε κατάσταση passive, ώστε να μπορούν να παρακολουθούν σημεία με σημαντικούς πόρους του εσωτερικού δικτύου. (Scarfone & Mell, 2007; Ashoor & Gore, 2011)

[152] Στην ενότητα «Λειτουργίες απόκρισης των IPS» αναλύεται ο λόγος για τον οποίο προτιμάται τα IPS να τοποθετούνται inline.

6.6 *Βασικοί μηχανισμοί και πρόσθετες δυνατότητες*

Όπως εξηγήθηκε στις προηγούμενες ενότητες, η λειτουργία τόσο των IDS, όσο και των IPS περιστρέφεται γύρω από τη διεργασία της ανίχνευσης περιστατικών. Συμπληρωματικά προς τον μηχανισμό ανίχνευσης, τα IDPS αξιοποιούν μια σειρά από άλλους μηχανισμούς για την περάτωση του έργου τους. Ο βασικότερος ίσως μετά τον μηχανισμό ανίχνευσης είναι ο μηχανισμός *καταγραφής* (*logging*) των δεδομένων που συσχετίζονται με ένα γεγονός. Τυπικά, τα δεδομένα αυτά περιλαμβάνουν την ημερομηνία και την ώρα, το είδος του γεγονότος, κάποια βαθμονόμηση της σημασίας του και, αν εφαρμόστηκαν, τα μέτρα αποτροπής. Ανάλογα με τον τύπο του IDPS καταγράφονται πρόσθετα πεδία, όπως για παράδειγμα ένα userID σε ένα host-based IDPS ή ακόμα και ολόκληρα πακέτα που κρίνονται σημαντικά σε ένα network-based IDPS. Αυτές οι καταγραφές έχουν πολλές χρήσεις: την επαλήθευση μιας ανίχνευσης, τη διερεύνηση ενός περιστατικού, τη συσχέτιση γεγονότων με δεδομένα που έχουν καταγραφεί σε άλλες πηγές κ.α. (Scarfone & Mell, 2007; Azeez, et al., 2020)

Ένας άλλος μηχανισμός των IDPS είναι ο μηχανισμός *ειδοποίησης* για τις ανιχνεύσεις που πραγματοποιούνται. Η παράδοση των ειδοποιήσεων (notifications ή alerts) μπορεί να γίνει με διάφορες μεθόδους: με αποστολή email, με προβολή pop-up μηνυμάτων ή παραγωγή σελίδων στην κονσόλα του IDPS, με εκτέλεση καθορισμένων σεναρίων εντολών (script) ή προγραμμάτων, με αξιοποίηση του πρωτοκόλλου διαχείρισης δικτύων SNMP ή του πρωτοκόλλου syslog κ.α. Τυπικά αυτές οι ειδοποιήσεις είναι αρκετά σύντομες και περιέχουν μόνο τις βασικές πληροφορίες για το γεγονός που παρατηρήθηκε και ο εκάστοτε αναλυτής ασφάλειας θα πρέπει να αναζητήσει περισσότερες λεπτομέρειες στο IDPS. (Scarfone & Mell, 2007)

Συμπληρωματικά προς τον μηχανισμό ειδοποίησης, αξιοποιείται ο μηχανισμός *παραγωγής αναφορών* (reports), που παρέχει ή κάποιου είδους σύνοψη των γεγονότων που παρατηρήθηκαν, ή/και εκτενείς πληροφορίες για ένα συγκεκριμένο γεγονός. Αυτές οι πληροφορίες προέρχονται από την επεξεργασία των δεδομένων του μηχανισμού καταγραφής και μπορούν να φτάνουν σε μεγάλο βάθος· ως το σημείο το σύστημα να συσχετίζει το γεγονός με μια γνωστή ευπάθεια CVE – μετά την αναγνώριση του προβλήματος η διευθέτηση γίνεται πιο απλή υπόθεση, αφού πηγές πληροφόρησης, όπως η NVD ή η λίστα CVE, μπορούν να παρέχουν οδηγίες σχετικά με την επίλυσή του. Είναι σημαντικό να παρατηρήσουμε ότι οι μηχανισμοί αυτοί είναι εξαιρετικής σημασίας για τα συστήματα IDS, καθώς μέσα από αυτούς θα καταφέρει ένας αναλυτής ασφάλειας να αντλήσει την απαραίτητη γνώση για να διερευνήσει αρχικά, και στη συνέχεια να αντιμετωπίσει αποτελεσματικά τα διάφορα περιστατικά. (McHugh, 2001; Scarfone & Mell, 2007; Ashoor & Gore, 2011)

Αυτοί οι βασικοί μηχανισμοί των συστημάτων IDPS προσφέρουν πρόσθετες δυνατότητες, πέρα από την αντιμετώπιση επιθέσεων. Καταρχάς, αξιοποιώντας πληροφορίες από ένα IDPS, ένας οργανισμός έχει τη δυνατότητα *αναγνώρισης προβλημάτων στις πολιτικές ασφάλειας*. Σε αυτή την περίπτωση, ένα IDPS που έχει ρυθμιστεί να εντοπίζει (και) πακέτα που θα έπρεπε να απορρίπτονται από το firewall, μπορεί να βοηθήσει στον εντοπισμό λαθών της παραμετροποίησής του. (Scarfone & Mell, 2007)

Μια άλλη δυνατότητα είναι *τεκμηρίωση μιας υπαρκτής απειλής* για τον οργανισμό. Αυτό γίνεται αξιοποιώντας τις εκτενείς καταγραφές των επιθέσεων που δέχεται ο οργανισμός. Οι καταγραφές επιτρέπουν τη μελέτη της συχνότητας και των χαρακτηριστικών των επιθέσεων, με σκοπό να αναζητηθούν και να εφαρμοστούν τα κατάλληλα μέτρα ασφαλείας που θα εξασφαλίσουν την προστασία των πόρων του οργανισμού. (Scarfone & Mell, 2007)

Ακόμα, με έμμεσο τρόπο επιτυγχάνεται η *αποτροπή της παραβίασης των πολιτικών ασφάλειας από άτομα εντός του οργανισμού*. Ακόμα και με καλό σκοπό (λ.χ. επιτάχυνση της εργασίας τους), είναι συνηθισμένο υπάλληλοι ενός οργανισμού να παρεκκλίνουν από τις πολιτικές ασφάλειας (με αποτέλεσμα πολλές φορές να εκθέτουν το σύστημα σε κίνδυνο). Η γνώση πως κάθε παραβίαση παρακολουθείται και καταγράφεται αυτόματα, αποτελεί ένα ισχυρό κίνητρο για την αποφυγή τέτοιων δυνητικά επιζήμιων πρακτικών. (Scarfone & Mell, 2007)

6.7 Λειτουργίες απόκρισης των IPS

Όσα αναφέρθηκαν μέχρι στιγμής είναι δυνατότητες που παρέχονται τόσο από τα IDS, όσο και από τα IPS. Όμως, αναφέρθηκε ήδη πως τα συστήματα IPS διαφέρουν, γιατί επιπλέον προσφέρουν λειτουργίες αυτόματης απόκρισης σε κάποιο περιστατικό που έχει ανιχνευθεί και επιχειρούν να λάβουν αντίμετρα, όσο ακόμα το περιστατικό εξελίσσεται, με σκοπό να αποτρέψουν τις επιπτώσεις στο σύστημα. Από το έργο των Scarfone & Mell (2007) μπορούμε να αντλήσουμε πληροφορίες για πολλές τεχνικές που εφαρμόζονται στα IPS.

Μία προσέγγιση είναι *το IPS να επιχειρήσει να σταματήσει μια επίθεση μόνο του*. Για να το επιτύχει αυτό, μπορεί:

- Να τερματίσει μια δικτυακή σύνδεση μεταξύ κάποιων κόμβων ή μια σύνδεση χρήστη που χρησιμοποιείται στην επίθεση.
- Να εμποδίσει την πρόσβαση σε έναν ή περισσότερους στόχους από έναν λογαριασμό ή μια διεύθυνση IP που χρησιμοποιείται στην επίθεση.
- Να εμποδίσει κάθε πρόσβαση σε έναν κόμβο, υπηρεσία, εφαρμογή ή άλλον πόρο.

Άλλη προσέγγιση είναι *το IPS να επαναρρυθμίσει άλλες συσκευές* με σκοπό να διακόψει μια επίθεση. Για να επιτύχει αυτό τον στόχο, μπορεί:

- Να ρυθμίσει ένα firewall, έναν δρομολογητή ή ένα switch ώστε να εμποδίσει την πρόσβαση από έναν συγκεκριμένο επιτιθέμενο ή γενικά προς τον κόμβο – στόχο.
- Να τροποποιήσει τους κανόνες φιλτραρίσματος πακέτων στο firewall του κόμβου – στόχου για να εμποδίσει την εισερχόμενη κυκλοφορία μιας επίθεσης.
- Να προκαλέσει την εγκατάσταση ενημερώσεων ασφαλείας σε κάποιον κόμβο που έχει ανιχνεύσει ευπάθειες.

Μια τρίτη προσέγγιση είναι *το IPS να αλλάξει το περιεχόμενο της επίθεσης*. Έτσι μπορεί:

- Να αφαιρέσει ένα μολυσμένο συνημμένο σε ένα email πριν παραδοθεί στον παραλήπτη του.
- Να δράσει ως proxy server και να απενθυλακώσει και στη συνέχεια να επανενθυλακώσει τα περιεχόμενα των εισερχόμενων πακέτων – αυτό θα είχε ως αποτέλεσμα να αποτύχουν κάποιες επιθέσεις που βασίζονται σε παραποιημένες επικεφαλίδες πακέτων.

Οι προσεγγίσεις που μπορούν να εφαρμοστούν εξαρτώνται από την κατάσταση λειτουργίας του IPS. Ένα passive IPS μπορεί να αξιοποιήσει την προσέγγιση της επαναρρύθμισης άλλων συσκευών για να διακόψει την επίθεση (είτε άμεσα, είτε με εκτέλεση κάποιου καθορισμένου προγράμματος). Εναλλακτικά, μπορεί να αποστείλει πακέτα TCP reset connection για να τερματίσει μια κακόβουλη σύνδεση. Αδυναμία της τεχνικής είναι πως δουλεύει μόνο για το TCP και όχι για άλλα πρωτόκολλα όπως το UDP.

Τα inline IPS από την άλλη, μπορούν να εφαρμόσουν και τις τρεις προσεγγίσεις αναλύθηκαν παραπάνω. Και αυτά μπορούν να αλλάζουν τη ρύθμιση άλλων συσκευών ($2^{η}$ προσέγγιση), αλλά επιπλέον, έχουν το πλεονέκτημα ότι παρεμβάλλονται στη μετάδοση των πακέτων. Με αυτόν τον τρόπο μπορούν να τα φιλτράρουν (και να τερματίζουν συνδέσεις κόμβων, εφαρμογών ή χρηστών απορρίπτοντάς τα – $1^{η}$ προσέγγιση) ή να αλλάζουν το περιεχόμενό τους ($3^{η}$ προσέγγιση). Για τον λόγο αυτό η κατάσταση λειτουργίας inline κρίνεται ως η πλέον αποδοτική και έχει επικρατήσει στα IPS.

6.8 Διαφορές IDS και IPS

Στο κεφαλαίο αυτό αναλύθηκαν διάφορα χαρακτηριστικά των συστημάτων IDPS. Και παρότι υπάρχει πολύ κοινό έδαφος μεταξύ των IDS και των IPS, υπάρχουν και διαφορές που πρέπει να επισημανθούν, καθώς επηρεάζουν τον τρόπο που χρησιμοποιούνται.

Χρησιμοποιώντας ως οδηγό τη σύγκριση των IDS και IPS από το A Spire Research Report της εταιρείας Spire Security, όπως εμφανίζεται στο (Ashoor & Gore, 2011, σ. 500), αλλά και τα έργα των ίδιων των Ashoor & Gore (2011) και των Scarfone & Mell (2007), μπορούμε να συνθέσουμε τον κάτωθι πίνακα που παρουσιάζει επιγραμματικά κάποια πολύ σημαντικά χαρακτηριστικά των δύο τύπων συστημάτων:

Τύπος IDPS	IDS	IPS
Κατάσταση λειτουργίας	passive	inline
Απόκριση σε περιστατικά	μη αυτόματη (απαραίτητη η ανθρώπινη παρέμβαση)	αυτόματη
Αποτέλεσμα μιας πιθανής διακοπής λειτουργίας	διακοπή της παρακολούθησης, όμως οι δικτυακές δραστηριότητες συνεχίζονται κανονικά	διακοπή κάθε δικτυακής δραστηριότητας
Χρόνος ανάλυσης	σε πραγματικό χρόνο ή σχεδόν πραγματικό χρόνο (ιδανικά), γενικά ανεκτή κάποια καθυστέρηση (τυπικά δευτερολέπτων ως και μερικών λεπτών), η παραλαβή πακέτων σε δέσμες [153] επιφέρει μεγαλύτερη καθυστέρηση (αρκετών λεπτών ως και ωρών)	σε πραγματικό χρόνο ή σχεδόν πραγματικό χρόνο (μια πιθανή επιβράδυνση μπορεί να προκαλέσει διακοπή νόμιμων δραστηριοτήτων)
Χρόνος απόκρισης	εξαρτώμενος από την ταχύτητα απόκρισης του διαχειριστή ασφαλείας	σε πραγματικό χρόνο ή σχεδόν πραγματικό χρόνο (μια πιθανή επιβράδυνση μπορεί να προκαλέσει διακοπή νόμιμων δραστηριοτήτων)
Αντίκτυπος ψευδώς θετικών ανιχνεύσεων	ο αναλυτής ασφάλειας επιβαρύνεται με διερεύνηση ανύπαρκτων επιθέσεων	προκαλείται διακοπή νόμιμων δραστηριοτήτων
Αντίκτυπος ψευδώς αρνητικών ανιχνεύσεων	οι επιθέσεις εκτελούνται επιτυχώς	οι επιθέσεις εκτελούνται επιτυχώς

Πίνακας 2: Σύγκριση IDS και IPS

[153] Αυτό αφορά κυρίως τον τύπο Network Behavior Analysis (δείτε την ενότητα «Πηγές δεδομένων ελέγχου και τύποι συστημάτων»).

Ένα σημείο που αξίζει να τονιστεί ιδιαίτερα είναι το εξής: παρατηρώντας τον παραπάνω πίνακα μπορούμε να συμπεράνουμε ότι οι επιπτώσεις για ένα σύστημα που βασίζεται στη χρήση IDS γίνονται μεγαλύτερες όσο κινούμαστε χαμηλότερα [154] – και φτάνουμε στο χειρότερο δυνατό αποτέλεσμα όταν μια επίθεση εκτελεστεί επιτυχώς. Τα πράγματα λειτουργούν αντίθετα για ένα IPS, αφού, αν παρατηρήσουμε προσεκτικά, η καλύτερη επίπτωση είναι όταν μια επίθεση εκτελεστεί επιτυχώς (!) [155]. Όσο κινούμαστε ψηλότερα στον πίνακα, τόσο αυξάνονται οι πιθανότητες να επηρεαστούν οι νόμιμες δραστηριότητες στο σύστημα, ως το σημείο που να διακοπούν πλήρως όλες οι δραστηριότητες (νόμιμες και κακόβουλες) στο δίκτυο.

Αυτό οφείλεται στο γεγονός ότι ένα IPS τοποθετείται σε κατάσταση inline για να μπορέσει να μπλοκάρει με τον πιο αποδοτικό τρόπο τις κακόβουλες δραστηριότητες που ανιχνεύει· με αυτόν τον τρόπο όμως μετατρέπεται σε ένα single point of failure για ολόκληρο το σύστημα. Τα IDS δεν έχουν αυτό το πρόβλημα, γιατί οι λειτουργίες που επιτελούν μπορούν να εκτελεστούν χωρίς να παρεμβάλλονται στη μετάδοση των πακέτων, συνεπώς αποδίδουν πλήρως σε κατάσταση passive.

Ένα τελευταίο σημείο που πρέπει να αναφερθεί είναι πως η παραπάνω σύγκριση με τις τιμές που εμφανίζονται στον πίνακα, αφορά κυρίως τα NIDS και τα NIPS. Τα HIDS και τα HIPS έχουν μεγαλύτερη ομοιογένεια στις ιδιότητές τους και διαφέρουν μόνο στο κομμάτι που έχει σχέση με τη διαχείριση των ενεργειών απόκρισης. Είναι όμοια όσον αφορά τον χρόνο ανάλυσης (λειτουργούν σε πραγματικό χρόνο ή σχεδόν πραγματικό χρόνο), ενώ επίσης μια πιθανή διακοπή λειτουργίας επηρεάζει μόνο τον συγκεκριμένο υπολογιστικό κόμβο, αλλά δεν επηρεάζει κανέναν άλλο.

Στην πράξη, στα μεγάλα πληροφοριακά συστήματα ειδικότερα, τα δύο είδη λειτουργούν σε συνδυασμό. Ένας IPS sensor τοποθετείται «πίσω» από το firewall, είτε ως αυτόνομη συσκευή, είτε με τη μορφή υβριδικής τεχνολογίας που ενσωματώνεται στο ίδιο το firewall και ελέγχει τα φιλτραρισμένα πακέτα του πριν επιτρέψει να περάσουν στο εσωτερικό δίκτυο. Σημαντικοί υπολογιστικοί κόμβοι του συστήματος προστατεύονται από έναν IDS ή IPS agent. Ένα πλήθος IDS sensor τοποθετείται σε σημεία που να μπορούν να εποπτεύουν την κυκλοφορία των εσωτερικών κόμβων. Ενδεχομένως κάποιοι IDS sensor έχουν επιφορτιστεί την εποπτεία της κυκλοφορίας πριν ακόμα να φιλτραριστεί από το firewall. Σε ένα τέτοιο σύστημα, σκοπός είναι όλοι αυτοί οι κόμβοι (IPS και IDS, host-based και network-based) να τροφοδοτούν διαρκώς έναν management server που θα έχει πλήρη γνώση ποιες απειλές δέχεται το σύστημα από το εξωτερικό του, ποιες φιλτράρονται από το firewall και ποιες καταφέρνουν να το περάσουν, ποιες μπλοκάρονται από το IPS και τελικά σε ποιους κόμβους προσπαθούν να φτάσουν. Με αυτό τον τρόπο μπορεί δυναμικά, είτε με αυτόματο τρόπο, είτε με ανθρώπινη παρέμβαση να γίνεται ρύθμιση όλων αυτών των μερών, ώστε να εξασφαλιστεί πώς οι πολιτικές ασφαλείας του συστήματος λειτουργούν σωστά και κάθε απειλή αντιμετωπίζεται έγκαιρα.

[154] Δεν υφίστανται επιπτώσεις για τις ιδιότητες που αναφέρονται στις πρώτες 2 σειρές του πίνακα, οπότε δεν λαμβάνονται υπόψη.

[155] Προφανώς υπάρχει μια δόση υπερβολής σε αυτή τη διατύπωση, για να τονιστεί ότι κάποια αρνητική επίδραση επί της νόμιμης δραστηριότητας είναι εντελώς ανεπιθύμητη.

6.9 Συνοψίζοντας τη σχέση των IDPS με άλλα συστήματα

Στο προηγούμενο κεφάλαιο εξηγήθηκε γιατί τα βασικά συστήματα κυβερνοάμυνας είναι ανεπαρκή απέναντι στην πληθώρα επιθέσεων των επίδοξων εισβολέων. Σε αυτό το κεφάλαιο αναλύθηκε η λειτουργία των IDPS και πώς αυτά μπορούν να ανιχνεύσουν κακόβουλες δραστηριότητες – ακόμα και να τις εξουδετερώσουν. Αυτό που πρέπει να τονιστεί είναι πως τα συστήματα IDPS δεν δρουν ανταγωνιστικά, αλλά συμπληρωματικά προς τα προγενέστερα.

Ξεκινώντας με ένα σύστημα firewall, αξίζει να υπενθυμίσουμε πως τυπικά τοποθετείται ανάμεσα στο εσωτερικό δίκτυο και το Διαδίκτυο για να φιλτράρει την εισερχόμενη και την εξερχόμενη κυκλοφορία. Και σε γενικές γραμμές κάνει τη δουλειά του καλά. Η κυριότερη αδυναμία του είναι ότι μπορεί να ελέγξει πακέτα που φτάνουν στην περίμετρο του συστήματος και όχι αυτά που κυκλοφορούν στο εσωτερικό δίκτυο. Επιπλέον, αν ένας επιτιθέμενος καταφέρει να προσβάλει έναν εσωτερικό κόμβο (λ.χ. με μια επίθεση phishing), τότε αρκεί να βρει και έναν τρόπο ώστε να μην παραβιάζει τους τις πολιτικές του firewall, για να μπορέσει να δράσει ανενόχλητος στο εσωτερικό δίκτυο.

Από την άλλη, ένα σύστημα antimalware ασχολείται αποκλειστικά με την παρακολούθηση των προγραμμάτων που εκτελούνται σε έναν συγκεκριμένο υπολογιστικό κόμβο και αγνοεί τελείως τι συμβαίνει στο υπόλοιπο δίκτυο. Η προσέγγιση αυτή δεν είναι ιδιαίτερα αποτελεσματική καθώς δεν προστατεύει το σύστημα έναντι μιας μη εξουσιοδοτημένης πρόσβασης από έναν επιτιθέμενο. Σε περίπτωση που ένας επιτιθέμενος καταφέρει να εισβάλει σε έναν κόμβο, μπορεί να εφαρμόσει διάφορες τακτικές που θα του επιτρέψουν, μεταξύ άλλων, να εγκαταστήσει κακόβουλο λογισμικό που θα εκτελείται «κρυφά» από το antimalware.

Τα συστήματα IDPS έρχονται να καλύψουν το κενό που αφήνουν τα δύο αυτά συστήματα λειτουργώντας στο εσωτερικό δίκτυο του πληροφοριακού συστήματος. Ένα HIDS ή ένα HIPS παρακολουθεί έναν συγκεκριμένο κόμβο, γενικεύοντας τη φιλοσοφία των antimalware πέρα από τα προγράμματα, σε κάθε μορφή δεδομένων που καταφτάνει στον κόμβο αυτόν με σκοπό να αναγνωριστεί κάποια εισβολή, είτε αυτή έχει τη μορφή κακόβουλου προγράμματος, είτε τη μορφή μη εξουσιοδοτημένης πρόσβασης από κάποιον κακόβουλο παράγοντα (εξωτερικό ή ακόμα και εσωτερικό χρήστη). Μεγάλο μειονέκτημα αυτής της προσέγγισης είναι το πρόσθετο υπολογιστικό κόστος για την παρακολούθηση κάθε λειτουργίας που εκτελείται στον κόμβο.

Ένα NIDS ή ένα NIPS ακολουθεί άλλη προσέγγιση και επεκτείνοντας τη φιλοσοφία των firewall, παρακολουθεί και ελέγχει όλα τα πακέτα που κυκλοφορούν στο εσωτερικό δίκτυο. Αυτά τα συστήματα αναζητούν εκείνα τα πακέτα που δεν ικανοποιούν τις καθορισμένες πολιτικές ασφαλείας. Πλεονέκτημα αυτής της προσέγγισης είναι πως η προστασία επεκτείνεται σε πολλούς κόμβους ταυτόχρονα. Μειονέκτημα είναι πως αδυνατεί να προστατέψει από επιθέσεις που πηγάζουν από το εσωτερικό κάποιων κόμβων και δεν επεκτείνονται στο δίκτυο.

Μια τρίτη προσέγγιση αφορά το συνδυασμό των δύο προηγούμενων και υλοποιείται με την ύπαρξη ενός κεντρικού κόμβου εποπτείας που λαμβάνει feedback τόσο από HIDS/HIPS που εποπτεύουν μεμονωμένα υπολογιστικούς κόμβους, όσο και από NIDS/NIPS που εποπτεύουν την κυκλοφορία στο εσωτερικό δίκτυο με σκοπό να εντοπίσει τις εισβολές.

6.10 Επιθέσεις κατά IDPS

Τα συστήματα IDPS μπορεί να δημιουργήθηκαν για να προστατέψουν τα πληροφοριακά συστήματα από απειλές, αλλά είναι και τα ίδια πολλές φορές ευάλωτα σε επιθέσεις. Όπως ειπώθηκε ξανά, υπάρχει περίπτωση ένας επιτιθέμενος να αποφασίσει να τα θέσει εκτός λειτουργίας και στη συνέχεια να εισβάλει στο σύστημα (McHugh, 2001) χωρίς να καταγράφονται οι δράσεις του.

Ένας τρόπος είναι επιθέσεις DDoS ή επιθέσεις στις οποίες αποστέλλονται πακέτα σε πολλά κομμάτια (fragments) με σκοπό να εξαντληθούν οι υπολογιστικοί πόροι του IDPS και να το οδηγήσουν σε αντικανονικό τερματισμό της λειτουργίας του. Μια άλλη τεχνική ονομάζεται *τύφλωση* (*blinding*) και λειτουργεί παράγοντας ένα τεράστιο πλήθος ειδοποιήσεων που θα κατακλύσουν το IDPS. Η επίθεση αυτή δεν έχει ουσιαστικά κάποιον υπολογιστικό κόμβο ως στόχο. Σκοπός είναι είτε να διακοπεί η λειτουργία του IDPS, είτε το πλήθος των ειδοποιήσεων να προσφέρει «κάλυψη» στην «αληθινή» επίθεση που πραγματοποιείται παράλληλα με την επίθεση τύφλωσης.

Σε αυτές τις περιπτώσεις υπάρχουν μηχανισμοί που βοηθούν τους sensor να αναγνωρίσουν τέτοιες επιθέσεις DDoS ή τύφλωσης και επιτρέπουν μετά την αποστολή της αρχικής ειδοποίησης να αγνοούνται όλα τα επόμενα πακέτα της επίθεσης για να μειωθεί ο φόρτος στον sensor.

(Scarfone & Mell, 2007)

6.11 Διαμοιρασμός πληροφοριών για τις κυβερνοαπειλές

Έχοντας αναλύσει πώς τα Συστήματα Ανίχνευσης και Πρόληψης Εισβολής έρχονται να συμπληρώσουν και να πλαισιώσουν τους κύριους μηχανισμούς και συστήματα της κυβερνοάμυνας, προκύπτει ένα ερώτημα: «*πώς μπορεί η κυβερνοασφάλεια να γίνει ακόμα πιο ισχυρή;*». Η απάντηση σε αυτό το ερώτημα μπορεί να δοθεί από μια φιλοσοφία που προωθείται ιδιαίτερα από τις Τεχνολογίες Ανοικτού Κώδικα, τη φιλοσοφία του διαμοιρασμού ιδεών και, το κυριότερο, γνώσης.

Για να κατανοήσουμε πώς η φιλοσοφία αυτή μπορεί να εφαρμοστεί στην κυβερνοασφάλεια, ας σκεφτούμε το εξής σενάριο: διαθέτουμε ένα πληροφοριακό σύστημα για την προστασία του οποίου έχουν εφαρμοστεί όλα τα απαραίτητα μέτρα. Όμως, μια ευπάθεια σε κάποιο μέρος τους συστήματος, άγνωστη σε εμάς και τον κατασκευαστή του, δίνει σε έναν επιδέξιο επιτιθέμενο την ευκαιρία να παρακάμψει τους μηχανισμούς άμυνας και να εισβάλει στο σύστημά μας. Ο εισβολέας θα προχωρήσει σε μια σειρά από παράτυπες ενέργειες, ώσπου, κάποια στιγμή, κάποια από αυτές να ανιχνευτεί από το IDS ή το IPS (ή κάποιος αναλυτής ασφάλειας να διαπιστώσει ότι κάτι δεν πάει καλά). Με βάση τις σχετικές ειδοποιήσεις και αναφορές, ο αναλυτής ασφάλειας θα διερευνήσει το περιστατικό και θα ανακαλύψει ότι πράγματι υπάρχει κακόβουλη δραστηριότητα. Το επόμενο βήμα είναι να βρει έναν τρόπο να στερήσει την πρόσβαση στον εισβολέα. Και μετά να ξεκινήσει έναν αγώνα αναψηλάφησης για να διαπιστώσει τι ζημία έχει γίνει, αλλά και να βεβαιωθεί ότι δεν έχει δημιουργηθεί κάποιος εναλλακτικός δίαυλος για να επανέλθει ο εισβολέας στο σύστημα. Και τέλος, να αρχίσει μια προσπάθεια αποκατάστασης της όποιας ζημιάς.

Ακόμα και αν έχει εκδιωχθεί από αυτό το σύστημα, ο εισβολέας έχει βρει έναν τρόπο για να προσβάλει παρόμοια συστήματα. Ή να αποκομίσει οικονομικά οφέλη, πουλώντας αυτή τη γνώση σε άλλους κακόβουλους παράγοντες που ενδιαφέρονται να κάνουν το ίδιο.

Και εύλογα μπορεί κάποιος να αναρωτηθεί τα εξής: «*αν μοιραζόμουν τη γνώση που απέκτησα από το περιστατικό, μήπως κάποιος άλλος αποσοβούσε τη ζημία που έπαθα εγώ;*» ή αντίστροφα «*αν κάποιος άλλος είχε παρόμοιο περιστατικό πριν από εμένα, μήπως αν μοιραζόταν τη γνώση να μπορούσα να αποσοβήσω τη ζημία;*». Και τελικά «*αφού οι επιτιθέμενοι ανταλλάσσουν γνώση, γιατί να μην το κάνουν και οι αμυνόμενοι;*».

Αυτή η ιδέα, του διαμοιρασμού της γνώσης, υπάρχει διάχυτη σε πολλά σημεία της παρούσας εργασίας: στην έρευνα γύρω από το κακόβουλο λογισμικό, στα μοντέλα μελέτης και κατανόησης των κυβερνοεπιθέσεων (όπως το Cyber Kill Chain και το ATT&CK), αλλά και στην ανάπτυξη και ανοικτή διάθεση εργαλείων οργάνωσης της κυβερνοάμυνας (όπως οι λίστες CWE και CVE, η βάση δεδομένων NVD και το πλαίσιο D3FEND). Και η γνώση αυτή προέρχεται και παρέχεται όχι μόνο από την ακαδημαϊκή κοινότητα, αλλά και από άμεσα ενδιαφερόμενα μέρη που κατανοούν πως ο διαμοιρασμός της γνώσης είναι προς όφελός τους.

Όλα αυτά τα εγχειρήματα μπορούν να ενταχθούν στο πλαίσιο μιας στρατηγικής που ονομάζεται **Cyber Threat Intelligence** (**CTI**) [156]. Γενικά, το CTI εστιάζει στις δυνατότητες, τα κίνητρα και τους σκοπούς των επιτιθέμενων, καθώς και στον τρόπο που μπορούν να επιτευχθούν. (Ramsdale, Shiaeles, & Kolokotronis, 2020) Η πληροφόρηση γύρω από αυτά τα θέματα προέρχεται από μια πληθώρα από τεχνολογίες που σκοπό έχουν τη συλλογή, την εξέταση, την ανάλυση και το διαμοιρασμό ψηφιακών πειστηρίων που σχετίζονται με ένα συγκεκριμένο περιστατικό ασφαλείας· με απώτερο στόχο αυτό το περιστατικό να μην επαναληφθεί κάπου αλλού. (Preuveneers & Joosen, 2021) Απαραίτητο όμως παραμένει, η γνώση που αποκομίζεται να μπορεί να μετατραπεί σε ενέργειες που θα εφαρμοστούν από τα συστήματα κυβερνοάμυνας για να αντιμετωπιστούν οι κυβερνοεπιθέσεις. (Ramsdale, Shiaeles, & Kolokotronis, 2020)

Το CTI είναι ένας αρκετά εκτενής τομέας της Ασφάλειας Πληροφοριών και μπορεί να λάβει πολλές διαφορετικές μορφές. Αρκετές πτυχές του εξετάστηκαν στην παρούσα εργασία (διανύσματα επιθέσεων, είδη επιθέσεων, μοντέλα μελέτης επιθέσεων κ.α.). Όσον αφορά το τεχνικό κομμάτι του CTI, υπάρχει μια σημαντική πτυχή του, που είναι οι **ενδείξεις προσβολής** (**indicators of compromise** – **IoC**) [157]. Τα IoC είναι δεδομένα, που μπορούν να λειτουργήσουν ως ψηφιακά πειστήρια και μπορούν να αποδώσουν λεπτομέρειες για κακόβουλες δραστηριότητες σε ένα σύστημα ή ένα δίκτυο. Ως παραδείγματα μπορούμε να αναφέρουμε τη διεύθυνση IP από την οποία πραγματοποιείται μια επίθεση, το URL ενός ιστότοπου που χρησιμοποιείται σε μια επίθεση phishing, το hash ενός αρχείου που έχει αναγνωριστεί ως κακόβουλο, τον αποστολέα ενός email, το θέμα του, ένα συνημμένο του και πολλά άλλα. (Preuveneers & Joosen, 2021)

Όπως γίνεται κατανοητό, τα IoC ως ψηφιακά πειστήρια μπορούν να αναζητηθούν και να εξαχθούν μέσα από τους μηχανισμούς καταγραφής και αναφορών των IDPS [158]. Όμως τα IDPS δεν λειτουργούν απλά ως πηγή για την παραγωγή των IoC, αλλά και ως προορισμός για την «κατανάλωσή» τους· τα IoC που παράχθηκαν με βάση τις ανιχνεύσεις σε ένα σύστημα IDPS θα μπορούσαν να αξιοποιηθούν για να παραχθούν νέοι κανόνες ανίχνευσης σε ένα άλλο IDPS, με σκοπό να προληφθούν τα ίδια περιστατικά προτού εξελιχθούν σε εισβολή. [159]

Παρότι ο διαμοιρασμός μπορεί να προσφέρει σημαντικά οφέλη για τους οργανισμούς που θα τον αξιοποιήσουν, υπάρχει πάντα το πρόβλημα του αποτελεσματικού χειρισμού των πληροφορίων, καθώς αυτές προέρχονται από πολλές, και κυρίως, ετερογενείς πηγές. (Guarascio, Cassavia, Pisani, & Manco, 2022)

[156] Στο συγκεκριμένο εννοιολογικό πλαίσιο, ο όρος *intelligence* αποδίδεται στα ελληνικά ως *πληροφορίες*. Όμως, έτσι αποδίδεται και ο όρος *information*, που είναι βασική έννοια της επιστήμης της Πληροφορικής και των Ηλεκτρονικών Υπολογιστών γενικότερα. Για την αποφυγή της σύγχυσης, η παρούσα εργασία θα χρησιμοποιεί συχνά το αρκτικόλεξο CTI.

[157] Πρέπει να τονιστεί ωστόσο ότι το CTI δεν βασίζεται αποκλειστικά σε IoC. Άλλες μορφές του CTI σχετίζονται με τις περιγραφές τακτικών, τεχνικών, περιστατικών, ακόμα και των ίδιων των κακόβουλων παραγόντων που έχουν αναγνωριστεί.

[158] Τα IDPS δεν είναι η μοναδική πηγή για την παραγωγή IoC, αλλά, δεδομένου του ρόλου τους στην ανίχνευση κακόβουλων δραστηριοτήτων, είναι σίγουρα από τις πιο σημαντικές.

[159] Ένα σημείο που χρειάζεται πολλή προσοχή είναι το γεγονός ότι τα IoC αποτελούν απλώς μια υπόνοια απειλής και μπορούν να γίνουν ανακριβή με την πάροδο του χρόνου (Preuveneers & Joosen, 2021)

Σε αυτό το σημείο προκύπτει η ανάγκη για ένα εργαλείο διαμοιρασμού των IoC (ή των CTI γενικότερα)· αυτό το εργαλείο είναι μία **Πλατφόρμα Διαμοιρασμού Πληροφοριών για τις Απειλές** (**Threat Intelligence Platform** – **TIP**), που σκοπό έχει:

- Την παροχή υπηρεσιών για τον διαμοιρασμό CTI.
- Την αυτοματοποίηση της διαδικασίας.
- Την παροχή λειτουργικότητας για τη συνεργατική ανάλυση απειλών.

(Guarascio, Cassavia, Pisani, & Manco, 2022)

Μια τέτοια πλατφόρμα βασίζεται σε πρότυπα, όπως το OpenIoC, το STIX ή το TAXII, για την απλοποίηση του διαμοιρασμού των CTI. (Preuveneers & Joosen, 2021) Η αλληλεπίδρασή τους με άλλα εργαλεία κυβερνοάμυνας επιτρέπει τη δημιουργία πολυεπίπεδων και αυτόματων μηχανισμών, που διαρκώς αναλύουν τα ετερογενή CTI που σχετίζονται με τις τακτικές και τις τεχνικές των επιτιθέμενων, τις ενδείξεις από περιστατικά κ.α. και σκοπό έχουν την παροχή μιας αποτελεσματικής άμυνας. (Ramsdale, Shiaeles, & Kolokotronis, 2020)

Από την άλλη, η χρήση μιας TIP δεν είναι πάντοτε απλή υπόθεση. Για παράδειγμα, όταν αξιοποιούνται συστήματα IDPS που βασίζονται στη χρήση τεχνικών Μηχανικής Μάθησης (Machine Learning) [160], υπάρχει μεγάλη δυσκολία στην εξαγωγή και τον διαμοιρασμό αυτής καθαυτής της γνώσης. (Guarascio, Cassavia, Pisani, & Manco, 2022) Μια άλλη δυσκολία προκύπτει από το γεγονός ότι τα IoC μπορεί να αποκαλύπτουν ευαίσθητες πληροφορίες για τον οργανισμό και για τον λόγο αυτό μην υπάρχει προθυμία διαμοιρασμού τους. (Preuveneers & Joosen, 2021) Για αυτά τα ζητήματα η ερευνητική κοινότητα προτείνει νέες προσεγγίσεις, με απώτερο σκοπό οι πλατφόρμες TIP να παρέχουν τον τρόπο για την ενδυνάμωση της ασφάλειας.

[160] Κλάδος της Τεχνητής Νοημοσύνης που αξιοποιείται σε μεγάλο βαθμό για την ανίχνευση ανωμαλιών από τα IDPS. Βασική της αρχή είναι ότι τα συστήματα μπορούν να εκπαιδευτούν από τα δεδομένα, ώστε στη συνέχεια να εντοπίζουν μοτίβα μόνα τους και να λαμβάνουν και τις ανάλογες αποφάσεις χωρίς ανθρώπινη παρέμβαση.

7

Ενοποίηση του Wazuh με το MISP

7.1 Εισαγωγή

Σκοπός αυτού του κεφαλαίου είναι να αναδειχθεί πώς η ενοποίηση ενός Συστήματος Ανίχνευσης / Πρόληψης Εισβολής με μια πλατφόρμα TIP μπορεί, μέσω της αυτοματοποίησης, να προσφέρει πολύ γρήγορη απόκριση απέναντι σε κάποιες κυβερνοαπειλές. Η αυτοματοποιημένη και ταχεία απόκριση είναι επιθυμητή, γιατί μπορεί να εξουδετερώσει μια απειλή στις πρώτες φάσεις της και συνεπώς να αποτρέψει τη ζημία στο σύστημα. Η εναλλακτική θα ήταν να υπάρχει αναμονή έρευνας των καταγραφών από κάποιον ανθρώπινο αναλυτή ασφάλειας· αυτό όμως θα επέτρεπε στην απειλή να εξελιχθεί και, ενδεχομένως, στο χρονικό διάστημα που απαιτείται για την έρευνα, να καταφέρει να προκαλέσει κάποια μικρή ή μεγάλη ζημία στο σύστημα.

Ως Σύστημα Ανίχνευσης και Πρόληψης Εισβολής αξιοποιήθηκε το *Wazuh*, ενώ ως πλατφόρμα TIP το *MISP*. Πρόκειται για δύο πλατφόρμες ανοικτού κώδικα που επιτρέπουν στον χρήστη, αν το επιθυμεί, να παρέμβει στη δομή τους για να τα προσαρμόσει στις δικές του ανάγκες, ενώ παράλληλα μπορεί να λάβει υποστήριξη από μια μεγάλη κοινότητα χρηστών και σχεδιαστών λογισμικού.

7.2 Wazuh

Όπως περιγράφεται στον ιστότοπό του, το *Wazuh* είναι ελεύθερο λογισμικό και λογισμικό ανοικτού κώδικα που προέρχεται από το επίσης ανοικτού κώδικα έργο OSSEC. Το OSSEC αναπτύχθηκε ως ένα host-based IDS με δυνατότητες ανάλυσης αρχείων καταγραφής, παρακολούθησης της ακεραιότητας των αρχείων του συστήματος, ειδοποιήσεων σε πραγματικό χρόνο καθώς και λειτουργίες απόκρισης με ενεργητικά μέτρα (κυβερνοάμυνας). (Wazuh, Inc., 2023) Το 2015, το Wazuh δημιουργήθηκε ως παρακλάδι του OSSEC, επεκτείνοντας τον βασικό πυρήνα του με νέα χαρακτηριστικά και βελτιώσεις, καθώς και με μια φιλικότερη προς τον χρήστη διεπαφή. Πλέον έχει εξελιχθεί σε μια πλατφόρμα *Εκτεταμένης Ανίχνευσης και Απόκρισης* (XDR [161], [162]), καθώς ενοποιεί την πρόληψη, την ανίχνευση και την απόκριση σε ένα εργαλείο. (Wazuh, Inc., 2023)

Οι δυνατότητες που προσφέρει το Wazuh περιλαμβάνουν:

- Την *αξιολόγηση παραμετροποίησης*, που επιτυγχάνεται μέσα από περιοδικές σαρώσεις των συστημάτων που προστατεύονται.
- Την *ανίχνευση κακόβουλου λογισμικού και κυβερνοεπιθέσεων*, μέσα από διάφορους μηχανισμούς που εντοπίζουν κακόβουλες ενέργειες και ανωμαλίες.
- Την *παρακολούθηση της ακεραιότητας των αρχείων*, μέσω ενός μηχανισμού που καταγράφει τις αλλαγές στο περιεχόμενο, τα δικαιώματα, τα χαρακτηριστικά των αρχείων, αλλά και τους χρήστες ή τις εφαρμογές που επιτελούν αυτές τις αλλαγές.
- Το λεγόμενο «*κυνήγι απειλών*» (threat hunting), που γίνεται δυνατή μέσα από την εκτενή καταγραφή γεγονότων, τη δυνατότητα εκτέλεσης ερωτημάτων στα καταγεγραμμένα γεγονότα, αλλά και τη χαρτογράφησή τους με τακτικές και τεχνικές του μοντέλου ATT&CK.
- Την *ανάλυση δεδομένων από αρχεία καταγραφής*, που προέρχονται είτε από το λειτουργικό σύστημα, είτε από τις εφαρμογές και επιτρέπουν στο Wazuh να ανιχνεύσει σφάλματα εκτέλεσης των εφαρμογών ή του συστήματος, παραβιάσεις πολιτικών κ.α.
- Την αυτοματοποιημένη *ανίχνευση ευπαθειών*, μέσω της σύγκρισης στοιχείων που λαμβάνονται από τα προστατευόμενα συστήματα με ενημερωμένες λίστες CVE.
- Την *απόκριση σε περιστατικά*, καθώς το Wazuh περιλαμβάνει έτοιμες λύσεις για την απόκριση σε απειλές με μέτρα ενεργητικής κυβερνοάμυνας, όπως πχ το μπλοκάρισμα της πρόσβασης από μια πηγή που αναγνωρίζεται ως απειλή.
- Τη *συμμόρφωση με κανονισμούς*, που επιτυγχάνεται από τους μηχανισμούς σάρωσης και ελέγχου του Wazuh και αφορά κανονισμούς όπως το GDPR, το NIST κα.
- Τον *έλεγχο της «υγιεινής»* του συστήματος, που επιτυγχάνεται με την καταγραφή των εγκατεστημένων εφαρμογών, των ενεργών διεργασιών, των ανοικτών port, πληροφοριών για το υλικό και το λειτουργικό σύστημα κ.α.

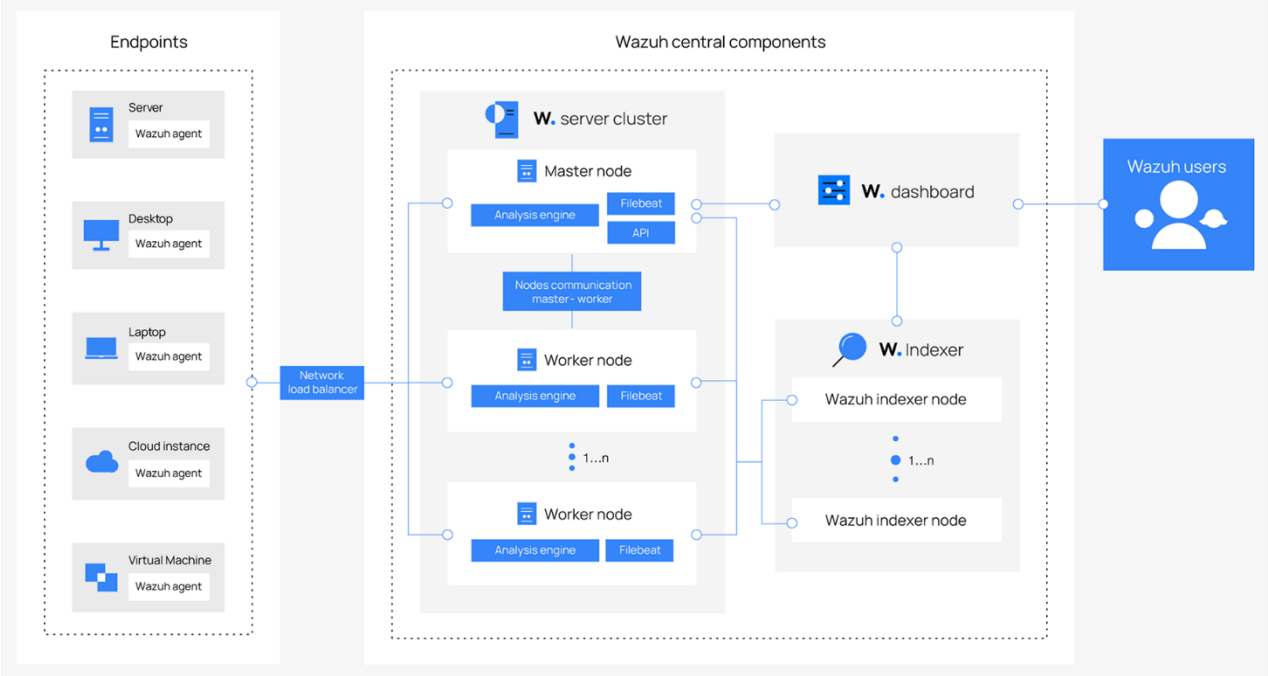
[161] Αρκτικόλεξο του όρου *eXtended Detection and Response*.

[162] Το Wazuh έχει τη δυνατότητα να λειτουργήσει ως SIEM, αλλά αυτή λειτουργικότητα δεν θα αποτελεί αντικείμενο της παρούσας εργασίας.

- Την *ασφάλεια των container* σε περιβάλλον Docker [163], καθώς το Wazuh υποστηρίζει εγγενώς μηχανισμούς για την επικοινωνία με τη μηχανή του Docker και τον έλεγχο μονάδων αποθήκευσης, δικτυακών συνδέσεων κ.α.
- Την *ενσωμάτωση με πλατφόρμες Υπολογιστικού Νέφους*, που επιτρέπουν τη λήψη δεδομένων από αυτές και την παραγωγή των ανάλογων ειδοποιήσεων.
- Την *προστασία του φόρτου εργασίας*, καθώς το Wazuh μπορεί να προσφέρει σε μια πλατφόρμα την παρακολούθηση τόσο τοπικών διακομιστών, όσο και διακομιστών του Υπολογιστικού Νέφους.

(Wazuh, Inc., 2024)

Ως προς τον τρόπο λειτουργίας του, το Wazuh προσφέρει δύο δυνατότητες. Στην πιο απλή κατάσταση λειτουργίας λειτουργεί ως ένα HIDS / HIPS και εποπτεύει μόνο τον υπολογιστικό κόμβο στον οποίο είναι εγκατεστημένο. Στην πιο προηγμένη μορφή του, το Wazuh μετατρέπεται σε ένα κατανεμημένο σύστημα IDS / IPS με κάποια κεντρικά συστατικά μέρη να λειτουργούν ως ένας ενιαίος management server (εφεξής *Wazuh manager*), ενώ στους υπόλοιπους κόμβους που εποπτεύονται (endpoints) εγκαθίσταται και λειτουργεί το λογισμικό του *Wazuh agent*, που σκοπό έχει να παρακολουθεί και να αποστέλλει στο Wazuh manager όλα τα παρατηρούμενα γεγονότα. Στο Wazuh manager γίνεται η ανάλυσή των δεδομένων και η ανίχνευση των απειλών. Το Wazuh manager είναι επίσης υπεύθυνο για να κρίνει πότε θα ενεργοποιούνται οι δυνατότητες απόκρισης στα endpoints, με σκοπό την εξουδετέρωση των απειλών που έχουν ανιχνευτεί.



Εικόνα 10: Βασικά συστατικά μέρη του Wazuh
[πηγή: (Wazuh, Inc., 2024)]

[163] Πρόκειται για μια τεχνολογία που προσφέρει παρόμοιες δυνατότητες με τις εικονικές μηχανές (VM), αλλά θεωρείται λιγότερο απαιτητική σε υπολογιστικούς πόρους. Αξιοποιείται σε μεγάλο βαθμό για την παροχή υπηρεσιών στο Υπολογιστικό Νέφος.

Όσον αφορά τη δομή του Wazuh manager, αυτός αποτελείται από τρία βασικά συστατικά μέρη [164]:

- Ένα είναι το *Wazuh server* [165], που έχει την ευθύνη της ανάλυσης των δεδομένων που λαμβάνονται από τους διάφορους agent και την ανίχνευση των απειλών. Επίσης, έχει τη δυνατότητα της απομακρυσμένης ρύθμισης των agent.
- Άλλο είναι το *Wazuh indexer*, που είναι μια μηχανή αναζήτησης και ανάλυσης. Έχει την ευθύνη της διαχείρισης των ειδοποιήσεων που παράγονται από το Wazuh server.
- Το τελευταίο είναι το *Wazuh dashboard*, που είναι μια διεπαφή χρήστη με τη μορφή ιστοσελίδας για την παρακολούθηση της λειτουργίας ολόκληρου του συστήματος από τους χειριστές – αναλυτές ασφαλείας.

Αξιοσημείωτο χαρακτηριστικό του Wazuh είναι πως προσφέρει τη δυνατότητα (μέσω της κατάλληλης παραμετροποίησης) για τη λήψη δεδομένων προς ανάλυση από συσκευές στις οποίες δεν είναι δυνατόν να εγκατασταθεί το λογισμικό agent (πχ. κάποιο firewall, switch, router ή ακόμα και άλλο IDS)· απαραίτητη προϋπόθεση είναι να υποστηρίζουν το πρωτόκολλο SSH ή το syslog ή με κάποιον άλλο τρόπο να αξιοποιείται το API του Wazuh για την «τροφοδοσία» των δεδομένων.

Ένα σημείο που πρέπει να αναφερθεί είναι πως το Wazuh δεν παρέχει εγγενώς τη δυνατότητα ανάλυσης της κίνησης όλου του δικτύου. Και είναι λογικό, αφού βασιζόμενο στη χρήση agent, που αποστέλλουν προς ανάλυση μόνο όσα δεδομένα καταγράφονται στα log των αντίστοιχων υπολογιστικών κόμβων, δεν μπορεί να έχει πλήρη εποπτεία των πακέτων στο δίκτυο. Ωστόσο, η αδυναμία αυτή είναι δυνατόν να αναπληρωθεί με τη χρήση ενός NIDS, το οποίο θα αναλάβει να τροφοδοτεί το Wazuh με τα απαραίτητα δεδομένα δικτύου [166]. Με μια τέτοια ενσωμάτωση το Wazuh ουσιαστικά μετατρέπεται σε ένα hybrid IDS / hybrid IPS.

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, πολλοί κατασκευαστές δημιουργούν ένα προϊόν, το οποίο με βάση την παραμετροποίησή του μπορεί να λειτουργεί είτε ως IDS, είτε IPS. Αυτό ισχύει και για το Wazuh, το οποίο από προεπιλογή λειτουργεί ως IDS. Ωστόσο, ο διαχειριστής έχει τη δυνατότητα να ενεργοποιήσει μια σειρά από μέτρα απόκρισης ώστε να το μετατρέψει σε IPS. Αυτό που πρέπει να παρατηρηθεί είναι, πως λόγω της αρχιτεκτονικής και του τρόπου λειτουργίας του, το Wazuh δεν μπορεί να λειτουργήσει σε κατάσταση inline. Συνεπώς, τα μέτρα απόκρισης που προσφέρει ως IPS βασίζονται κυρίως στην επαναρρύθμιση των endpoint που έχει υπό τον έλεγχό του.

[164] Αξίζει να σημειωθεί πως τα τρία αυτά συστατικά μέρη μπορούν να εγκατασταθούν ως ένα ενιαίο σύστημα σε έναν υπολογιστικό κόμβο ή να εγκατασταθούν σε διαφορετικούς κόμβους για λόγους επεκτασιμότητας του συστήματος.

[165] Αναγράφεται ως *server cluster* στην εικόνα 10.

[166] Το Wazuh μπορεί να λαμβάνει τα δεδομένα χάρη σε ένα agent που είναι εγκατεστημένο στον κόμβο – sensor και εξάγει δεδομένα από τα κατάλληλα αρχεία καταγραφής για να τα προωθήσει στον Wazuh manager. Αν δεν υπάρχει η δυνατότητα εγκατάστασης agent, τότε μπορεί να αξιοποιηθούν άλλα πρωτόκολλα ή το API όπως περιγράφεται στην προηγούμενη παράγραφο.

7.3 MISP

Το *MISP* είναι λογισμικό ανοικτού κώδικα που ασχολείται με τη συλλογή, την αποθήκευση, τη διανομή και τον διαμοιρασμό πληροφοριών σχετικά με απειλές ασφαλείας, ενώ γενικότερα στοχεύει στη διευκόλυνση της ανάλυσής τους. (MISP, χ.χ.)

Ξεκίνησε ως προσωπικό έργο του Christophe Vandeplas το 2011, που διαπίστωσε πως τα IoC που διαμοιράζονταν μέσω email ή αρχείων pdf, δεν ήταν κατανοητά από αυτοματοποιημένους μηχανισμούς. Ονόμασε το έργο του *CyDefSIG* [167] και σύντομα κατάφερε να το κάνει αποδεκτό αρχικά από τις Ένοπλες Δυνάμεις του Βελγίου και λίγο αργότερα από το NATO, που αναγνώρισε την ανοικτότητά του ως πλεονέκτημα. Με την υιοθέτησή του από το NATO, ξεκίνησε η συνεργατική ανάπτυξή του και η προσθήκη επιπλέον λειτουργικότητας, ενώ αποφασίστηκε συλλογικά από την ομάδα ανάπτυξής του να διατίθεται ως ελεύθερο λογισμικό. Υπό την αιγίδα του NATO το έργο μετονομάστηκε σε MISP: Malware Intelligence Sharing Platform [168]. Όπως μαρτυρούσε το όνομά του, αρχικά ασχολούταν μόνο με IoC που αφορούσαν κακόβουλο λογισμικό. Καθώς όμως αναπτυσσόταν περεταίρω, άρχισε να περιλαμβάνει επιπλέον πληροφορίες για κάθε είδους απειλές, ακόμα και για απάτες ή ευπάθειες ασφαλείας. Η εξέλιξη αυτή επέφερε και μια νέα αλλαγή του ονόματός της πλατφόρμας σε *MISP Threat Sharing* (ή απλά MISP). (MISP, χ.χ.)

Ως πλατφόρμα TIP, το MISP προσφέρει λειτουργίες περιστρέφονται γύρω από την αποθήκευση, τον διαμοιρασμό, τη συσχέτιση και φυσικά την ανάλυση απειλών. Οι πληροφορίες που διαχειρίζεται δεν περιορίζονται στη μορφή των IoC που αναλύθηκε πρωτύτερα στην παρούσα εργασία, αλλά έχουν μια γενικότερη μορφή που καλύπτει ένα μεγάλο εύρος απειλών· από τις στοχευμένες επιθέσεις και τις οικονομικές απάτες, ως και την αντιτρομοκρατία. Η δομή των πληροφοριών είναι απλή, ώστε να αποθηκεύονται και να είναι εύκολα κατανοητές από ανθρώπους. Επιπλέον, επεκτείνονται με τη προσθήκη μεταδεδομένων, ώστε να είναι εύκολα διαχειρίσιμες και από αυτοματοποιημένες διεργασίες. (MISP, χ.χ.)

Ο διαμοιρασμός των πληροφοριών εξασφαλίζει ότι η ανάλυσή τους μπορεί να γίνει συνεργατικά, αλλά και ότι δεν θα κάνει κάποιος δουλειά που έχει ήδη γίνει αλλού· το κυριότερο όμως, συμβάλει στην έγκαιρη και αποτελεσματική ανίχνευση των απειλών. Σε αυτά τα πλαίσια, το MISP παρέχει λειτουργίες αυτοματοποίησης, όπως ο συγχρονισμός με άλλους server του MISP ή η απευθείας εξαγωγή κανόνων για χρήση σε IDS [169]. Ακόμα υποστηρίζει την εισαγωγή ή την εξαγωγή δεδομένων σε μορφή σύμφωνη με το πρότυπο OpenIoC ή σε μορφότυπο CSV ή ακόμα και σε μορφή ελεύθερου κειμένου [170]. Με όλους αυτούς τους τρόπους ή μέσω του δικού του RESTful API, το MISP υποστηρίζει την ενοποίηση με άλλα εργαλεία ασφάλειας. (MISP, χ.χ.)

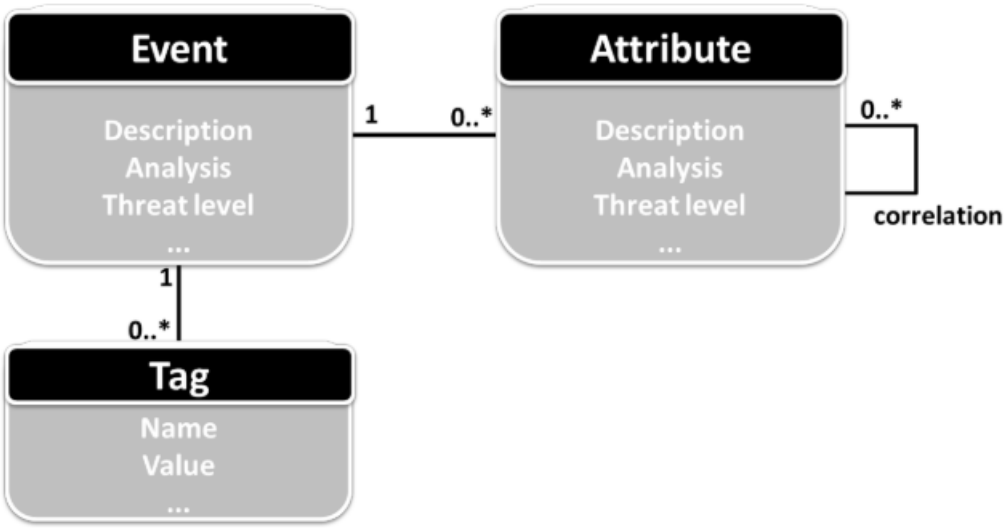
[167] CyDefSIG: Cyber Defence Signatures, δηλαδή Υπογραφές Κυβερνοάμυνας.

[168] Το όνομα εμπνεύστηκε ο Alex Vandurme του NATO. (MISP, χ.χ.)

[169] Υποστηρίζεται εγγενώς η εξαγωγή κανόνων για τα διαδεδομένα συστήματα IDS ανοικτού κώδικα: Snort, Suricata και Zeek (πρώην Bro).

[170] Επιπλέον, υποστηρίζεται η εξαγωγή δεδομένων σύμφωνα με το πρότυπο STIX.

Ένα σημείο που πρέπει να αναλυθεί περεταίρω είναι το μοντέλο των δεδομένων που αξιοποιεί το MISP για την αποθήκευση, αλλά και γενικά για τη διαχείριση των πληροφοριών σχετικά με τις απειλές. Κεντρικό ρόλο για τη διατήρηση αυτών των πληροφοριών παίζει μια οντότητα που ονομάζεται **event** (*γεγονός*). Κάθε τέτοια οντότητα αποτελείται από ένα σύνολο από **attributes** (*χαρακτηριστικά*), που αποθηκεύουν σημαντικά δεδομένα σχετικά με την απειλή. Για κάθε attribute είναι απαραίτητο να προσδιοριστούν ένα **category** (*κατηγορία*), που το εντάσσει σε ένα γενικό πλαίσιο – λ.χ. δραστηριότητα δικτύου, τύπος απάτης κ.α. – και ένα **type** (*τύπος*), που ορίζει το συγκεκριμένο είδος – λ.χ. md5 ή sha256 checksum, URL, όνομα αρχείου, domain name, διεύθυνση IP πηγής ή διεύθυνση IP προορισμού κ.α. – που θα έχει η τιμή του attribute αυτού. [171] Επιπλέον, κάθε event μπορεί να επεκταθεί με τη χρήση **tags** (*ετικετών*) που παρέχουν πρόσθετα μεταδεδομένα για αυτή την οντότητα και μπορούν να αξιοποιηθούν ιδιαίτερα σε διεργασίες αυτοματοποίησης. [172] (Wagner, Dulaunoy, Wagener, & Iklody, 2016)



Εικόνα 11: Διαγραμματική αναπαράσταση ενός event του MISP [πηγή: (Wagner, Dulaunoy, Wagener, & Iklody, 2016)]

[171] Οι νεότερες εκδόσεις του MISP, πέρα από τον απλό τύπο attribute, υποστηρίζουν και μια νέα μορφή που ονομάζεται *object* (*αντικείμενο*). Ουσιαστικά πρόκειται για μια οργανωμένη ομάδα από attributes, που παρέχουν περισσότερες πληροφορίες όταν εξετάζονται ως σύνολο. Στην παρούσα εργασία δεν θα ασχοληθούμε με αυτό τον τύπο.

[172] Τα tags μπορούν να οργανωθούν σε *taxonomies* ανάλογα με τις κατηγορίες στις οποίες θέλουμε να εντάξουμε το event που εξετάζεται. Υπάρχουν έτοιμα taxonomies, αλλά δίνεται και η δυνατότητα στον χρήστη να δημιουργήσει τα δικά του.

Εκτός των taxonomies, οι νεότερες εκδόσεις του MISP διαθέτουν τη δυνατότητα οργάνωσης σε *galaxies*. Πρόκειται για έναν πιο προηγμένο τρόπο καθορισμού μεταδεδομένων, που αντικατοπτρίζουν κάποια συσχέτιση με μια μεγάλη οντότητα. Για παράδειγμα, μέσω galaxies θα μπορούσε να αποτυπωθεί η συσχέτιση ενός event με μια τακτική του μοντέλου ATT&CK.

Στην παρούσα εργασία, τα taxonomies και τα galaxies δεν θα αξιοποιηθούν και αναφέρονται μόνο για λόγους πληρότητας της παρουσίασης του MISP.

Ένας μηχανισμός που πρέπει να εξεταστεί είναι ο μηχανισμός **correlation** (**συσχέτισης**) των event. Κάθε φορά που προστίθεται ένα νέο attribute, το MISP αυτόματα αναζητά την ίδια τιμή σε άλλα event. Πρόκειται για την πιο απαιτητική σε υπολογιστικούς πόρους διεργασία στο MISP, όταν διατηρείται μεγάλο πλήθος από event. (MISP, 2022) Ωστόσο, είναι ένας μηχανισμός εξαιρετικά χρήσιμος, καθώς αποκαλύπτει σχέσεις μεταξύ event, που θα ήταν χρονοβόρο και κάποιες φορές εξαιρετικά κοπιαστικό να ανακαλύψει ένας ανθρώπινος χειριστής μόνος του.

Η επόμενη λειτουργία του MISP που πρέπει να αναφερθεί αφορά τον διαμοιρασμό των πληροφοριών. Οποιοδήποτε άτομο ή οργανισμός (εφεξής χρήστης) στον κόσμο μπορεί ελεύθερα και χωρίς χρέωση να εγκαταστήσει το MISP σε έναν δικό του υπολογιστικό κόμβο και να αρχίσει να καταχωρεί τα δικά του event. Ωστόσο, ο διαμοιρασμός των πληροφοριών είναι η ουσία σε μια TIP. Οι διάφοροι χρήστες του MISP μπορούν να ενταχθούν σε κοινότητες, καθεμία με δικούς της κανόνες ως προς τη συμμετοχή. Το MISP διαθέτει τους μηχανισμούς για τη διαχείριση των χρηστών και των δικαιωμάτων καθενός από αυτούς. Οι πληροφορίες που διαχειρίζεται ένας MISP server μπορούν να οριστούν για ιδιωτική χρήση ή για δημόσια. Το MISP διαθέτει μηχανισμούς ώστε ένας MISP server να μπορεί να «συγχρονιστεί» με έναν άλλο MISP server, αντλώντας event με βάση πάντα τα δικαιώματα που έχουν οριστεί. (MISP, 2022)

Με αυτή τη λογική και πάντα σεβόμενοι το γεγονός ότι υπάρχουν ευαίσθητες πληροφορίες μόνο για εσωτερική χρήση και άλλες που μπορούν να διατεθούν δημόσια, είναι δυνατόν να οργανωθεί μια αρκετά πολύπλοκη ιεραρχία στο MISP. Αυτή τη λογική ακολουθούν το NATO και οι οργανισμοί που συνεργάζονται με αυτό, για να διαθέσουν πληροφορίες δημόσια. Οι MISP server που βρίσκονται «ψηλά» στην ιεραρχία κρατούν όλες τις πληροφορίες και διαχέουν μόνο τις πληροφορίες που δεν είναι απόρρητες ή ευαίσθητες κτλ. στα χαμηλότερα επίπεδα.

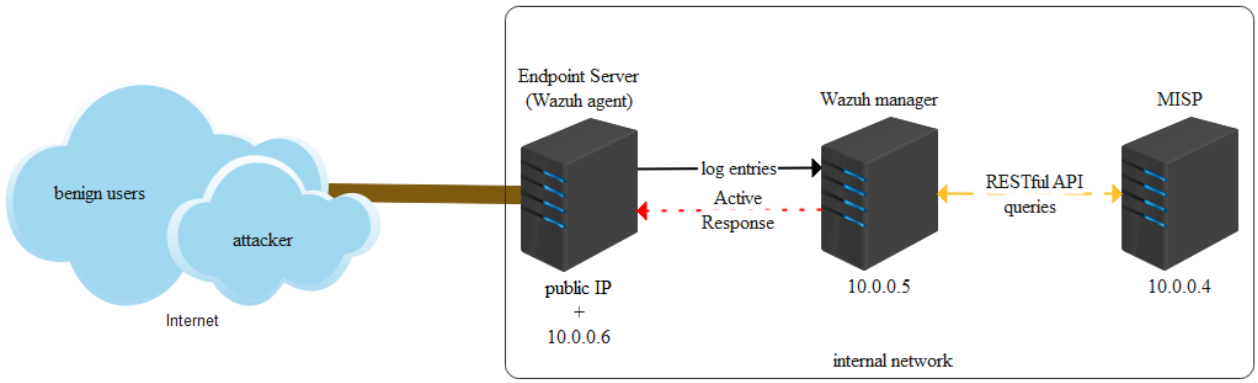
Μια παρόμοια, αλλά πιο απλή λογική διαμοιρασμού, καθώς δεν απαιτεί την ένταξη σε κάποια κοινότητα του MISP, βασίζεται στην αξιοποίηση των *feeds*. Ένα feed είναι ένας τοπικός ή απομακρυσμένος πόρος ο οποίος περιέχει πληροφορίες σχετικά με απειλές και από τον οποίο μπορούν να αντληθούν αυτές οι πληροφορίες ανά τακτά χρονικά διαστήματα. Υποστηρίζεται η εισαγωγή πληροφοριών με το μορφότυπο του MISP, με μορφότυπο CSV ή με μορφή ελεύθερου κειμένου. Η βασική εγκατάσταση του MISP περιέχει μια λίστα με feed που είναι διαθέσιμα προς όλους, αλλά υποστηρίζεται και η δυνατότητα κάθε χρήστης να ορίσει και δικά του. (MISP, 2022)

Τέλος, πρέπει να αναφέρουμε ότι το MISP περιλαμβάνει πλήθος άλλων λειτουργιών, όμως η αναφορά σε όλες τους είναι πέρα από τα όρια της παρούσας εργασίας.

7.4 *«Η ισχύς εν τη ενώσει»*

Αφού έγινε μια πρώτη παρουσίαση του IDS / IPS Wazuh και της TIP MISP, το επόμενο βήμα είναι να εξεταστεί πώς μέσα από την ενοποίηση της λειτουργίας τους μπορεί να βελτιωθεί το επίπεδο της προστασίας σε ένα σύστημα.

Η μελέτη πραγματοποιείται με αξιοποίηση της πλατφόρμας Azure, όπου δημιουργούνται τρεις εικονικές μηχανές (VM) που εκτελούν το λειτουργικό σύστημα Ubuntu Server 22.04: μία για το MISP, μία για το Wazuh manager [173] και μια ακόμη για ένα γενικής χρήσης endpoint server που εποπτεύεται από το Wazuh agent. Οι τρεις υπολογιστικοί κόμβοι βρίσκονται στο ίδιο εσωτερικό δίκτυο ώστε να επικοινωνούν ανεμπόδιστα μεταξύ τους, αλλά μόνο ο endpoint server διαθέτει απευθείας σύνδεση στο Διαδίκτυο μέσω μιας δημόσιας IP. Στο εσωτερικό δίκτυο δεν χρησιμοποιείται άλλο εργαλείο προστασίας (firewall ή antimalware), πέραν του Wazuh.



Εικόνα 12: Η δομή του υπό εξέταση συστήματος

Στην παραπάνω εικόνα εμφανίζεται η δομή του συστήματος που εξετάζεται. Στο Διαδίκτυο υπάρχουν αγαθοί και κακόβουλοι χρήστες που μπορούν να επικοινωνήσουν μόνο με το endpoint server. Στο endpoint server λειτουργεί το λογισμικό του Wazuh agent, το οποίο είναι υπεύθυνο να τροφοδοτεί με δεδομένα ελέγχου το κεντρικό μηχανισμό ανάλυσης στο Wazuh manager. Στο Wazuh manager έχουν ρυθμιστεί τα κατάλληλα σενάρια, ώστε αν κάποια δραστηριότητα κριθεί ύποπτη, να υποβάλει με αυτοματοποιημένο τρόπο ένα ερώτημα προς το MISP, με σκοπό να εξακριβωθεί αν η δραστηριότητα σχετίζεται με κάποιο καταγεγραμμένο IoC. Σε περίπτωση που η απάντηση είναι θετική, το Wazuh manager έχει τη δυνατότητα να ενεργοποιήσει τα κατάλληλα μέτρα εξουδετέρωσης της απειλής μέσω του Wazuh agent στο endpoint server.

Μια δεύτερη περίπτωση είναι το Wazuh manager να ανιχνεύσει μια επίθεση που στα πρώτα στάδιά της δεν είχε συσχετιστεί με κάποιο IoC του MISP. Σε αυτή την περίπτωση, αφενός το Wazuh manager ενεργοποιεί αυτόματα τα κατάλληλα αντίμετρα, αλλά παράλληλα κάνει μια καταχώρηση ενός νέου IoC στο MISP, ώστε μελλοντικά η ίδια απειλή να αναγνωριστεί ταχύτερα.

[173] Το Wazuh manager και το MISP εγκαθίστανται σε διαφορετικούς κόμβους για δύο λόγους: Πρώτον, για λόγους απόδοσης· καθεμία από τις πλατφόρμες εκτελεί υπολογιστικά απαιτητικές εργασίες, οπότε είναι προτιμότερο να διαθέτει αποκλειστικά δικούς της πόρους. Δεύτερον, για λόγους κλιμακωσιμότητας του συστήματος· σε περίπτωση που έχουμε έναν μεγάλο οργανισμό, με μεγάλο πλήθος υπολογιστικών κόμβων, πιθανόν να απαιτούνται περισσότερα IDS για την αποτελεσματικότερη εποπτεία όλων των κόμβων. Με αυτή τη δομή υπάρχει μια πλατφόρμα MISP που συγκεντρώνει όλες τις πληροφορίες για τις απειλές και από την οποία αντλούν πληροφόρηση όλα τα IDS.

7.4.1 Σενάριο 1ο: Ανίχνευση με βάση μια υπογραφή

Σε αυτό το σενάριο θα εξετάσουμε πως το Wazuh σε συνεργασία με το MISP μπορούν να προσφέρουν προστασία ανάλογη ενός antimalware, εφαρμόζοντας ανίχνευση κακόβουλων αρχείων με βάση το checksum hash τους. Το Wazuh έχει τη δυνατότητα να καταγράφει κάθε αλλαγή (δημιουργία, τροποποίηση ή διαγραφή) στο σύστημα αρχείων του endpoint. Εστιάζοντας στη δημιουργία ενός νέου αρχείου, μπορούμε να ορίσουμε ότι το Wazuh θα απευθύνει ένα ερώτημα στο MISP, για να εξετάσει αν το checksum hash του νέου αρχείου σχετίζεται με κάποιο γνωστό κακόβουλο λογισμικό. Αν υπάρχει συσχέτιση, τότε το Wazuh θα προκαλέσει τη διαγραφή του νέου αρχείου.

Για αυτό το σενάριο μελέτης θα πρέπει να γίνει η εξής παραμετροποίηση στο σύστημα που εξετάζεται:

- Η ενεργοποίηση της παρακολούθησης ενός φακέλου για αλλαγές [174].
- Η ρύθμιση της επικοινωνίας του Wazuh με το MISP. Πρόκειται για το πιο απαιτητικό στάδιο και απαιτεί τη δημιουργία ενός σεναρίου εντολών (το *custom-misp*) [175]. Συνολικά, για τη ρύθμιση της επικοινωνίας τους για αυτό το σενάριο μελέτης απαιτούνται τα εξής βήματα:
    - Πρώτον, πρέπει να αναγνωριστεί ποιο alert παράγεται στο Wazuh manager όταν δημιουργηθεί ένα νέο αρχείο στο endpoint.
    - Δεύτερον, πρέπει να οριστεί ένας κανόνας στο Wazuh ώστε κάθε φορά που εμφανίζεται αυτό το alert, να εκτελείται το σενάριο εντολών custom-misp.
    - Τρίτον, ο κώδικας στο custom-misp πρέπει να αντλεί το md5 hash του νέου αρχείου από το alert του Wazuh και μέσω της κατάλληλης κλήσης του API του MISP, να του ζητά να εξετάσει αν αυτό το hash υπάρχει σε κάποιο IoC. [176] Αν το MISP επιστρέψει αποτέλεσμα – δηλαδή υπάρχει ταύτιση του hash με κάποιο IoC, τότε το custom-misp αναλαμβάνει να παραγάγει ένα νέο alert στο Wazuh.
- Η δημιουργία ενός νέου κανόνα στο Wazuh, που επιβάλει πως η εμφάνιση του alert για την ταύτιση με IoC στο MISP θα προκαλεί την ενεργοποίηση του μέτρου εξουδετέρωσης της απειλής. Το Wazuh manager δίνει εντολή στο Wazuh agent να εκτελέσει το σενάριο εντολών *remove-threat.sh* [177], το οποίο διαγράφει το κακόβουλο αρχείο από το endpoint server.
- Η δημιουργία ενός πρόσθετου alert που θα ενεργοποιείται μετά την επιτυχή διαγραφή του κακόβουλου αρχείου και θα συνοψίζει τις πληροφορίες σχετικά με το αρχείο αυτό και την ταύτισή του με ένα IoC στο MISP.

---

[174] Με βάση τις οδηγίες από το https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/basic-settings.html#real-time-monitoring.

[175] Το αρχείο αυτό είναι ο συνδετικός κρίκος μεταξύ του Wazuh και του MISP. Το σημαντικότερο κομμάτι του κώδικά του παρουσιάζεται στο Παράρτημα Ι.

[176] Για να έχει αποτέλεσμα αυτή η αναζήτηση, θα πρέπει να έχουμε εξασφαλίσει πως το MISP έχει πρόσβαση σε IoC που περιέχουν md5 hash γνωστών κακόβουλων αρχείων. Για το συγκεκριμένο σενάριο μελέτης ενεργοποιήθηκε το feed #73 – List of malicious hashes – Banco do Brasil S.A.

[177] Με βάση τις οδηγίες από το https://documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html.

Όπως εξηγήθηκε στο σχετικό κεφάλαιο, μια κυβερνοεπίθεση με κακόβουλο λογισμικό βασίζεται είτε στην εξαπάτηση ενός χρήστη με μεθόδους της κοινωνικής μηχανικής, είτε στην αξιοποίηση κάποιας ευπάθειας. Το αποτέλεσμα (στις περισσότερες περιπτώσεις [178]) είναι η δημιουργία ενός νέου αρχείου, που θα πρέπει να εκτελεστεί για επιτελέσει το κακόβουλο έργο του.

Για να εξετάσουμε κατά πόσον η ενοποίηση του Wazuh με το MISP μπορεί να επιφέρει το επιθυμητό αποτέλεσμα, θα κάνουμε μια προσομοίωση εισβολής ενός κακόβουλου αρχείου στο endpoint server με την εκτέλεση της κάτωθι εντολής:

```
curl -o malicious.txt https://secure.eicar.org/eicar.com
```

Η εντολή αυτή προκαλεί τη μεταφόρτωση του αρχείου *EICAR* [179] με το όνομα malicious.txt. Μετά τη μεταφόρτωση του αρχείου στο endpoint, στο Wazuh βλέπουμε τα alert όπως απεικονίζονται κάτωθι:

	Time	agent.name	rule.description	rule.level	rule.id
›	May 13, 2024 @ 21:53:1	agent-srv	Active response removed threat /home/[redacted]/malicious.txt confirmed in MISP IOC 29.	7	100092
›	May 13, 2024 @ 21:53:11.593	agent-srv	File deleted.	7	553
›	May 13, 2024 @ 21:53:11.467	agent-srv	MISP - IoC found in Threat Intel - Category: Payload delivery, Attribute: 44d88612fea8a8f36de82e1278abb02f	12	100622
›	May 13, 2024 @ 21:53:09.506	agent-srv	File added to the system.	5	554

Εικόνα 13: Ανίχνευση και εξουδετέρωση ενός κακόβουλου αρχείου

Ξεκινώντας από το κάτω μέρος της εικόνας, βλέπουμε το alert που εμφανίζεται πρώτο χρονικά και το οποίο προκύπτει μετά τη μεταφόρτωση του αρχείου EICAR στον εποπτευόμενο φάκελο. Στη συνέχεια εμφανίζεται το alert που παράγεται επειδή το MISP συσχέτισε το md5 hash του αρχείου EICAR με κάποιο IoC. Αυτό το alert πυροδοτεί τα αντίμετρα που οδηγούν στη διαγραφή του αρχείου EICAR από το endpoint (επισημαίνεται από το σχετικό alert του Wazuh). Τέλος, παράγεται ένα ακόμα alert που συγκεντρώνει όλες τις απαραίτητες πληροφορίες, ώστε κάποιος χειριστής – αναλυτής ασφάλειας να μπορέσει να διερευνήσει περαιτέρω το συμβάν στο Wazuh και στο MISP.

[178] Θα πρέπει να τονιστεί ότι η συγκεκριμένη μέθοδος ανίχνευσης σε καμία περίπτωση δεν μπορεί προσφέρει απόλυτη ασφάλεια απέναντι στο κακόβουλο λογισμικό, αλλά παρουσιάζεται καθαρά για λόγους επίδειξης των δυνατοτήτων που προκύπτουν από τη σύμπραξη ενός IDS / IPS με μια πλατφόρμα TIP.

Για παράδειγμα, όταν το κακόβουλο λογισμικό εκτελείται από μια φορητή συσκευής αποθήκευσης (λ.χ. USB stick), τότε δεν είναι απαραίτητο να δημιουργηθεί ένα νέο αρχείο στο σύστημα, οπότε η μέθοδος αυτή δεν έχει αποτέλεσμα. Επίσης, θυμίζουμε ότι ένας ιός δεν χρειάζεται να δημιουργήσει κάποιο νέο αρχείο, αλλά να τροποποιήσει κάποιο υπάρχον, οπότε και πάλι η συγκεκριμένη μέθοδος δεν θα μπορούσε να εμποδίσει την εξάπλωσή του στο σύστημα.

[179] Το αρχείο αναπτύχθηκε από το ινστιτούτο *European Institute for Computer Antivirus Research* (*EICAR*) από το οποίο έλαβε και το όνομά του. Είναι εσκεμμένα φτιαγμένο έτσι, ώστε να αναγνωρίζεται ως κακόβουλο από συστήματα antimalware. Σκοπός του είναι να δοκιμάζεται η ανταπόκριση των συστημάτων antimalware, χωρίς να υπάρχει η πιθανότητα να προκληθεί ζημία – μια υπαρκτή πιθανότητα αν γινόταν δοκιμή ενός πραγματικού κακόβουλου λογισμικού.

7.4.2 Σενάριο 2ο: Ανίχνευση με βάση μια ανωμαλία

Σε αυτό το σενάριο θα εξετάσουμε πως το Wazuh μπορεί να συνεργαστεί με το MISP για να καταχωρήσει με αυτόματο τρόπο ένα νέο IoC, που μελλοντικά θα βοηθήσει στην ταχύτερη εξουδετέρωση μιας απειλής από την ίδια πηγή.

Το Wazuh καταγράφει σε alert κάθε αποτυχημένη απόπειρα σύνδεσης στο endpoint server. Επίσης, διαθέτει ενεργούς κανόνες, με τους οποίους είναι σε θέση να αναγνωρίσει πότε οι διαδοχικές αποτυχίες σύνδεσης προκαλούνται από μια επίθεση brute force. Σε τέτοιες περιπτώσεις, το Wazuh μπορεί να επαναρρυθμίσει το endpoint server, αλλάζοντας τους κανόνες φιλτραρίσματος πακέτων στο firewall του, ώστε να γίνεται απόρριψη κάθε πακέτου από τη συγκεκριμένη IP.

Αυτή είναι μια δυνατότητα που προσφέρει το Wazuh χωρίς να έχει ανάγκη το MISP. Το πλεονέκτημα που μπορούμε να αποκομίσουμε από τη συνεργασία των δύο είναι το εξής: μετά την ανίχνευση της επίθεσης brute force και την ενεργοποίηση των μέτρων απόκρισης για την εξουδετέρωσή της από το Wazuh, μπορεί να δημιουργηθεί αυτόματα ένα νέο IoC στο MISP, ώστε την επόμενη φορά που η ίδια IP θα εμπλακεί σε κάποιο ύποπτο περιστατικό, να αναγνωριστεί άμεσα ως κακόβουλη [180].

Για αυτό το σενάριο μελέτης θα πρέπει να γίνει η εξής παραμετροποίηση στο σύστημα που εξετάζεται:

- Δεδομένου ότι οι κανόνες για την αναγνώριση της επίθεσης brute force είναι ήδη ενεργοί στο Wazuh, αυτό που πρέπει να γίνει είναι η ενεργοποίηση του αντίμετρου που προκαλεί το μπλοκάρισμα της IP του επιτιθέμενου στο endpoint server [181].
- Η δημιουργία ενός νέου κανόνα, ώστε κάθε φορά που εμφανίζεται το alert ανίχνευσης μιας επίθεσης brute force, να ενεργοποιείται και το αντίμετρο για το μπλοκάρισμα της IP του επιτιθέμενου [182].
- Ο ορισμός ενός νέου alert για το επιτυχημένο μπλοκάρισμα της IP.
- Η ρύθμιση της επικοινωνίας του Wazuh με το MISP, που γίνεται μέσω της επέκτασης του σεναρίου εντολών custom-misp. Για τον σκοπό αυτό, τα βήματα που πρέπει να γίνουν είναι:
    - Πρώτον, να οριστεί ένας κανόνας ώστε κάθε φορά που εμφανίζεται το alert για το επιτυχημένο μπλοκάρισμα της IP, να καλείται το σενάριο εντολών custom-misp.
    - Δεύτερον, το custom-misp να επεκταθεί με κώδικα που να αντλεί την IP του επιτιθέμενου από το alert και μέσω της κατάλληλης κλήσης του API του MISP, να ζητά την καταχώρηση ενός νέου IoC για αυτή την IP σε ένα προεπιλεγμένο event [183]. Αν το MISP απαντήσει ότι καταχώρηση ήταν επιτυχής, τότε το custom-misp αναλαμβάνει να παραγάγει και ένα νέο alert στο Wazuh.

[180] Το όφελος αυτής της προσέγγισης συνδέεται άμεσα και με το 3ο σενάριο που παρουσιάζεται παρακάτω.

[181] Ακολουθώντας τη συνήθη πρακτική, το Wazuh λειτουργεί ως IDS από προεπιλογή. Για τη μετατροπή του σε IPS θα πρέπει να γίνει ρητή ενεργοποίηση των επιθυμητών αντίμετρων.

[182] Για τον κανόνα αυτόν δείτε το Παράρτημα ΙΙ.

[183] Όλες οι IP που ταυτίζονται με επιθέσεις brute force καταχωρούνται ως attribute σε ένα προκαθορισμένο event, που ονομάζεται *LocallyMonitoredMaliciousIPsEvent* (σε περίπτωση που το event δεν υπάρχει ήδη από μια προηγούμενη επίθεση, τότε δημιουργείται από το σενάριο εντολών custom-misp).

Σε αντίθεση με το 1ο σενάριο, που έγινε προσομοίωση της επίθεσης, η συνεργασία του Wazuh με το MISP για το 2ο σενάριο δοκιμάστηκε απέναντι σε πραγματικές απειλές. Και δεν ήταν καθόλου δύσκολο, αφού το μόνο που χρειαζόταν ήταν να δοθεί μια δημόσια IP στο endpoint server για να γίνει διαθέσιμο στο Διαδίκτυο. Μόλις έγινε αυτό, άρχισαν να γίνονται πραγματικές επιθέσεις brute force στο πρωτόκολλο SSH [184] (!).

Παρακάτω απεικονίζεται μια τέτοια επίθεση όπως καταγράφηκε μέσα από τα alert του Wazuh:

	Time	rule.description	rule.level	rule.id
›	May 13, 2024 @ 22:26:59.921	MISP - new IoC created for 221.156.105.215	7	100624
›	May 13, 2024 @ 22:26:59.812	Active response blocked IP 221.156.105.215.	7	100095
›	May 13, 2024 @ 22:26:57.811	sshd: brute force trying to get access to the system. Non existent user.	10	5712
›	May 13, 2024 @ 22:26:57.810	Maximum authentication attempts exceeded.	8	5758
›	May 13, 2024 @ 22:26:55.807	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:51.803	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:47.801	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:43.796	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:39.791	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:35.832	sshd: Attempt to login using a non-existent user	5	5710
›	May 13, 2024 @ 22:26:35.789	sshd: Attempt to login using a non-existent user	5	5710

Εικόνα 14: Ανίχνευση και εξουδετέρωση μιας επίθεσης με παράλληλη δημιουργία νέου IoC

Ξεκινώντας και πάλι από το κάτω μέρος της εικόνας, βλέπουμε 7 διαδοχικά alert που σχετίζονται με αποτυχημένες προσπάθειες σύνδεσης στο SSH. Μετά εμφανίζονται δύο alert που σχετίζονται με την υπέρβαση των μέγιστων προσπαθειών αυθεντικοποίησης και την αναγνώριση μιας επίθεσης brute force αντιστοίχως. [185] Στη συνέχεια εμφανίζεται το alert σχετικά με την ενεργοποίηση των αντίμετρων που προκαλούν το μπλοκάρισμα της IP στο συγκεκριμένο endpoint. Τέλος, εμφανίζεται το alert σχετικά με την καταχώρηση ενός νέου IoC στο προκαθορισμένο event του MISP, με σκοπό κάποιος χειριστής – αναλυτής ασφάλειας να μπορέσει να διερευνήσει περαιτέρω το συμβάν.

[184] Όπως αναφέραμε στο κεφάλαιο 4, το Διαδίκτυο σαρώνεται διαρκώς από bots που προσπαθούν να βρουν το επόμενο «θύμα» τους. Λογικό λοιπόν είναι πως όλοι οι δημόσια προσβάσιμοι κόμβοι στο Διαδίκτυο αποτελούν στόχο μιας επίθεσης ανά πάσα στιγμή.

[185] Το Wazuh από προεπιλογή αναγνωρίζει μία επίθεση brute force μετά από 8 αποτυχίες σύνδεσης. Φυσικά αυτή η τιμή μπορεί να αναπροσαρμοστεί σε όποια τιμή κρίνει κατάλληλη ο εκάστοτε διαχειριστής, ότι είναι η καλύτερη για την προστασία του δικού του συστήματος.

7.4.3 Σενάριο 3ο: Ανίχνευση με βάση μια καταχώρηση σε μια μαύρη λίστα

Σε αυτό το σενάριο θα εξετάσουμε πως το Wazuh σε συνεργασία με το MISP μπορούν να εξαλείψουν, στα πρώτα βήματά τους, επιθέσεις brute force στο πρωτόκολλο SSH από IP που έχουν στο παρελθόν έχουν συσχετιστεί με κάποια κακόβουλη δραστηριότητα.

Όπως είπαμε και στην προηγούμενη ενότητα, το Wazuh καταγράφει σε alert κάθε αποτυχημένη απόπειρα σύνδεσης στο endpoint server. Αυτή τη φορά όμως θα αντιμετωπίσει την πρώτη αποτυχία ως ύποπτο περιστατικό και θα απευθύνει ένα ερώτημα στο MISP, για να εξετάσει αν η IP από την οποία προέρχεται το αίτημα σύνδεσης περιλαμβάνεται σε κάποιο IoC. Αν υπάρχει ταύτιση, τότε το Wazuh θα προκαλέσει το άμεσο μπλοκάρισμα της IP με επαναρρύθμιση του firewall του endpoint server (με τον ίδιο τρόπο που έγινε στο προηγούμενο σενάριο μελέτης).

Για αυτό το σενάριο μελέτης θα πρέπει να γίνει η εξής παραμετροποίηση στο σύστημα που εξετάζεται:

- Η ρύθμιση της επικοινωνίας του Wazuh με το MISP, με νέα επέκταση του σεναρίου εντολών custom-misp. Για τον σκοπό αυτό, τα βήματα που πρέπει να γίνουν είναι:
    - Πρώτον, πρέπει να αναγνωριστεί ποιο alert παράγεται στο Wazuh manager όταν παρατηρηθεί μια αποτυχημένη απόπειρα σύνδεσης.
    - Δεύτερον, πρέπει να οριστεί ένας κανόνας ώστε την πρώτη φορά που εμφανίζεται αυτό το alert να παράγεται ένα νέο alert, που να επισημαίνει ότι παρατηρήθηκε ένα ύποπτο συμβάν. [186]
    - Τρίτον, πρέπει να δημιουργηθεί ένας νέος κανόνας στο Wazuh, ώστε κάθε φορά που εμφανίζεται αυτό το alert για το ύποπτο συμβάν να καλείται το σενάριο εντολών custom-misp.
    - Τέταρτον, το custom-misp να επεκταθεί με κώδικα που να αντλεί την IP από την οποία προέρχεται αυτό το ύποπτο συμβάν και μέσω της κατάλληλης κλήσης του API του MISP, να του ζητά να εξετάσει αν αυτή η IP υπάρχει σε κάποιο IoC. [187] Αν το MISP επιστρέψει αποτέλεσμα – δηλαδή υπάρχει ταύτιση της IP με κάποιο IoC, τότε το custom-misp αναλαμβάνει να παραγάγει ένα νέο σχετικό alert στο Wazuh. Αν δεν υπάρχει ταύτιση, πάλι το custom-misp θα παραγάγει ένα alert, αλλά αυτό απλά θα ενημερώνει πως δεν βρέθηκε κάποιο IoC.
- Η δημιουργία ενός νέου κανόνα στο Wazuh, που επιβάλει πως η εμφάνιση του alert σχετικά με την ταυτοποίηση της IP σε IoC του MISP, θα προκαλεί την ενεργοποίηση και του αντίμετρου για το μπλοκάρισμά της ως κακόβουλης.

[186] Η χρήση ενός συμπληρωματικού alert είναι απαραίτητη από τεχνικής απόψεως, ώστε να εμφανίζεται μόνο μία φορά σε έναν χρονικό ορίζοντα κάποιων λεπτών. Με αυτό τον τρόπο εξασφαλίζεται πως δεν θα υπάρχουν συνεχείς κλήσεις προς το MISP για την ίδια IP.

[187] Για να έχει αποτέλεσμα αυτή η αναζήτηση, θα πρέπει να έχουμε εξασφαλίσει πως το MISP έχει πρόσβαση σε IoC που περιέχουν γνωστές κακόβουλες IP. Θυμίζουμε ότι στο 2ο σενάριο μιλήσαμε για το LocallyMonitoredMaliciousIPsEvent, που δημιουργείται για τους σκοπούς του σεναρίου μελέτης και συγκεντρώνει όλες τις IP που σχετίζονται με κακόβουλες επιθέσεις στο σύστημά μας.

Όπως και το 2ο σενάριο και αυτό το σενάριο μελέτης εξετάστηκε με δοκιμάστηκε απέναντι σε πραγματικές απειλές και δεν χρειάστηκε κάποια προσομοίωση επίθεσης.

Time	rule.description	rule.level	rule.id
May 18, 2024 @ 19:27:55.321	Active response blocked IP 221.222.184.230	12	100095
May 18, 2024 @ 19:27:55.067	MISP - new IoC created for 221.222.184.230	8	100624
May 18, 2024 @ 19:27:53.312	sshd: brute force trying to get access to the system. Non existent user.	10	5712
May 18, 2024 @ 19:27:49.307	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:47.306	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:41.299	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:39.296	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:35.293	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:33.290	sshd: Attempt to login using a non-existent user	5	5710
May 18, 2024 @ 19:27:33.127	MISP - no IoC found for 221.222.184.230	5	100625
May 18, 2024 @ 19:27:31.288	Suspicious authentication failure. Searching MISP IoCs for IP 221.222.184.230	7	100096
May 18, 2024 @ 19:27:29.286	sshd: Attempt to login using a non-existent user	5	5710

Εικόνα 15: Ανίχνευση ύποπτης αποτυχίας σύνδεσης με αρνητική συσχέτιση με IoC
[η επίθεση εξελίσσεται μέχρι την ανίχνευση και την εξουδετέρωσή της με βάση το σενάριο 2]

Παρακάτω απεικονίζεται μια επίθεση από μια IP που δεν συσχετίστηκε με κάποιο IoC στο MISP.

Όπως βλέπουμε στην παραπάνω εικόνα (από κάτω προς τα πάνω), αρχικά εμφανίζεται ένα alert σχετικά με μια αποτυχημένη απόπειρα σύνδεσης. Αυτό το alert πυροδοτεί το δεύτερο alert που δηλώνει πως ανιχνεύτηκε ύποπτο συμβάν και θα διερευνηθεί αν υπάρχει συσχέτιση της IP με κάποιο IoC στο MISP. Στο επόμενο alert βλέπουμε πως η απάντηση από το MISP ήταν αρνητική. Παρατηρώντας τα επόμενα alert βλέπουμε πως η επίθεση εξελίσσεται κανονικά μέχρι το σημείο που το Wazuh αναγνωρίζει την επίθεση brute force και ενεργοποιεί το μπλοκάρισμα της IP (ακριβώς με τον ίδιο τρόπο που εξηγήθηκε στο 2ο σενάριο μελέτης).

Στην επόμενη εικόνα εξετάζουμε το ίδιο σενάριο με μια άλλη τροπή. Σε αυτή την περίπτωση υπάρχει συσχέτιση της IP με ένα IoC στο MISP και η επίθεση εξουδετερώθηκε άμεσα.

Time	rule.description	rule.level	rule.id
May 18, 2024 @ 20:06:05.551	Active response blocked IP 183.81.169.238 found in MISP IOC 45	12	100094
May 18, 2024 @ 20:06:03.948	MISP - IoC found in Threat Intel - Category: Network activity, Attribute: 183.81.169.238	14	100623
May 18, 2024 @ 20:06:01.546	Suspicious authentication failure. Searching MISP IoCs for IP 183.81.169.238	7	100097
May 18, 2024 @ 20:05:55.539	sshd: authentication failed.	5	5760

Εικόνα 16: Ανίχνευση ύποπτης αποτυχίας σύνδεσης και εξουδετέρωσή μετά από συσχέτιση με IoC

Στην παραπάνω εικόνα (ξεκινώντας πάλι από κάτω προς τα πάνω), βλέπουμε ακριβώς τα ίδια δύο alert με πριν. Η θετική απάντηση όμως από την επικοινωνία με το MISP – που φαίνεται στο επόμενο alert – πυροδοτεί άμεσα το μπλοκάρισμα της IP – όπως φαίνεται στο τελευταίο alert.

Όπως παρατηρούμε, η βελτίωση είναι πολύ σημαντική, αφού ο κίνδυνος αποσοβείται πολύ γρήγορα σε σχέση με την προηγούμενη περίπτωση.

8

Συμπεράσματα

Σημαντικό μέλημα του τομέα της Ασφάλειας Πληροφοριών είναι η μελέτη των ανθρώπινων παραγόντων που βρίσκονται πίσω από κάθε κακόβουλη δραστηριότητα. Η κατανόηση του τρόπου δράσης τους αποτελεί τον ακρογωνιαίο λίθο της αποτελεσματικής αντιμετώπισής τους, καθώς η Ασφάλεια Πληροφοριών προσπαθεί για κάθε δράση τους να υποδείξει την κατάλληλη αντίδραση. Αλλά να πάει και ένα επίπεδο παραπάνω· να προωθήσει μέτρα που μπορούν να λειτουργήσουν προληπτικά και να εκμηδενίσουν τις ευκαιρίες που δίνονται στους κακόβουλους.

Στο πλαίσιο αυτό αναδεικνύονται δύο διαφορετικές προσεγγίσεις: η μία βασίζεται στα εργαλεία IDS / IPS. Υπάρχουν διαφορετικοί τύποι των συστημάτων αυτών, με διαφορετικά χαρακτηριστικά για κάθε τύπο. Κοινό τους γνώρισμα ωστόσο, είναι ότι ενσωματώνουν τόσο λειτουργίες που χαρακτηρίζονται από την υψηλή αξιοπιστία τους (λ.χ. ανίχνευση υπογραφών), όσο και λειτουργίες που εστιάζουν στην ταχύτερη προσαρμογή σε μεταβαλλόμενες απειλές (λ.χ. ανίχνευση ανωμαλιών).

Η άλλη προσέγγιση ουσιαστικά βασίζεται στο ρητό «*η γνώση είναι δύναμη*». Ακολουθώντας τη φιλοσοφία της ανοικτότητας, οι πληροφορίες που συγκεντρώνονται σχετικά με μια απειλή αναλύονται και οργανώνονται με απώτερο σκοπό να διαμοιραστούν. Για αυτόν τον σκοπό αξιοποιούνται συγκεκριμένα εργαλεία, οι πλατφόρμες TIP. Οι χρήστες των TIP έχουν τη δυνατότητα να λάβουν πληροφορίες από άλλους χρήστες, να αναλύσουν αυτή την πληροφόρηση και να καταλήξουν στην κατάλληλη προσαρμογή της προστασίας του δικού τους συστήματος· αλλά παράλληλα και να κοινοποιήσουν γνώση που απέκτησαν οι ίδιοι για να βοηθήσουν τους άλλους.

Το «πάντρεμα» των δύο προσεγγίσεων ανοίγει νέες δυνατότητες. Οι πλατφόρμες TIP παρέχουν πληροφορίες που απευθύνονται όχι μόνο σε ανθρώπους, αλλά εμπλουτίζονται ώστε να είναι αξιοποιήσιμες με έναν αυτόματο τρόπο και από υπολογιστικά συστήματα. Και αντίστροφα, κακόβουλες ενέργειες που μπορούν να αναγνωριστούν από τα IDS / IPS με υψηλή βεβαιότητα, μπορούν με αυτόματο τρόπο να τροφοδοτήσουν μια πλατφόρμα TIP, για να διαμοιραστεί η πληροφορία ταχύτερα.

Στην παρούσα εργασία, αυτό το «πάντρεμα» δοκιμάστηκε με χρήση του IDS / IPS Wazuh και της TIP MISP. Και οι δύο είναι πλατφόρμες ελεύθερου λογισμικού και λογισμικού ανοικτού κώδικα, που επιτρέπουν στον χρήστη τους να τις παραμετροποιήσει ή ακόμα και να τις τροποποιήσει σε όποιο βαθμό κρίνει ότι ταιριάζει στις δικές του ανάγκες. Σε κάθε περίπτωση, η ενοποίηση προσφέρει τη δυνατότητα να επικοινωνούν μεταξύ τους χωρίς τη μεσολάβηση ενός ανθρώπινου χειριστή. Αφού αναπτύχθηκε ο κατάλληλος κώδικας για την αυτοματοποίηση της επικοινωνίας, εξετάστηκαν τρία σενάρια.

Στο πρώτο σενάριο, το Wazuh αρχικά λειτουργεί ως IDS και ανιχνεύει ένα ύποπτο περιστατικό. Τότε, ζητάει από το MISP να του απαντήσει αν σχετίζεται με κάποια επιβεβαιωμένη δράση κακόβουλου λογισμικού. Σε αυτή την περίπτωση, λειτουργεί ως IPS και επιβάλει την εκτέλεση μέτρων για την εξουδετέρωση της απειλής.

Στο δεύτερο σενάριο, τα πράγματα λειτουργούν αντίστροφα. Το Wazuh ανιχνεύει επιτυχώς μια επίθεση, την εξουδετερώνει, αλλά αναλαμβάνει να ειδοποιήσει και το MISP. Το όφελος από αυτό το δεύτερο σενάριο φαίνεται ουσιαστικά στο τρίτο σενάριο μελέτης.

Στο τρίτο λοιπόν σενάριο, το Wazuh διαθέτει έναν πιο «ευαίσθητο» κανόνα για εντοπίσει τα πρώτα βήματα μιας επίθεσης. Μόλις ανιχνεύσει ένα τέτοιο περιστατικό, ζητάει και πάλι από το MISP να ελέγξει αν υπάρχει συσχέτιση με ενδείξεις που είναι γνωστές από προηγούμενες επιθέσεις. Αν η απάντηση είναι θετική, τότε το Wazuh πάλι ενεργεί ως IPS και αναλαμβάνει να εξουδετερώσει την απειλή.

Σε όλες τις περιπτώσεις βλέπουμε ότι ο αντίκτυπος της συνεργασίας των δύο μερών είναι θετικός. Το κύριο γνώρισμα της συνεργασίας είναι πως οι επιθέσεις αποσοβούνται σε σύντομο χρονικό διάστημα και μάλιστα στα πρώτα τους στάδια. Αυτή, η ταχύτερη απόκριση στα περιστατικά (σε σχέση με την αναμονή διερεύνησης από τους αναλυτές και τη μη αυτόματη ενεργοποίηση των μέτρων εξουδετέρωσης), προσφέρει αύξηση της προστασίας του συστήματος, αφού, όπως εξηγήθηκε και στις πρώτες ενότητες, όσο πιο νωρίς αντιμετωπιστεί μια επίθεση, τόσο ελαχιστοποιούνται και οι επιπτώσεις στο σύστημα. Ο επιτιθέμενος στερείται τον απαιτούμενο χρόνο για να προκαλέσει ζημία, ενώ το ίδιο το σύστημα δεν στερείται υπολογιστικούς πόρους (που θα δεσμεύονταν λόγω της κακόβουλης δράσης).

Όπως εξετάσαμε σε αυτή την εργασία, τα είδη των επιθέσεων που μπορούν να εκτελεστούν σε ένα Πληροφοριακό Σύστημα είναι πάρα πολλά. Μέσα από αυτά τα τρία σενάρια, εξετάσαμε και εξερευνήσαμε ένα μικρό μόνο κομμάτι των δυνατοτήτων που προσφέρει η συνεργασία ενός IDS / IPS με μία TIP. Μελλοντικά, θα μπορούσε η συνεργασία αυτή να εξεταστεί με πρόσθετα σενάρια και να διερευνηθεί πώς μπορεί να βελτιωθεί, μέσω της αυτοματοποίησης, η απόκριση και σε άλλου είδους απειλές και επιθέσεις.

Βιβλιογραφία

Alhazmi, O. H., Malaiya, Y. K., & Ray, I. (2007). Measuring, analyzing and predicting security vulnerabilities in software systems. *Computers & Security, 26*(3), σσ. 219-228. doi:https://doi.org/10.1016/j.cose.2006.10.002

Ashoor, A. S., & Gore, S. (2011). Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). *International Conference on Network Security and Applications. CNSA 2011* (σσ. 497-501). Chennai, India: Springer. doi:https://doi.org/10.1007/978-3-642-22540-6_48

Ask, K. (2006). *Automatic Malware Signature Generation.* Master Thesis. Ανάκτηση από https://chschulte.github.io/teaching/theses/ICT-ECS-2006-122.pdf

Azeez, N. A., Bada, T. M., Misra, S., Adewumi, A., Vyver, C. V., & Ahuja, R. (2020). Intrusion Detection and Prevention Systems: An Updated Review. (N. Sharma, A. Chakrabarti, & V. E. Balas, Επιμ.) *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing, 1042*. doi:https://doi.org/10.1007/978-981-32-9949-8_48

Bahrami, P. N., Dehghantanha, A., Dargahi, T., Parizi, R. M., Choo, K.-K. R., & Javadi, H. H. (2019). Cyber Kill Chain-Based Taxonomy of Advanced Persistent Threat Actors: Analogy of Tactics, Techniques, and Procedures. *Journal of Information Processing Systems, 15*(4), σσ. 865 - 889. doi:https://doi.org/10.3745/JIPS.03.0126

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). A Survey of Botnet Technology and Defenses. *2009 Cybersecurity Applications & Technology Conference for Homeland Security* (σσ. 299-304). Washington, DC, USA: IEEE. doi:https://doi.org/10.1109/CATCH.2009.40

Biju, J. M., Gopal, N., & Prakash, A. J. (2019). Cyber attacks and its different types. *International Research Journal of Engineering and Technology (IRJET), 6*(3), σσ. 4849-4852. Ανάκτηση από https://www.irjet.net/archives/V6/i3/IRJET-V6I31244.pdf

BlackBerry. (χ.χ.). *MITRE D3FEND Framework: The Ultimate Guide*. Ανάκτηση Ιανουάριος 2024, από BlackBerry: https://www.blackberry.com/us/en/solutions/endpoint-security/mitre-attack/mitre-defend

Bravo, P., & García, D. F. (2020). *Rootkits Survey: A concealment story.* Ανάκτηση από https://pablo-bravo.com/files/survey.pdf

Bretthauer, D. (2001). *Open Source Software: A History.* UConn Library. Ανάκτηση από https://digitalcommons.lib.uconn.edu/libr_pubs/7

Brown, S. (2023). *3 Types of Access Control: IT Security Models Explained*. Ανάκτηση Ιανουάριος 2024, από StrongDM Blog: https://www.strongdm.com/blog/types-of-access-control

Cárdenas, A. A., Roosta, T., Taban, G., & Sastry, S. (2008). Cyber Security Basic Defenses and Attack Trends. *Homeland Security Technology Challenges*, σσ. 73-101. Ανάκτηση από https://people.eecs.berkeley.edu/~sastry/pubs/Pdfs%20of%202008/CardenasCyber2008.pdf

Carvalho, M., DeMott, J., Ford, R., & Wheeler, D. A. (2014). Heartbleed 101. *IEEE Security & Privacy, 12*(4), σσ. 63 - 67. doi:https://doi.org/10.1109/MSP.2014.66

Castillo, J. A. (2023). *Cyberspace Origin, Applications & Examples*. Ανάκτηση Οκτώβριος 2023, από study.com: https://study.com/academy/lesson/cyberspace-history-origin-overview.html

Cloudflare. (χ.χ.). *What is access control? Authorization vs authentication*. Ανάκτηση Ιανουάριος 2024, από Cloudflare: https://www.cloudflare.com/learning/access-management/what-is-access-control/

Conti, M., Dragoni, N., & Lesyk, V. (2016). A Survey of Man In The Middle Attacks. *IEEE Communications Surveys & Tutorials, 18*(3), σσ. 2027 - 2051. doi:https://doi.org/10.1109/COMST.2016.2548426

Denning, D. E. (2014). Framework and principles for active cyber defense. *Computers & Security, 40*, σσ. 108-113. doi:https://doi.org/10.1016/j.cose.2013.11.004

ENISA. (2023). *What is "Social Engineering"?* Ανάκτηση Νοέμβριος 2023, από https://www.enisa.europa.eu/: https://www.enisa.europa.eu/topics/incident-response/glossary/what-is-social-engineering

Farinholt, B. R. (2019). *Understanding the Remote Access Trojan malware ecosystem through the lens of the infamous DarkComet RAT.* San Diego: University of California. Ανάκτηση από https://escholarship.org/uc/item/3vv544n5

Feily, M., Shahrestani, A., & Ramadass, S. (2009). A Survey of Botnet Technology and Defenses. *2009 Third International Conference on Emerging Security Information, Systems and Technologies* (σσ. 268-273). Αθήνα: IEEE. doi:https://doi.org/10.1109/CATCH.2009.40

Ferdous, S. (2022). Counter Cyber Attack: Issues to be considered. *PSC Journal, 9*(2), σσ. 67-74. Ανάκτηση από https://psc.gov.bd/storage/app/media/uploaded-files/08.%20PSC_Journal_Vol_9_Issue_2_july-dec_2022.pdf#page=76

Folsom, T. C. (2007). Defining Cyberspace (Finding Real Virtue in the Place of Virtual Reality). *Tulane Journal of Technology & Intellectual Property, 9*, σσ. 75-121. Ανάκτηση από https://journals.tulane.edu/index.php/TIP/article/download/2525/2347

Griffin, K., Schneider, S., Hu, X., & Chiueh, T.-c. (2009). Automatic Generation of String Signatures for Malware Detection. *Recent Advances in Intrusion Detection - 12th International Symposium* (σσ. 101–120). Saint-Malo, France: Springer. doi:https://doi.org/10.1007/978-3-642-04342-0_6

Groš, S. (2021). *Myths and Misconceptions about Attackers and Attacks.* doi:https://doi.org/10.48550/arXiv.2106.05702

Guarascio, M., Cassavia, N., Pisani, F. S., & Manco, G. (2022). Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection. *Future Generation Computer Systems, 135*, σσ. 30-43. doi:https://doi.org/10.1016/j.future.2022.04.028

Heartbleed Bug. (2020). Ανάκτηση Δεκέμβριος 2023, από heartbleed.com: https://heartbleed.com/

Hellenica World. (2023). *Κερκόπορτα (υπολογιστής)*. Ανάκτηση Νοέμβριος 2023, από hellenicaworld.com: https://www.hellenicaworld.com/Science/Informatics/gr/Kerkoporta.html

Humayun, M., Niazi, M., Jhanjhi, N., Alshayeb, M., & Mahmood, S. (2020). Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study. *Arabian Journal for Science and Engineering, 45*, σσ. 3171–3189. doi:https://doi.org/10.1007/s13369-019-04319-2

Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains.* Lockheed Martin Corporation. Ανάκτηση από http://gauss.ececs.uc.edu/Project4/Documents/kill-chain.pdf

Imperva. (2023). *DDoS Attacks*. Ανάκτηση Δεκέμβριος 2023, από imperva: https://www.imperva.com/learn/ddos/ddos-attacks/

In.gr. (2013). *Creepware ή ποιος είναι αυτός που σας παρακολουθεί;*. Ανάκτηση Νοέμβριος 2023, από in science (In.gr Τεχνολογία): https://www.in.gr/2013/12/30/in-science/future/creepware-i-poios-einai-aytos-poy-sas-parakoloythei/

Intel. (2007). *Threat Agent Library Helps Identify Information Security Risks.* White Paper, Intel Enterprise Security. Ανάκτηση από https://www.oasis-open.org/committees/download.php/66239/Intel%20Corp_Threat%20Agent%20Library_07-2202w.pdf

Irwin, L. (2020). *Risk terminology: Understanding assets, threats and vulnerabilities*. Ανάκτηση Οκτώβριος 2023, από vigilantsoftware.co.uk: https://www.vigilantsoftware.co.uk/blog/risk-terminology-understanding-assets-threats-and-vulnerabilities

ISO/IEC. (2015). *ISO/IEC 27039:2015(en) Information technology — Security techniques — Selection, deployment and operations of intrusion detection and prevention systems (IDPS)*. Ανάκτηση Οκτώβριος 2023, από https://www.iso.org/: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27039:ed-1:v2:en

Johnson, K. (2023). *Use these 6 user authentication types to secure networks*. Ανάκτηση Ιανουάριος 2024, από TechTarget: https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks

Jouini, M., Rabai, L. B., & Aissa, A. B. (2014). Classification of security threats in information systems. *Procedia Computer Science, 32*, σσ. 489–496. doi:https://doi.org/10.1016/j.procs.2014.05.452

Kamal, S. U., Ali, R. J., Alani, H. K., & Abdulmajed, E. S. (2016). Survey and brief history on malware in network security case study: Viruses, worms and bots. *ARPN Journal of Engineering and Applied Sciences, 2011*(1), σσ. 683-698. Ανάκτηση από https://portal.arid.my/Publications/bc2d592b-a192-4a.pdf

Kimura, M., DeVries, T., & Gordon, S. (2017). *The Cyber Defense (CyDef) Model for Assessing Countermeasure Capabilities.* Albuquerque, New Mexico and Livermore, California: Sandia National Laboratories. Ανάκτηση από https://www.osti.gov/servlets/purl/1367477/

Lai, W. L., Goh, V. T., Yap, T. T., & Ng, H. (2023). Phishing and Spoofing Websites: Detection and Countermeasures. *International Journal on Advanced Science, Engineering and Information Technology, 13*(5), σσ. 1672-1678. doi:https://doi.org/10.18517/ijaseit.13.5.19037

Lee, M. (2023). A Survey on the Real-world Cyberattacks on the Industrial Internet of Things. *TechRxiv*. doi:https://doi.org/10.36227/techrxiv.22307413.v2

Li, C., Jiang, W., & Zou, X. (2009). Botnet: Survey and Case Study. *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)* (σσ. 1184-1187). Kaohsiung, Taiwan: IEEE. doi:https://doi.org/10.1109/ICICIC.2009.127

LinkedIn. (2024). *How do you create and use malware signatures and indicators of compromise?* Ανάκτηση Φεβρουάριος 2024, από LinkedIn: https://www.linkedin.com/advice/3/how-do-you-create-use-malware-signatures-indicators

Mallikarjunan, K., K.Muthupriya, & Shalinie, S. (2016). A survey of Distributed Denial of Service attack. *10th International Conference on Intelligent Systems and Control (ISCO)* (σσ. 1-6). Coimbatore, India: IEEE. doi:https://doi.org/10.1109/ISCO.2016.7727096

McHugh, J. (2001). Intrusion and intrusion detection. *International Journal of Information Security , 1*, σσ. 14–35. doi:https://doi.org/10.1007/s102070100001

Medlock, B. (2023). *Different types of DDoS attacks: how to protect your clients*. Ανάκτηση Δεκέμβριος 2023, από ConnectWise: https://www.connectwise.com/blog/cybersecurity/types-of-ddos-attacks

MISP. (2022). *MISP - User Guide A Threat Sharing Platform.* Ανάκτηση από Computer Incident Response Center Luxembourg (CIRCL): https://www.circl.lu/doc/misp/book.pdf

MISP. (χ.χ.). *Home*. Ανάκτηση Μάιος 2024, από MISP Threat Sharing: https://www.misp-project.org/

MISP. (χ.χ.). *WHO*. Ανάκτηση Μάιος 2024, από MISP Threat Sharing: https://www.misp-project.org/who/

MITRE. (2023). *About the D3FEND Knowledge Graph Project*. Ανάκτηση Ιανουάριος 2024, από MITRE D3FEND: https://d3fend.mitre.org/about/

MITRE. (2023). *CWE Common Weakness Enumeration*. Ανάκτηση από CWE Common Weakness Enumeration: https://cwe.mitre.org/about/new_to_cwe.html

MITRE. (2024). *MITRE ATT&CK*. Ανάκτηση Ιανουάριος 2024, από MITRE ATT&CK: https://attack.mitre.org/

MITRE. (χ.χ.). *CVE*. Ανάκτηση από CVE: https://www.cve.org/About/RelatedEfforts

NCSC. (2015). *How cyber attacks work*. Ανάκτηση Ιανουάριος 2024, από National Cyber Security Centre - NCSC.GOV.UK: https://www.ncsc.gov.uk/information/how-cyber-attacks-work

Ngo, F. T., Agarwal, A., Ramakrishna, G., & MacDonald, C. (2020). Malicious software threats. (T. J. Holt, & A. M. Bossler, Επιμ.) *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, σσ. 793-813. doi:https://doi.org/10.1007/978-3-319-78440-3_35

NIST. (2022). *NIST SP 800-160v1r1 - Engineering Trustworthy Secure Systems.* NIST Special Publication. doi:https://doi.org/10.6028/NIST.SP.800-160v1r1

NIST. (χ.χ.). *advanced persistent threat*. Ανάκτηση Δεκέμβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/advanced_persistent_threat

NIST. (χ.χ.). *attacker*. Ανάκτηση Οκτώβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/attacker

NIST. (χ.χ.). *countermeasures*. Ανάκτηση Ιανουάριος 2024, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/countermeasures

NIST. (χ.χ.). *impact*. Ανάκτηση Οκτώβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/impact

NIST. (χ.χ.). *INFOSEC*. Ανάκτηση Οκτώβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/infosec

NIST. (χ.χ.). *plaintext*. Ανάκτηση Δεκέμβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/plaintext

NIST. (χ.χ.). *Trojan horse*. Ανάκτηση Νοέμβριος 2023, από NIST Computer Security Resource Center - Glossary: https://csrc.nist.gov/glossary/term/trojan_horse

Okta. (2023). *Authentication vs. Authorization*. Ανάκτηση Ιανουάριος 2024, από okta.com: https://www.okta.com/identity-101/authentication-vs-authorization/

Open Source Initiative. (2024). *The Open Source Definition*. Ανάκτηση Φεβρουάριος 2024, από Open Source Initiative: https://opensource.org/osd/

Oppliger, R., & Gajek, S. (2005). Effective Protection Against Phishing and Web Spoofing. *Communications and Multimedia Security - 9th IFIP TC-6 TC-11 International Conference* (pp. 32-45). Salzburg, Austria: Springer. doi:https://doi.org/10.1007/11552055_4

Othman, S. M., T. Alsohybe, N., Ba-Alwi, F. M., & Zahary, A. T. (2018). Survey on Intrusion Detection System Types. *International Journal of Cyber-Security and Digital Forensics, 7*(4), σσ. 444-463. Ανάκτηση από https://www.researchgate.net/profile/Ammar-Zahary/publication/329360916_Survey_on_Intrusion_Detection_System_Types/links/5c0454f792851c63cab623f0/Survey-on-Intrusion-Detection-System-Types.pdf

OWASP. (χ.χ.). *Cross Site Scripting (XSS)*. Ανάκτηση Δεκέμβριος 2023, από OWASP: https://owasp.org/www-community/attacks/xss/

Panigrahi, K. K. (2022). *Difference between Threat and Attack*. Ανάκτηση Οκτώβριος 2023, από Tutorials Point: https://www.tutorialspoint.com/difference-between-threat-and-attack

Praseed, A., & Thilagam, P. S. (2019). DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications. *IEEE Communications Surveys & Tutorials, 21*(1), σσ. 661 - 685. doi:https://doi.org/10.1109/COMST.2018.2870658

Preuveneers, D., & Joosen, W. (2021). Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence. *Cybersecurity and Privacy, 1*, σσ. 140–163. doi:https://dx.doi.org/10.3390/jcp1010008

Rad, B. B., Masrom, M., & Ibrahim, S. (2011). Evolution of Computer Virus Concealment and Anti-Virus Techniques: A Short Survey. *IJCSI International Journal of Computer Science Issues, 8*(1), σσ. 113-121. Ανάκτηση από https://arxiv.org/ftp/arxiv/papers/1104/1104.1070.pdf

Ramsdale, A., Shiaeles, S., & Kolokotronis, N. (2020). A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages. *Electronics, 9*(5). doi:https://doi.org/10.3390/electronics9050824

Raymond, E. S. (1999). A Brief History of Hackerdom. In *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (Vol. 12). O'Reilly & Associates. Retrieved from http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history/

Raymond, E. S. (1999). The Cathedral and the Bazaar. Στο *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly. Ανάκτηση από http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

Raymond, E. S. (1999). The Origins of 'Open Source'. Στο *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly & Associates. Ανάκτηση από http://www.catb.org/~esr/writings/cathedral-bazaar/

Rodríguez-Gómez, R. A., Maciá-Fernández, G., & García-Teodoro, P. (2013). Survey and Taxonomy of Botnet Research through Life-Cycle. *ACM Computing Surveys*, σσ. 1–33. doi:https://doi.org/10.1145/2501654.2501659

Roundy, K. A., Mendelberg, P. B., Dell, N., McCoy, D., Nissani, D., Ristenpart, T., & Tamersoy, A. (2020). The Many Kinds of Creepware Used for Interpersonal Attacks. *2020 IEEE Symposium on Security and Privacy* (σσ. 626-643). San Francisco, CA, USA: IEEE. doi:https://doi.org/10.1109/SP40000.2020.00069

Rouse, M. (2024). *Cyber Defense*. Ανάκτηση Ιανουάριος 2024, από Techopedia: https://www.techopedia.com/definition/6705/cyber-defense

Rudd, E. M., Rozsa, A., Günther, M., & Boult, T. E. (2017). A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions. *IEEE Communications Surveys & Tutorials, 19*(2), σσ. 1145-1172. doi:https://doi.org/10.1109/COMST.2016.2636078

Saeed, I. A., Selamat, A., & Abuagoub, A. M. (2013). A Survey on Malware and Malware Detection Systems. *International Journal of Computer Applications, 67*(16), σσ. 25-31. Ανάκτηση από https://www.researchgate.net/profile/Imtithal-Saeed/publication/272238656_A_Survey_on_Malwares_and_Malware_Detection_Systems/links/566284c608ae192bbf8cf1a5/A-Survey-on-Malwares-and-Malware-Detection-Systems.pdf

Samonas, S., & Coss, D. (2014). The CIA strikes back: Redefining Confidentiality, Integrity and Availability in Security. *Journal of Information System Security, 10*(3), σσ. 21–45. Ανάκτηση από https://www.proso.com/dl/Samonas.pdf

Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS) - NIST SP 800-94.* NIST. Ανάκτηση από https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-94.pdf

Shabut, A. M., Lwin, K. T., & Hossain, M. A. (2016). Cyber Attacks, Countermeasures, and Protection Schemes – A State of the Art Survey. *10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)* (σσ. 37-44). Chengdu, China: IEEE. doi:https://doi.org/10.1109/SKIMA.2016.7916194

Shacklett, M. E. (2021). *attack vector*. Ανάκτηση Οκτώβριος 2023, από techtarget.com: https://www.techtarget.com/searchsecurity/definition/attack-vector

Shah, D., Shah, V., & Shah, H. (2017). Survey on Computer Worms. *International Journal on Recent and Innovation Trends in Computing and Communication, 5*(8), σσ. 184 – 194. Ανάκτηση από https://www.researchgate.net/publication/341271183_Survey_on_Computer_Worms

Shahriar, H., & Zulkernine, M. (2012). Mitigating Program Security Vulnerabilities: Approaches and Challenges. *ACM Computing Surveys, 44*(3), σσ. 1-46. doi:http://doi.acm.org/10.1145/2187671.2187673

Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. (2009). AVOIDIT: A Cyber Attack Taxonomy. *University of Memphis, Technical Report CS-09-003*. Ανάκτηση από https://www.researchgate.net/profile/S-Shiva/publication/229020163_AVOIDIT_A_Cyber_Attack_Taxonomy/links/544e58360cf2bca5ce90aebb/AVOIDIT-A-Cyber-Attack-Taxonomy.pdf

Sinha, P., Rai, A. k., & Bhushan, B. (2019). Information Security threats and attacks with conceivable counteraction. *2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (σσ. 1208-1213). Kannur, India: IEEE. doi:https://doi.org/10.1109/ICICICT46008.2019.8993384

Solms, B. v., & Solms, R. v. (2017). Cybersecurity and information security – what goes where? *Information and Computer Security, 26*(1). doi:https://doi.org/10.1108/ICS-04-2017-0025

Solms, R. v., & Niekerk, J. v. (2013). From information security to cyber security. *Computers & Security, 38*, σσ. 97-102. doi:https://doi.org/10.1016/j.cose.2013.04.004

Sommestad, T., Holm, H., & Ekstedt, M. (2012). Estimates of success rates of remote arbitrary code execution attacks. *Information Management & Computer Security, 20*(2), σσ. 107-122. doi:https://doi.org/10.1108/09685221211235625

swetha_vazhakkat. (2022). *Difference between Threat and Attack*. (pp_pankaj, & raj2002, Επιμελητές) Ανάκτηση Οκτώβριος 2023, από geeksforgeeks: https://www.geeksforgeeks.org/difference-between-threat-and-attack/

Synopsys. (χ.χ.). *Cross Site Request Forgery*. Ανάκτηση Δεκέμβριος 2023, από Synopsys - Glossary: https://www.synopsys.com/glossary/what-is-csrf.html

Tekiner, E., Acar, A., Uluagac, A. S., Kirda, E., & Selcuk, A. A. (2021). SoK: Cryptojacking Malware. *2021 IEEE European Symposium on Security and Privacy (EuroS&P)* (σσ. 120-139). IEEE. doi:https://doi.org/10.1109/EUROSP51992.2021.00019

The Virus Encyclopedia. (2023). *virus*. Ανάκτηση Νοέμβριος 2023, από virus.wikidot.com: http://virus.wikidot.com/virus

Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M. A., & Rashid, A. (2018). Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications, 101*(1), σσ. 18-54. doi:https://doi.org/10.1016/j.jnca.2017.10.016

Uma, M., & Padmavathi, G. (2013). A Survey on Various Cyber Attacks and Their Classification. *International Journal of Network Security, 15*(5), σσ. 390-396. Ανάκτηση από https://web.archive.org/web/20190804170947id_/http://ijns.jalaxy.com.tw/contents/ijns-v15-n5/ijns-2013-v15-n5-p390-396.pdf

Vanitha, K., UMA, S. V., & Mahidhar, S. (2017). Distributed Denial of Service: Attack techniques and mitigation. *Proceeding of Second International conference on Circuits, Controls and Communications* (σσ. 226-231). Bangalore, India: IEEE. doi:https://doi.org/10.1109/CCUBE.2017.8394146

Vidalis, S., & Jones, A. (2005). Analyzing Threat Agents & Their Attributes. *ECIW*. Ανάκτηση από https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cbe8048d145d6f48a9c6aea3487d204e238c22dc

Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A. (2016). MISP - The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, σσ. 49-56. doi:http://dx.doi.org/10.1145/2994539.2994542

Wazuh, Inc. (2023). *Migrating from OSSEC to Wazuh*. Ανάκτηση Μάιος 2024, από wazuh.com: https://wazuh.com/blog/migrating-from-ossec-to-wazuh/

Wazuh, Inc. (2024). *One unified platform for complete protection*. Ανάκτηση Μάιος 2024, από wazuh.com: https://wazuh.com/platform/overview/

WhatIs.com. (2023). *boot sector virus*. Ανάκτηση Νοέμβριος 2023, από www.techtarget.com: https://www.techtarget.com/whatis/definition/boot-sector-virus

WhatIs.com. (2023). *bot*. Ανάκτηση Νοέμβριος 2023, από techtarget.com: https://www.techtarget.com/whatis/definition/bot-robot

WhatIs.com. (2023). *macro virus*. Ανάκτηση Νοέμβριος 2023, από www.techtarget.com: https://www.techtarget.com/searchsecurity/definition/macro-virus

WhatIs.com. (2023). *virus (computer virus)*. Ανάκτηση Νοέμβριος 2023, από www.techtarget.com: https://www.techtarget.com/searchsecurity/definition/virus

Wu, J.-x., Li, J.-h., & Ji, X.-s. (2018). Security for cyberspace: challenges and opportunities. *Frontiers of Information Technology & Electronic Engineering, 19*, σσ. 1459–1461. doi:https://doi.org/10.1631/FITEE.1840000

Xcitium. (2022). *What Is Antimalware (Anti-Malware)?* Ανάκτηση από Xcitium: https://www.xcitium.com/blog/malware/what-is-antimalware/

Xcitium. (2024). *Cyber Kill Chain vs Mitre Att&ck: What's the Difference?* Ανάκτηση Ιανουάριος 2024, από Xcitium: https://www.xcitium.com/cyber-kill-chain-vs-mitre-att&ck/

Yasar, K. (2024). *man-in-the-middle attack (MitM)*. Ανάκτηση Ιανουάριος 2024, από TechTarget: https://www.techtarget.com/iotagenda/definition/man-in-the-middle-attack-MitM

Yasin, D. (2019). *Crypto Security: What Is Cryptojacking? How to Prevent and Defend Against It?* Ανάκτηση Δεκέμβριος 2023, από CryptoPotato: https://cryptopotato.com/crypto-security-what-is-cryptojacking-how-to-prevent-and-defend-against-it/

Yee, C. K., & Zolkipli, M. F. (2021). Review on Confidentiality, Integrity and Availability in Information Security. *Journal of ICT in Education, 8*(2), σσ. 34-42. doi:https://doi.org/10.37134/jictie.vol8.2.4.2021

Yuchong, L., & Qinghui, L. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports, 7*, σσ. 8176-8186. doi:https://doi.org/10.1016/j.egyr.2021.08.126

Zheng, Y., & Zhang, X. (2013). Path Sensitive Static Analysis of Web Applications for Remote Code Execution Vulnerability Detection. *2013 35th International Conference on Software Engineering (ICSE)* (σσ. 652-661). San Francisco, CA, USA: IEEE. doi:https://doi.org/10.1109/ICSE.2013.6606611

Παράρτημα Ι [Το αρχείο custom-misp]

Η συνεργασία του Wazuh με το MISP εξαρτάται από το σενάριο εντολών *custom-misp*. Το σενάριο αυτό είναι γραμμένο σε γλώσσα Python και εκκινείται από το Wazuh μετά από την εμφάνιση συγκεκριμένων alert. Σκοπός του είναι να διακινεί και να «μεταφράζει» δεδομένα μεταξύ του Wazuh και του MISP. Αξίζει ιδιαίτερα να τονιστεί ότι το σενάριο προέρχεται από την τροποποίηση του αρχείου εντολών *MISP API Integration*, που διατίθεται ως ανοικτός κώδικας [188].

Παρακάτω παρουσιάζονται τρία αποσπάσματα του τροποποιημένου κώδικα. Σε καθένα από τα αποσπάσματα ελέγχονται κάποιες τιμές του alert που πυροδότησε το σενάριο και ανάλογα εκτελείται ο απαιτούμενος κώδικας που παράγει και εκτελεί μια κλήση στο MISP μέσω του API του. Ανάλογα με την απάντηση του MISP, το σενάριο παράγει κάποιο νέο alert στο Wazuh μέσω του οποίου του διαβιβάσει την πληροφορία.

Το πρώτο απόσπασμα αφορά την ανίχνευση μια απειλής με βάση την υπογραφή ενός αρχείου.

```
## MISP is queried whether the file's md5 hash belongs to a known threat.
if event_source == "ossec" and event_type == "syscheck_entry_added":
    try:
        wazuh_event_param = alert["syscheck"]["md5_after"]
    except IndexError:
        sys.exit()
    # Form the full MISP API search query
    misp_apicall_url = misp_api_url + "/attributes/restSearch/" + "value:"
f"{wazuh_event_param}"
    try:
        misp_api_response = requests.get(
            misp_apicall_url, headers=misp_apicall_headers, verify=False
        )
    except ConnectionError:
        …
    else:
        misp_api_response = misp_api_response.json()
        # Check if response includes Attributes (IoCs)
        if misp_api_response["response"]["Attribute"]:
            # Generate Alert Output from MISP Response
            alert_output["misp"] = {}
            alert_output["integration"] = "misp"
            alert_output["misp"]["event_id"] =
misp_api_response["response"]["Attribute"][0]["event_id"]
            alert_output["misp"]["category"] =
misp_api_response["response"]["Attribute"][0]["category"]
            alert_output["misp"]["value"] =
misp_api_response["response"]["Attribute"][0]["value"]
            alert_output["misp"]["type"] =
misp_api_response["response"]["Attribute"][0]["type"]
            # The alert also holds the malicious file path
            alert_output["file_path"] = alert["syscheck"]["path"]
            send_event(alert_output, alert["agent"])
```

[188] Από την ιστοσελίδα: https://github.com/karelumair/MISP-Wazuh-Integration/blob/main/custom-misp.py

Το επόμενο απόσπασμα αφορά την καταχώρηση στο MISP ενός IoC που δημιουργείται μετά από την ανίχνευση μιας επίθεσης brute force. Η καταχώρηση γίνεται σε ένα προκαθορισμένο event που δημιουργείται στο MISP για τέτοιες επιθέσεις.

```
## When an a ssh brute force attack is monitored by Wazuh,
## Wazuh's Active Response will block the attacker's IP
## and then call this script to add the IP in an predefined MISP event.
elif event_source == "syslog" and event_type == "authentication_failures":
    try:
        # Open the options file to get the options JSON object
        options_file = open(options_file_name)
        options = json.loads(options_file.read())
        options_file.close()
        # Extract the name of the predefined MISP event
        misp_event_name = options["MISP_event_name"]
    except:
        …
    else:
        misp_apicall_url = misp_api_url + "/events/index/searcheventinfo:"
f"{misp_event_name}"
        try:
            misp_api_response = requests.post(
                misp_apicall_url, headers=misp_apicall_headers, verify=False
            )
        except ConnectionError:
            …
        else:
            misp_api_response = misp_api_response.json()
            # If the MISP event does not exist, then create one!
            if misp_api_response == []:
                misp_data = {
                    "info": f"{misp_event_name}"
                }
                misp_apicall_url = misp_api_url + "/events/add"
                try:
                    misp_api_response = requests.post(
                        misp_apicall_url, headers=misp_apicall_headers,
json=misp_data, verify=False
                    )
                except ConnectionError:
                    …
                else:
                    misp_api_response = misp_api_response.json()
                    # Grab the new MISP event's id
                    misp_event_id = misp_api_response["Event"]["id"]
                    # EXTRA start
                    # Now add an the tag "automatic-collection" for this new event
                    misp_apicall_url = misp_api_url + "/events/addTag/"
f"{misp_event_id}" "/2/local:{local}"
                    try:
                        misp_api_response = requests.post(
                            misp_apicall_url, headers=misp_apicall_headers,
verify=False
                        )
                    except ConnectionError:
                        …
```

```
            # The MISP event exists, so just grab it's id
            else:
                misp_event_id = misp_api_response[0]["id"]
            # Now add the malicious IP as an attribute to the predined MISP event
            try:
                wazuh_event_param = alert["data"]["srcip"]
            except IndexError:
                sys.exit()
            misp_apicall_url = misp_api_url + "/attributes/add/"
f"{misp_event_id}"
            misp_data = {
                "category": "Network activity",
                "type": "ip-src",
                "value": f"{wazuh_event_param}"
            }
            try:
                 misp_api_response = requests.post(
                     misp_apicall_url, headers=misp_apicall_headers,
json=misp_data, verify=False
                 )
            except ConnectionError:
                …
            else:
                misp_api_response = misp_api_response.json()
                # Check if response is success
                if misp_api_response["Attribute"]:
                    # Generate Alert Output from MISP Response
                    alert_output["misp"] = {}
                    alert_output["integration"] = "misp"
                    alert_output["misp"]["ioc_creation"] = "SUCCESS"
                    alert_output["misp"]["value"] = wazuh_event_param
                    send_event(alert_output, alert["agent"])
```

Το τρίτο απόσπασμα κώδικα ενεργοποιείται όταν αναγνωρίζεται ένα ύποπτο συμβάν (αποτυχημένη σύνδεση από ανύπαρκτο χρήστη του συστήματος – κάτι τέτοιο είναι χαρακτηριστικό μιας επίθεσης brute force). Σε αυτή την περίπτωση το σενάριο αναλαμβάνει να αναζητήσει αν υπάρχει IoC στο MISP και να παραγάγει το ανάλογο alert στο Wazuh.

```
## When an authentication failure event is monitored by Wazuh,
## MISP is queried whether the source IP is found in an event.
elif event_source == "misp" and event_type == "authentication_failed":
    try:
        wazuh_event_param = alert["data"]["srcip"]
    except IndexError:
        sys.exit()
    # Form the full MISP API search query
    misp_apicall_url = misp_api_url + "/attributes/restSearch/" + "value:"
f"{wazuh_event_param}"
    try:
        misp_api_response = requests.get(
            misp_apicall_url, headers=misp_apicall_headers, verify=False
        )
    except ConnectionError:
        …
```

```
 else:
        misp_api_response = misp_api_response.json()
        # Check if response includes Attributes (IoCs)
        if misp_api_response["response"]["Attribute"]:
            # Generate Alert Output from MISP Response
            alert_output["misp"] = {}
            alert_output["integration"] = "misp"
            alert_output["misp"]["event_id"] =
misp_api_response["response"]["Attribute"][0]["event_id"]
            alert_output["misp"]["category"] =
misp_api_response["response"]["Attribute"][0]["category"]
            alert_output["misp"]["value"] =
misp_api_response["response"]["Attribute"][0]["value"]
            alert_output["misp"]["type"] =
misp_api_response["response"]["Attribute"][0]["type"]
            # The alert also holds the malicious IP
            alert_output["srcip"] = wazuh_event_param
            send_event(alert_output, alert["agent"])
        else:
            #If no IoC is matched, notify for that
            alert_output["misp"] = {}
            alert_output["integration"] = "misp"
            alert_output["misp"]["ioc_match"] = "FAILURE"
            alert_output["srcip"] = wazuh_event_param
            send_event(alert_output, alert["agent"])
…
```

Παράρτημα ΙΙ [Κανόνες παραμετροποίησης του Wazuh]

Η ενοποίηση της λειτουργίας του Wazuh με το MISP απαιτεί την παραμετροποίηση με τη δημιουργία νέων κανόνων του Wazuh. Τόσο η ενεργοποίηση ή η απενεργοποίηση κάποιων δυνατοτήτων του Wazuh, όσο και ο ορισμός νέων alert γίνεται με τη χρήση κανόνων, που περιγράφονται σε μορφή XML.

Ένα ενδεικτικό απόσπασμα κανόνων που δημιουργήθηκαν στα πλαίσια της εργασίας για τον ορισμό νέων alert που σχετίζονται με τα μέτρα απόκρισης του Wazuh είναι:

```
<group name="misp,">
<!-- Rules for removing MISP matched malware with Active Response -->
  <rule id="100092" level="12">
    <if_sid>657</if_sid>
    <match>Successfully removed threat</match>
    <description>Active response removed threat $(parameters.alert.data.file_path)
confirmed in MISP IOC $(parameters.alert.data.misp.event_id)</description>
  </rule>
  <rule id="100093" level="12">
    <if_sid>657</if_sid>
    <match>Error removing threat</match>
    <description>Error removing threat located in MISP IOC</description>
  </rule>
  <!-- Rule for blocking a MISP matched malicious IP with Active Response -->
  <rule id="100094" level="12">
    <if_sid>651</if_sid>
    <field name="parameters.alert.rule.id">100623</field>
    <description>Active response blocked IP $(parameters.alert.data.misp.value)
found in MISP IOC $(parameters.alert.data.misp.event_id)</description>
  </rule>
  <!-- Wrapper rule for blocking a malicious IP with Active Response - this will
trigger a MISP search -->
  <rule id="100095" level="12">
    <if_sid>651</if_sid>
    <field name="parameters.alert.rule.id">5712|5763</field>
    <description>Active response blocked IP
$(parameters.alert.data.srcip)</description>
  </rule>
<!-- Rules for triggering a MISP IoC search for an IP -->
  <rule id="100096" level="7" ignore="300">
    <if_matched_sid>5710</if_matched_sid>
    <same_srcip />
    <description>Suspicious authentication failure. Searching MISP IoCs for IP
$(srcip)</description>
    <group>sshd,authentication_failed,</group>
  </rule>
  <rule id="100097" level="7" ignore="300">
    <if_matched_sid>5760</if_matched_sid>
    <same_srcip />
    <description>Suspicious authentication failure. Searching MISP IoCs for IP
$(srcip)</description>
    <group>sshd,authentication_failed,</group>
  </rule>
</group>
```

Για να λειτουργήσουν οι παραπάνω κανόνες είναι απαραίτητη η κάτωθι παραμετροποίηση, που καθορίζει μια σειρά από λεπτομέρειες για την επικοινωνία του Wazuh με το MISP. Ενδεικτικά μπορούμε να δούμε σε ποια διεύθυνση επικοινωνήσει το custom-misp με το API του MISP, ποιο API key χρησιμοποιεί [189], για ποιους κανόνες ενεργοποιείται το σενάριο και ποιο είναι το όνομα του event στο οποίο θα καταχωρεί τις IP που εμπλέκονται σε μια επίθεση brute force [190].

```
<!-- Custom external Integration for MISP -->
<ossec_config>
  <integration>
    <name>custom-misp</name>
    <hook_url>https://10.0.0.4</hook_url>
    <api_key>Rrci5ck9g…JLpN0wn…</api_key>
    <alert_format>json</alert_format>
    <!-- Optional filters -->
    <rule_id>554, 5710, 5760, 5712, 5763</rule_id>
    <options>{"MISP_event_name":
"LocallyMonitoredMaliciousIPsEvent"}</options>
  </integration>
</ossec_config>
```

[189] Αυτό το κλειδί δημιουργήθηκε στο MISP και αξιοποιείται για να πιστοποιήσει ότι το ερώτημα που καταφτάνει στο MISP προέρχεται από μια νόμιμη οντότητα (χρήστη ή εφαρμογή) του συνολικότερου συστήματος.

Οι κανόνες της Ασφάλειας Πληροφοριών επιβάλλουν ότι αυτού του είδους τα κλειδιά πρέπει να προφυλάσσονται και να μην κοινοποιούνται. Ακολουθώντας αυτό το πνεύμα, παραπάνω αποκαλύπτονται μόνο μερικά τμήματά του και όχι ολόκληρο.

[190] Για την κατανόηση της αξίας αυτού του event, δείτε και τις υποσημειώσεις 183 και 187.

Η σελίδα αυτή είναι σκόπιμα λευκή