Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Αυτοματοποίηση Εκτίμησης Επικινδυνότητας

Κωνσταντίνος Κοντόπουλος

Επιβλέπων Καθηγητής
Ιωάννης Μαυρίδης

Μάιος 2022

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Αυτοματοποίηση Εκτίμησης Επικινδυνότητας

Κωνσταντίνος Κοντόπουλος

Επιβλέπων Καθηγητής
Ιωάννης Μαυρίδης

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

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2022

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή εντάσσεται στο πεδίο της ασφάλειας πληροφοριών και ειδικότερα της διαχείρισης αυτής (information security management). Η προστασία από πιθανές διαδικτυακές επιθέσεις είναι ένα πολύ σημαντικό ζήτημα, καθώς πλέον χρησιμοποιούνται πληθώρα υπολογιστών και άλλων συσκευών που είναι συνδεδεμένα στο διαδίκτυο.

Στους σκοπούς της παρούσας μεταπτυχιακής διατριβής συμπεριλαμβάνεται η κωδικοποίηση των ζητημάτων που παρουσιάζονται κατά την αυτοματοποίηση της εκτίμησης επικινδυνότητας υπολογιστικών και διαδικτυακών υποδομών. Από μελέτη της σχετικής βιβλιογραφίας εντοπίζονται και αξιολογούνται διάφορες προσεγγίσεις στο ζήτημα, ενώ επιλέγεται μία από αυτές και συγκεκριμένα η μεθοδολογία ARES, στην οποία στηρίζεται η έρευνα που διεξάγεται στην διατριβή. Το αποτέλεσμα της διατριβής είναι η ανάπτυξη μιας εφαρμογής λογισμικού, με την ονομασία Automatic Risk Assessment Tool (ARAT), με την οποία υπολογίζεται το συνολικό επίπεδο επικινδυνότητας ενός συνόλου υπολογιστικών και δικτυακών συστημάτων, αλλά και τα επίπεδα επικινδυνότητας των επί μέρους στοιχείων που τα απαρτίζουν.

Η παρούσα μεταπτυχιακή διατριβή αναπτύσσεται σε 6 κεφάλαια. Στο 1ο κεφάλαιο παρουσιάζεται μια σύντομη εισαγωγή στο ζήτημα της επικινδυνότητας στην κυβερνοασφαλεια. Στο 2ο κεφάλαιο παρέχονται οι βασικές έννοιες που είναι απαραίτητες για την παρουσίασή των επομένων κεφαλαίων. Στο 3ο κεφάλαιο παρουσιάζονται μεθοδολογίες εκτίμησης επικινδυνότητας υπολογιστικών και δικτυακών συστημάτων , καθώς και η επιλεγμένη μεθοδολογία ARES η οποία παρουσιάζεται αναλυτικότερα. Στο 4ο κεφάλαιο παρουσιάζεται η ανάπτυξη του λογισμικού Automatic Risk Assessment Tool (A.R.A.T), το οποίο υλοποιεί την μεθοδολογία ARES και αποτελεί το ερευνητικό μέρος της μεταπτυχιακής διατριβής. Στο 5ο κεφάλαιο παρουσιάζεται η λειτουργία του Automatic Risk Assessment Tool v1.0 (A.R.A.T) σε ένα συγκεκριμένο υπολογιστικό και δικτυακό σύστημα και εξάγονται τα αποτελέσματα σχετικά με τα επίπεδα επικινδυνότητάς του, ενώ παρατίθεται συζήτηση σχετικά με τα παραγόμενα αποτελέσματα. Το 6ο και τελευταίο κεφάλαιο είναι ο επίλογος της μεταπτυχιακής διατριβής.

Summary

This M.Sc. dissertation is part of the field of information security and in particular its management (information security management). Protection against possible cyber attacks is a very important issue, as a variety of computers and other devices connected to the Internet are now used.

The purposes of this master's thesis include the codification of the issues that arise during the automation of the risk assessment of computer and network infrastructure. From the study of the relevant literature, various approaches to the issue are identified and evaluated, while one of them is selected, specifically the ARES methodology, on which the research conducted in the dissertation is based. The result of the dissertation is the development of a software application, called Automatic Risk Assessment Tool (ARAT), which calculates the total level of risk of a set of computer and network systems, but also the level of risk of the individual components that make them up.

This M.Sc. dissertation is part of the field of computer systems security. The security of computer systems against possible online attacks against them is a very important issue, as a plethora of personal computers and other devices are connected to the Internet are now used.

The present postgraduate dissertation is developed in 6 chapters. Chapter 1 provides a brief introduction to the issue of cyber security risk. Chapter 2 provides the basic concepts that are necessary for the presentation of the following chapters. Chapter 3 presents methodologies for risk assessment of computer and network systems, as well as the selected ARES methodology which is presented in more detail. Chapter 4 presents the development of the Automatic Risk Assessment Tool (A.R.A.T) software, which implements the ARES methodology and is the research part of the master's thesis. Chapter 5 presents the operation of the Automatic Risk Assessment Tool (A.R.A.T) on a specific computer and network system and outlines the results regarding its risk levels, while discussing the results produced. The 6th and last chapter is the epilogue of the postgraduate dissertation.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ιωάννη Μαυρίδη για την πολύτιμη συμβολή του στην εκπόνηση αυτής της μεταπτυχιακής διατριβής. Επίσης, θα ήθελα να ευχαριστήσω την υπέροχη σύζυγό μου και την οικογένεια μου για την συμπαράστασή τους και την υποστήριξή τους κατά τη διάρκεια εκπόνησης της διατριβής μου.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Με την συνεχώς αυξανόμενη χρήση του διαδικτύου, όχι μόνο σε ηλεκτρονικούς υπολογιστές, αλλά και σε μια πληθώρα συσκευών που είναι συχνά συνδεδεμένες στο διαδίκτυο (κινητά τηλέφωνα, έξυπνες ηλεκτρικές και ηλεκτρονικές συσκευές, μικροελεγκτές κλπ), η ανάγκη για προστασία από κυβερνοεπιθέσεις είναι απαραίτητη. Το Διαδίκτυο, ένα πολύ σημαντικό εργαλείο και σημαντικός καταλύτης της τέταρτης βιομηχανικής επανάστασης, γνωστής και ως Industry 4.0, είναι ταυτόχρονα το μέσο μέσω του οποίου συντελείται πλήθος κυβερνοεπιθέσεων, ενώ η χρήση κατάλληλων εργαλείων, όπως πχ firewalls, αποτρέπει μέρος αυτών.

1.1 Σκοπός έρευνας

Η παρούσα μεταπτυχιακή διατριβή έχει ως αντικείμενο τον εντοπισμό των ευπαθειών υπολογιστικών συστημάτων, καθώς και στην εκτίμηση της επικινδυνότητά τους (risk assessment analysis). Το κάθε υπολογιστικό σύστημα θεωρείται ως ένα σύνολο από επιμέρους αντικείμενα τόσο λογισμικού και όσο υλισμικού τα οποία αλληλοεπιδρούν μεταξύ τους.

Το κάθε ένα επιμέρους στοιχείο είναι δυνατόν να παρουσιάζει σημεία ευπάθειας, επηρεάζοντας την ευπάθεια του συστήματος ως σύνολο. Ωστόσο, η αθροιστική εκτίμηση της συνολικής επικινδυνότητας είναι μια προσέγγιση που ίσως να μην οδηγεί σε σωστό αποτέλεσμα. Μια εξήγηση για αυτό είναι ότι ένα ιδιαιτέρως ευπαθές τμήμα του συστήματος μπορεί να προκαλέσει την κατάρρευσή του, ακόμα και αν τα υπόλοιπα τμήματά του παραμείνουν ακέραια μετά από μια επίθεση.

Για παράδειγμα, ένα ανεπαρκές τείχος προστασίας (firewall) επιτρέπει την εισχώρηση κακόβουλου λογισμικού στα υπόλοιπα μέρη του υπολογιστικού συστήματος, με δυσμενείς συνέπειες στην ασφάλειά του. Ακόμα και αν τα υπόλοιπα μέρη του συστήματος παρουσιάζουν χαμηλή επικινδυνότητα, η ύπαρξη ενός μέρους με υψηλή επικινδυνότητα, καθιστά όλο το σύστημα υψηλής επικινδυνότητας.

1.2 Βασικά ερευνητικά ερωτήματα

Ββασικό ερώτημα είναι αν υπάρχει δυνατότητα αυτοματοποίησης της εκτίμησης κινδύνου καθώς είναι μια ποσοτική αλλά ταυτόχρονα ποιοτική εργασία υπολογισμού. Ο εντοπισμός των σημαντικότερων ευπαθειών με ΄τρόπο τέτοιο ώστε το αποτέλεσμα που προκύπτει να παρέχει μία σαφής απεικόνιση της επικινδυνότητας και ταυτόχρονα να βοηθά στον εντοπισμό των πιο επικίνδυνών ευπαθειών ώστε μέσα από συνεχής κύκλους περιορισμού κινδύνου (mitigation) να πετυχαίνετε η μείωση της επικινδυνότητας στο ελάχιστον

1.3 Αναγκαιότητα και σπουδαιότητα έρευνας

Μεγάλο μέρος των χρηστών του Διαδικτύου δεν κατέχει τις εξειδικευμένες γνώσεις για τον εντοπισμό και την προστασία από κακόβουλες επιθέσεις, καθιστώντας το ευάλωτο σε κυβερνοεπιθέσεις, με δυσμενείς για αυτούς συνέπειες. Επιπλέον, το τελευταίο διάστημα έχουν παρατηρηθεί πολλές οργανωμένες επιθέσεις σε μεγάλους οργανισμούς σε Ελλάδα και εξωτερικό, οι οποίες μπορούν να προκαλέσουν απώλεια προσωπικών δεδομένων πελατών ή/και χρηστών, καθυστερήσεις και διακοπές στην παροχή υπηρεσιών, αλλά και απώλεια χρημάτων. Χαρακτηριστική είναι η πρόσφατη επίθεση στο λειτουργικό σύστημα των Ελληνικών Ταχυδρομείων, λόγω της οποίας πολλές από τις προσφερόμενες υπηρεσίες δεν μπορούσαν να παρασχεθούν στους πολίτες για μία και πλέον εβδομάδα.

Στον αντίποδα των κυβερνοεπιθέσεων, η κυβερνοασφάλεια ασχολείται με την αντιμετώπιση αυτών των απειλών. Με τον όρο κυβερνοασφάλεια εννοείται η προσέγγιση και οι ενέργειες που σχετίζονται με τις διαδικασίες διαχείρισης κινδύνων ασφαλείας που ακολουθούνται από οργανισμούς και κράτη για την προστασία της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας δεδομένων και περιουσιακών στοιχείων που χρησιμοποιούνται στον κυβερνοχώρο. Η έννοια περιλαμβάνει κατευθυντήριες γραμμές, πολιτικές καθώς και συλλογές διασφαλίσεων, τεχνολογιών, εργαλείων και εκπαίδευσης με σκοπό την παροχή της βέλτιστης προστασίας της κατάστασης του κυβερνοχώρου και των χρηστών του. (Schatz, et al., 2017)

Η συστηματική αντιμετώπιση των κυβερνοεπιθέσεων είναι απαραίτητη για τη διασφάλιση της ομαλής λειτουργίας των υπολογιστικών συστημάτων, από το επίπεδο μεμονωμένων χρηστών, μέχρι το επίπεδο κρατών και μεγάλων οργανισμών. Η ανάγκη για την μείωση αυτών των κινδύνων μπορεί να επιτευχθεί με την χρήση καλών πρακτικών και με τον έγκαιρο εντοπισμό και εκτίμηση επικίνδυνων ευπαθειών.

Η τεχνολογία της πληροφορικής είναι ζωτικής σημασίας για τους δημόσιους και μη οργανισμούς και για όλη την κοινωνία. Το διαδίκτυο παρέχει τεράστιες δυνατότητες αλλά κρύβει και πάρα πολλούς κινδύνους, όπως για παράδειγμα:

- Παιδική Πορνογραφία
- Παραβίαση δικαιωμάτων πνευματικής ιδιοκτησίας (Πειρατεία)
- Πλαστοπροσωπία
- Διακίνηση Ναρκωτικών μέσω του DARK WEB
- Κακόβουλο Λογισμικό (Malware)
- Λυτρισμικό (Ransomware)
- Παράνομη Εισβολή σε Υπολογιστικά Συστήματα (Hacking)
- Εμπόριό Λευκή Σαρκός (Trafficking)
- Revenge Porn

Η αντιμετώπιση αυτών των κινδύνων γίνεται υιοθετώντας μέτρα ασφάλειας κατόπιν προσεχτικής μελέτης και λαμβάνοντας υπόψιν τις απειλές για τα αγαθά , την πιθανότητα εμφάνισής τους (likelihood), τις συνέπειες τους (impact) και τέλος το κόστος των μέτρων ασφαλείας σχετικά με τις συνέπειες από την εμφάνιση κάποιας απειλής.

Ο υπολογισμός της επικινδυνότητας (risk) που προκύπτει μέσα από το likelihood και το impact βοηθά στην υιοθέτηση των απαραίτητων μέτρων ασφαλείας μέσα στους παραπάνω αποδεκτούς παραμέτρους. Η δε αυτοματοποίηση αυτού του υπολογισμού σχεδόν σε πραγματικό χρόνο είναι ισχυρό εργαλείο για ένα διαχειριστή συστήματος για την γρήγορη αντιμετώπιση απειλών.

Για την επίτευξη του σκοπού της διατριβής, προτείνεται και αναπτύσσεται κατάλληλο λογισμικό, το οποίο υπολογίζει με αυτοματοποιημένο τρόπο τα επίπεδα κινδύνου (risk level) των επί μέρους στοιχείων που απαρτίζουν ένα υπολογιστικό σύστημα, αλλά και το συνολικό επίπεδο κινδύνου του υπολογιστικού συστήματος ως ενιαίο σύνολο.

Ένα από τα πλεονεκτήματα του λογισμικού είναι ότι, με την εισαγωγή των κατάλληλων δεδομένων, μπορεί να εφαρμοστεί σε οποιοδήποτε σύστημα. Επίσης, η πρόσβαση στις βάσεις δεδομένων του NIST και του MITRE, από τις οποίες αντλούνται τα στοιχεία που χρειάζονται για την εκτίμηση του επιπέδου κινδύνου, γίνεται με δυναμικό τρόπο και αυτό έχει ως συνέπεια τα αποτελέσματα να είναι πάντα επικαιροποιημένα, εφ' όσον οι εν λόγω βάσεις δεδομένων ενημερώνονται συνεχώς για νέα μοτίβα επιθέσεων και νέες ευπάθειες. Τέλος, το προτεινόμενο

λογισμικό είναι εύχρηστο, αν και μια νέα έκδοση σε περιβάλλον web θα ήταν πιο προσιτή ακόμα και σε μη εξειδικευμένους χρήστες.

Κεφάλαιο 2
Θεωρητικό Υπόβαθρο

Για την καλύτερη κατανόηση της λειτουργίας του προτεινόμενου λογισμικού είναι απαραίτητη η παρουσίαση και επεξήγηση ορισμένων εννοιών που καθορίζουν τα στοιχεία που χρησιμοποιούνται από το προτεινόμενο λογισμικό εκτίμησης επικινδυνότητας.

2.1 Βασικές Απαιτήσεις Ασφαλείας Πληροφοριών

Η ασφάλεια των πληροφοριακών συστημάτων βασίζεται σε τρεις απαιτήσεις.

- Εμπιστευτικότητα (Confidentiality)
  - Η αποκάλυψη πληροφοριών σε μη εξουσιοδοτημένα άτομα απαγορεύεται
  - Η προστασία προσωπικών πληροφοριών θεωρείται πολύ υψηλής προτεραιότητας

- Η αποκάλυψη μπορεί να είναι είτε εκούσια είτε ακούσια

- Ακεραιότητα(Integrity)
  - Απαγορεύεται η μη – εξουσιοδοτημένη τροποποίηση πληροφοριών
  - Οι πληροφορίες πρέπει να διατηρούν την αυθεντική τους κατάσταση
- Διαθεσιμότητα (Availability)
  - Η συνεχής και αδιάλειπτη δυνατότητα χρήσης των δεδομένων και υπολογιστικών υποδομών

Αυτές οι απαιτήσεις είναι στο επίκεντρο της κυβερνοασφάλειας.

2.2 Πρότυπα Ασφάλειας

Ο μη κερδοσκοπικός οργανισμός MITRE, ο οποίος εδρεύει στις ΗΠΑ και δραστηριοποιείται μεταξύ άλλων και με θέματα ασφάλειας Διαδικτύου, έχει ασχοληθεί με την ταξινόμηση των κυβερνοεπιθέσεων. Στη σελίδα του στο Διαδίκτυο υπάρχει δημοσιευμένη μια προσπάθεια ταξινόμησης των βασικών στοιχείων που χρησιμοποιούνται για την εκτίμηση της ασφάλειας. Τα βασικά στοιχεία που ορίζονται από το MITRE (MITRE6, 2008) αναφέρονται στην παρούσα ενότητα.

2.2.1 Common Platform Enumeration - CPE

Το CPE (Common Platform Enumeration) [03, 04] (MITRE4) (MITRE1), αποτελεί τον αναγνωριστικό ορισμό που χαρακτηρίζει κάθε στοιχείο μίας υπολογιστικής ή δικτυακής υποδομής, για παράδειγμα όπως εφαρμογές, λειτουργικά συστήματα, λογισμικό και υλισμικό (hardware) που συμπεριλαμβάνει και συστήματα βιομηχανικού ελέγχου, όπως είναι ο εποπτικός έλεγχος και η συλλογή δεδομένων (SCADA). Το λογικό κατασκεύασμα ενός CPE ονομάζεται "Well-Formed CPE Name (WFN) και μπορούμε να το δούμε στο παρακάτω σχεδιάγραμμα. Το σχήμα των cpe δεν έχει όμως την δυνατότητα να περιγράψει και να αναγνωρίσει πιο συγκεκριμένα χαρακτηριστικά ή και λεπτομέρειες, όπως για παράδειγμα εφαρμογών ανά σειριακό αριθμό και συγκεκριμένες άδειες αυτών (MITRE1).

cpe-name:part:vendor:product:version:update:edition:language:sw_edition:target_sw:target_hw:other
cpe:2.3:a:in:out:1.9:sp3:NA:EN:home:windows:x64:*

2.2.2 Common Vulnerability and Exposures - CVE

Το CVE (Common Vulnerabilities and Exposures) (MITRE2) (NIST) αναφέρεται σε μια γνωστή αδυναμία που βρίσκεται σε λογισμικό ή υλισμικό και μπορεί να γίνει εκμεταλλεύσιμή με σοβαρό αντίκτυπο πάνω στα χαρακτηριστικά της εμπιστευτικότητας, ακεραιότητας ή διαθεσιμότητας. Κάθε CVE εμπεριέχει ένα μοναδικό χαρακτηριστικό αριθμό, μια περιγραφή, μια δημόσια αναφορά, μια αναφορά στο λογισμικό ή υλισμικο που επηρεάζει (κάνοντας χρήση του CPE), μια αναφορά σε μία αδυναμία (όπως περιγράφεται παρακάτω) και στην σοβαρότητά της.

2.2.3 Common Weakness Enumeration - CWE

Τα CWE (Common Weakness Enumeration) (MITRE3) αναπαριστούν ελαττώματα στην αρχιτεκτονική, σχεδιασμό και κώδικα λογισμικού που μπορεί να οδηγήσει σε συγκεκριμένο CVE σε μια εφαρμογή. Για παράδειγμα το CVE-2019-1675 "Cisco Aironet Active Sensor Static Credentials Vulnerability" που βρίσκεται στο Cisco Aironet Active Sensor είναι μια εκδήλωση του CWE-798 "Use of hard-coded Credentials". Επομένως μπορεί να διατυπωθεί το συμπέρασμα ότι τα CVE είναι πράγματα που γνωρίζουμε και αντιλαμβανόμαστε ενώ τα CWE είναι πράγματα που αντιλαμβανόμαστε αλλά δεν γνωρίζουμε και θα μπορούσαν να ορισθούν ως γενικότερου επιπέδου αδυναμίες/ευπάθειες.

2.2.4 Common Attack Pattern Enumeration and Classification - CAPEC

Το CAPEC (Common Attack Pattern Enumeration and Classification) (MITRE4) είναι ένα εμπεριστατωμένο λεξικό και μια κατηγοριοποίηση ταξινόμησης όλων των γνωστών απειλών ασφαλείας . Κάθε CAPEC περιγράφει τα κοινά χαρακτηριστικά μιας κυβερνοαπειλής (γνωστό και ως μοτίβο επίθεσης), βοηθώντας να εξηγηθεί πως εφαρμογές και άλλες κυβερνο-ενεργοποιημένες δυνατότητες μπορούν να γίνουν θύμα επίθεσης. Ταυτόχρονα οι CAPEC μπορούν να παρέχουν την πιθανότητα και την επίδραση του μοτίβου επίθεσης. Οι CAPEC χρησιμοποιούν μια ποιοτική προσέγγιση που βαθμονομεί την πιθανότητα και την επίδρασή σε μια κλίμακα πέντε επιπέδων που κυμαίνεται από πολύ χαμηλή έως πολύ υψηλή. Τέλος κάθε CAPEC καταγράφει την αδυναμία που το μοτίβο επίθεσης μπορεί να εκμεταλλευθεί.

2.2.5 Common Vulnerability Scoring System - CVSS

Το CVSS (Common Vulnerability Scoring System) (FIRST) είναι ένα σύστημα βαθμονόμησης κοινών ευπαθειών που αναπτύχθηκε από την FIRST (Forum of Incident Response and Security Teams) αποτελεί ένα πλαίσιο για την επικοινωνία των χαρακτηριστικών και της σοβαρότητας επιπτώσεων των CVE. Το CVSS αναπαριστά την σοβαρότητα επιπτώσεων των CVE χρησιμοποιώντας ένα συνοπτικό σύστημα βαθμολόγησης και ακολουθώντας 3 συστήματα μέτρησης, το βασικό, το χρονικό και το περιβαλλοντικό.

Το βασικό σύστημα μέτρησης αντανακλά τα εγγενή χαρακτηριστικά της CVE. Αυτά τα χαρακτηριστικά είναι σταθερά και κατηγοριοποιούνται στις ακόλουθες δυο κατηγορίες.

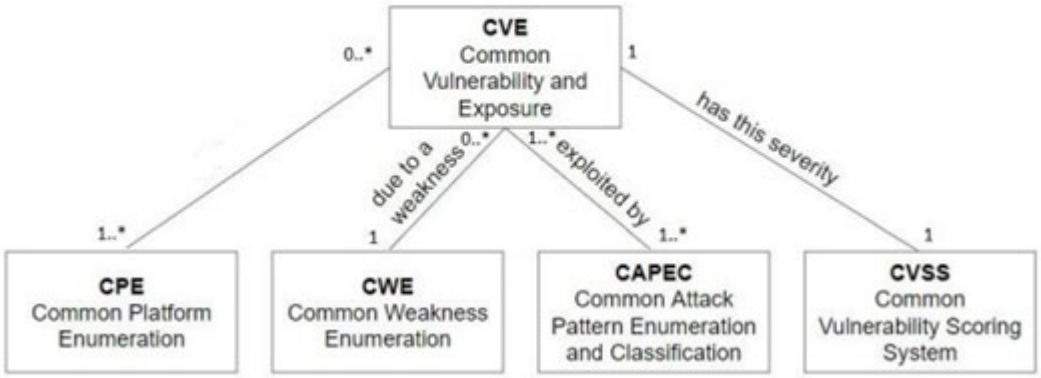
a. Η κατηγορία ικανότητας εκμετάλλευσης ευπάθειας που αντικατοπτρίζει την ευκολία και τα τεχνικά μέσα από τα οποία μια ευπάθεια μπορεί να γίνει προϊόν εκμετάλλευσης.

b. Η κατηγορία σοβαρότητας επιπτώσεων που αναπαριστά την επίπτωση της CVE στην Εμπιστευτικότητα , Ακεραιότητα και την Διαθεσιμότητα που είναι τρία χαρακτηριστικά της ασφάλειας πληροφοριών . Τα επίπεδα της κατηγορίας μπορεί να είναι είτε καθόλου , Χαμηλή , Μέτρια, ή Υψηλή ανάλογα με πόσο μεγάλη είναι η απώλεια των CIA χαρακτηριστικών

Το χρονικό σύστημα μέτρησης προσαρμόζει το βασικό που βασίζεται σε παράγοντες που μπορούν να αλλάξουν με τον χρόνο όπως οι διαθεσιμότητα της ικανότητας εκμετάλλευσης ενός CVE.

Το περιβαλλοντικό σύστημα μέτρησης μπορεί να διαμορφώσει το χρονικό και το βασικό σε ένα συγκεκριμένο περιβάλλον σχετικά με συγκεκριμένους παράγοντες όπως αντίμετρα που μπορεί να υπάρχουν σε ένα περιβάλλον . (Dimitriadis, 2022)

Στην εικόνα 1 (Dimitriadis, et al., 2020) φαίνονται πιο παραστατικά οι σχέσεις μεταξύ των προτύπων ασφαλείας και του CVSS.



Εικόνα 1: Διάγραμμα απεικόνισης σχέσεων μεταξύ CVE, CPE, CWE, CAPEC και CVSS [11]

Ο πίνακας 1 μεταφράζει την βαθμονόμηση αριθμών με επίπεδα.

CVSS Score	Qualitative Rating
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical

Πίνακας 1: Αντιστοιχία βαθμονόμησης CVSS

2.3 Εκτίμηση Επικινδυνότητας

Επικινδυνότητα είναι ένα μέτρο του εύρους κινδύνου από τον οποίο απειλείται μια οντότητα, όσον αφορά την έλευση ενός γεγονότος με αρνητικές για την οντότητα συνέπειες. Η επικινδυνότητα θα

πρέπει να διατηρείται σε αποδεκτά επίπεδα ώστε να διατηρείται το επιθυμητό επίπεδο ασφαλείας. Αυτό ικανοποιείται μέσω της διαχείρισης επικινδυνότητας, η οποία αναγνωρίζει , εκτιμά και διατηρεί την επικινδυνότητα στα επιθυμητά ή/και ανεκτά επίπεδα.

Σε γενικές γραμμές η διαχείριση επικινδυνότητας αποτελείται από τέσσερα συστατικά:

- Πλαισίωση της επικινδυνότητας
- Πρόσβαση στην επικινδυνότητα
- Αντίδραση στην επικινδυνότητα
- Παρακολούθηση και έλεγχος επικινδυνότητας

Ο στόχος της εκτίμησης επικινδυνότητας είναι να υπολογισθεί η έκταση του πόσο μια οντότητα μπορεί να απειληθεί από συγκεκριμένες συνθήκες ή γεγονότα και πρωταρχικά να υπολογισθεί η επικινδυνότητα αυτού. Σύμφωνα με το NIST (Joint Task Force Transformation Initiative, 2012) η εκτίμηση και η διαχείριση της επικινδυνότητας διεξάγεται σε τρία επίπεδα (Tier):

- Οργανισμού (Tier 1)
- Διεργασιών (Tier 2)
- Πληροφοριακού Συστήματος (Tier 3)

Σύμφωνα με το πρότυπο του NIST SP800-30 (Joint Task Force Transformation Initiative, 2012), η επικινδυνότητα εξαρτάται από την πιθανότητα εμφάνισης μιας απειλής σε σχέση με κάποια ευπάθεια αλλά και από τις συνέπειες αν αυτή η απειλή υλοποιηθεί, ενώ με τις οδηγίες του NIST (National Institute of Standards and Technology) (Joint Task Force Transformation Initiative, 2012), μια μεθοδολογία εκτίμησης επικινδυνότητας πρέπει να περιλαμβάνει τα παρακάτω:

a) Ένα μοντέλο κινδύνου, στο οποίο καθορίζονται οι παράγοντες επικινδυνότητας και οι συσχετίσεις τους που χρησιμοποιούνται για τον υπολογισμό των επιπέδων . Παράγοντες επικινδυνότητας μπορεί να είναι απειλές, ευπάθειες, η επίπτωση και η πιθανότητα επέλευσης μιας απειλής.

b) Μια μεθοδολογία ανάλυσης, στην οποία καθορίζεται πώς προσδιορίζονται και αναλύονται. Υπάρχουν τρεις βασικές προσεγγίσεις (asset/impact oriented, threat oriented, vulnerability oriented). Ανεξάρτητα από το ποια προσέγγιση επιλέγεται,

θεμελιώδες στοιχείο είναι να οριστούν τα στοιχεία του συστήματος που χρειάζονται προστασία.

c) Μια μεθοδολογία εκτίμησης επικινδυνότητας, η οποία καθορίζει την έκταση της και μπορεί να είναι ποσοτική, ποιοτική ή ημι-ποιοτική.

d) Μια διαδικασία για την εκτίμηση επικινδυνότητας, η οποία περιλαμβάνει τις ενέργειες που απαιτούνται για την προετοιμασία της ανάλυσης της, την ίδια την εκτίμηση της, καθώς και την διάδοση των αποτελεσμάτων και την συνεχή συντήρηση της διαδικασίας.

Κεφάλαιο 3
Μεθοδολογίες Αυτόματης Εκτίμησης Επικινδυνότητας

Το ζήτημα της εκτίμησης επικινδυνότητας των υπολογιστικών συστημάτων αναπτύσσεται συνεχώς στη βιβλιογραφία, καθώς είναι ένα πρωταρχικό ζήτημα για την ασφάλειά τους. Κάποια από τα σχετικά με την παρούσα μεταπτυχιακή διατριβή ζητήματα παρουσιάζονται σε αυτό το κεφάλαιο.

3.1. Μεθοδολογία ROPE

Μια άλλη μεθοδολογία που έχει αναπτυχθεί για την διαχείριση επιχειρηματικών διεργασιών με επίγνωση ρίσκου που χρησιμοποιείται στο εργαλείο ROPE (Risk Oriented Process Evaluation), (Tjoa, et al., 4-7 March 2008) (Jakoubi, et al., 7-9 June 2007). Η βασική ιδέα αυτής της προσέγγισης είναι η μοντελοποιήση των απειλών και της ανίχνευσης με προσανατολισμό προς τις διεργασίες (process-oriented modeling of threats and detection), αντίμετρα και μέτρα ανάκτησης για κάθε απειλή. Η μεθοδολογία αποτελείται από 5 βήματα και ένα μοντέλο 3 επιπέδων ώστε να αναγνωρίσουν απειλές ,την διαχείριση και την στρατηγική ανάκτησης τους. Για να γίνει αυτό το πρέπει το επίπεδο μοντελοποιήσης ξεκινά με την αναγνώριση σημαντικών επιχειρηματικών διεργασιών και των δραστηριοτήτων τους για σκοπούς ιεράρχησης σχετικά με τον εντοπισμό απειλών. Στο επόμενο επίπεδο η ανακάλυψη υπολογιστικών συστημάτων (IS) των δραστηριοτήτων των επιχειρηματικών διεργασιών λαμβάνει χώρα. Τέλος στο επόμενο επίπεδο οι απειλές και η διαχείριση και στρατηγική ανάκτησης. Αν και η μεθοδολογία του ROPE αναγνωρίζει απειλές χρησιμοποιώντας επιχειρηματικές διεργασίες , δεν χρησιμοποιεί τα κοινά πρότυπα ασφαλείας για να περιγράψει τα υπολογιστικά συστήματα με ένα δομημένο τρόπο . Τέλος το βασικό συστατικό που δεν έχει το ROPE είναι ένα μοντέλο εκτίμησης ρίσκου για να

υπολογίζεται το ρίσκο. Χωρίς αυτό το ρίσκο δεν μπορεί να υπολογισθεί ούτε να αυτοματοποιηθεί ο υπολογισμός του.

3.2. Μεθοδολογία για Εκτίμηση Επικινδυνότητας στον Τραπεζικό Τομέα

Η μεθοδολογία εκτιμά την επικινδυνότητα ενός τραπεζικού υπολογιστικού συστήματος χρησιμοποιώντας τις απαιτήσεις ασφαλείας του τραπεζικού τομέα (SSRB) που έχει προέλθει από τις βέλτιστες πρακτικές αυτού και από τα CAPEC (Rongrat, et al., 2017). Εκτιμάται πως το ρίσκο ενός τραπεζικού υπολογιστικού συστήματος είναι το άθροισμα των τιμών της επικινδυνότητας των υπολειπόμενων απαιτήσεων ασφαλείας του εξεταζόμενου υπολογιστικού συστήματος. Για τον υπολογισμό του ρίσκου κάθε απαίτησης αυτή αντιστοιχίζεται με CAPEC και στην συνέχεια αυτό προκύπτει ως το άθροισμα των τιμών όλων των σχετικών CAPEC. Για αναγνωρισθούν οι υπολειπόμενες απαιτήσεις ενός τραπεζικού υπολογιστικού συστήματος, τεχνικές ομοιότητας κειμένου χρησιμοποιούνται για να αντιστοιχισθούν τα SSRBs με το κείμενο των απαιτήσεων ασφαλείας. Όποιο SSRB δεν αντιστοιχήθηκε θεωρείτε υπολειπόμενη απαίτηση και επομένως το ρίσκο του χρησιμοποιείται για να υπολογισθεί η συνολική τιμή. Ενώ ως μέθοδος παρέχει ποσοτικά ρίσκα και η εκτίμηση μπορεί να γίνει γρήγορα και να είναι βασισμένη στα στάνταρτ του τραπεζικού τομέα , δεν βασίζεται σε ένα συγκεκριμένο μοντέλο ρίσκου (επίπτωση, απειλή πιθανότητα, ευπάθειες) τέτοιο ώστε να αντιστοιχεί σε ρίσκο βασισμένο σε υπάρχουσες ευπάθειες του υπολογιστικού συστήματος και των απειλών τους. Εκτός αυτού το μοντέλο είναι περιορισμένο να χρησιμοποιεί μόνο CAPEC.

3.3. Μεθοδολογία Πρόβλεψης Επικινδυνότητας μέσω ανάλυσης ευπαθειών πλατφόρμας

Μία μεθοδολογία πρόβλεψης κινδύνου που βασίζεται σε ιστορικά δεδομένα σχετικά με τα CVE και το CVSS εισήχθη με τα ακόλουθα βήματα:

- περιγραφή πλατφόρμας με χρήση CPE.

- αναγνώριση προηγούμενων ευπαθειών της πλατφόρμας με ανάκτηση και εξέταση CVE (κάθε CPE μπορεί να σχετίζεται με πολλαπλά CVE).
- Πρόβλεψη ευπάθειας με εφαρμογή δοκιμών Kolmogorov-Smirnov για να αποκαλυφθεί εάν ο αριθμός των μελλοντικών περιστατικών ευπάθειας θα είναι λιγότερα ή περισσότερα.
- Πρόβλεψη επικινδυνότητας με τη χρήση ενός μοντέλου της που βασίζεται σε τοπολογία Bayesian Belief Network (BBN), θεωρία του Von Mises και προβλεπόμενα μελλοντικά CVE και το CVSS τους.

Η μεθοδολογία χρησιμοποιεί ιστορικά δεδομένα για να προβλέψει την τάση του αριθμού ευπαθειών στο μέλλον. Η πρόβλεψη είναι μόνο από την πλευρά της ευπάθειας, χωρίς να εξετάζονται οι απειλές (π.χ. CAPEC). Ωστόσο, οι απειλές πρέπει να προσδιορίζονται και να λαμβάνονται υπόψη κατά την εκτίμηση της επικινδυνότητας. Επιπλέον, οι τιμές επιπτώσεων CVSS στα χαρακτηριστικά CIA μπορεί να είναι υψηλότερες σε σύγκριση με τον αντίκτυπο που το CAPEC μπορεί να προκαλέσει. Χωρίς να λαμβάνεται υπόψη το CAPEC, η παραπάνω μεθοδολογία μπορεί υπερεκτιμά τον κίνδυνο.

3.4. Μεθοδολογία ARES

Η μεθοδολογία του ARES (Joint Task Force Transformation Initiative, 2012)(Automated Risk Estimation in Smart Sensor Environments) έχει αναπτυχθεί με σκοπό την αυτοματοποιημένη εκτίμηση επικινδυνότητας σε υπολογιστικά συστήματα και μέρη αυτών. Ειδικότερα μπορεί να βρει εφαρμογή σε διάφορες ηλεκτρονικές συσκευές, οι οποίες εφόσον είναι συνδεδεμένες στο Διαδίκτυο, εμφανίζουν σημεία με ευπάθειες. Τα σημεία ευπάθειας είναι δυνατόν να αποτελέσουν την κερκόπορτα, μέσω της οποίας μπορεί να εκτελεσθεί με επιτυχία μια κυβερνοεπίθεση.

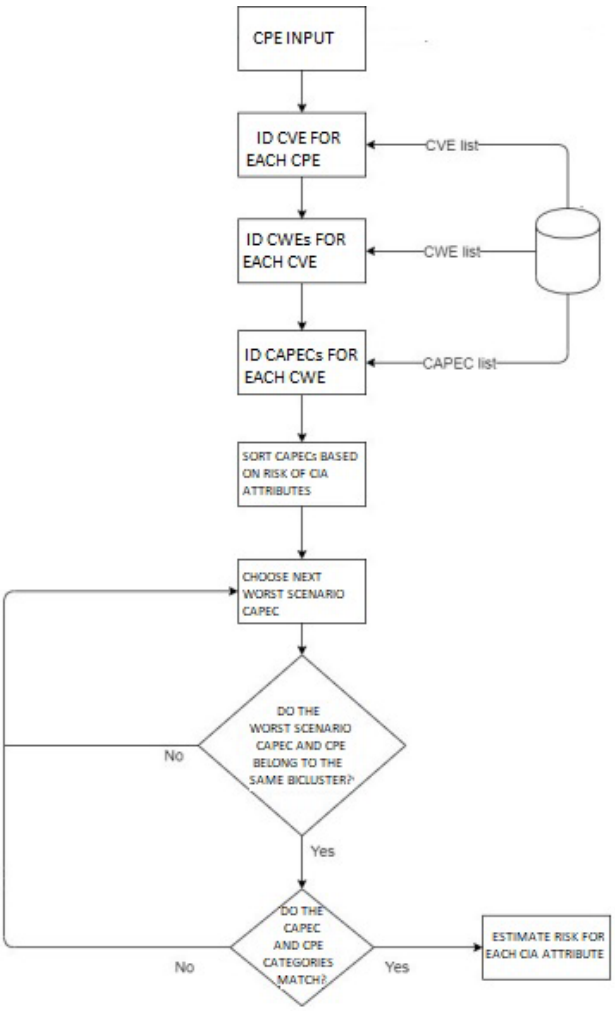
Η μεθοδολογία ARES ακολουθεί μια ποιοτική προσέγγιση στο ζήτημα της εκτίμησης κινδύνου του κάθε μελετώμενου CPE. Η εκτίμηση κινδύνου βασίζεται σε μοντέλο κινδύνου το οποίο συνεκτιμά την πιθανότητα να συμβεί μια απειλή (προερχόμενη από κάποιο CAPEC) και την πιθανή επίπτωση

αυτής της απειλής σε κάθε χαρακτηριστικό CIA. Τα αποτελέσματα είναι ποιοτικά και εκφράζονται με τις παρακάτω τιμές: None, Low, Medium, High και Very High όπως φαίνεται στον πίνακα 3.1.

Impact / Likelihood	None	Very Low	Low	Medium	High	Very high
Low	None	Low	Low	Medium	Medium	High
Medium	None	Low	Medium	Medium	High	High
High	None	Low	Medium	High	High	Very High

Πίνακας 2: Εκτίμηση επικινδυνότητας ως συνάρτηση επίπτωσης (impact) και πιθανότητας εμφάνισης (likelihood), σύμφωνα με τη μεθοδολογία ARES [11]

Καθώς το κάθε CPE μπορεί να έχει πολλαπλά CVE και το καθένα από αυτά μπορεί να είναι ευάλωτο σε διάφορα CAPEC, είναι πιθανό η προσέγγιση της εκτίμησης κινδύνου να έχει ως αποτέλεσμα ένα δυσανάλογα υψηλό επίπεδο κινδύνου. Για το λόγο αυτό, το εργαλείο ARES θεωρεί ότι ο συνολικός κίνδυνος για κάθε CPE είναι ένα άνω όριο που αντιστοιχεί στην χειρότερη περίπτωση και όχι το άθροισμα των επιμέρους κινδύνων. Στην εικόνα 3.1 οπτικοποιείται ο μεθοδολογία τους ARES.



Εικόνα 2: Διάγραμμα ροής του μεθοδολογίας του ARES (Dimitriadis, et al., 2020)

3.5. Σύγκριση μεθοδολογιών εκτίμησης επικινδυνότητας

Στον παρακάτω πίνακα παρουσιάζετε ένας συγκριτικός πίνακας με ποιοτικά χαρακτηριστικά εξαιτίας των οποίων γίνεται η επιλογή της μεθοδολογίας

Μεθοδολογία	**Rope**	**ARES**	**Εκτίμησης στον Τραπεζικό Τομέα**	**Εκτίμηση μέσω ανάλυσης ευπαθειών**
Βαθμός Αυτοματοποίησης	**Χαμηλός**	**Υψηλός**	**Μέτριος**	**Χαμηλός**
Εκτίμηση με αξιοποίηση γνωστών β.δ γνώσης	**Μοντελοποίηση απειλών & ανίχνευσης**	**Χρήση CVE, CWE, CAPEC**	**Χρήση Προτύπου**	**Χρήση CVE**
Υπολογισμός στη βάση μετρήσεων	**Δεν υπάρχει**	**Επίδραση & Πιθανότητα Εμφάνισης**	**Επίδραση & Πιθανότητα Εμφάνισης**	**Πιθανότητα εμφάνισης**

Πίνακας 3: Βαθμονόμηση αυτοματοποίησης

Όπως γίνεται αντιληπτό η μεθοδολογία με τον υψηλότερο βαθμό αυτοματοποίησης είναι η μεθοδολογία ARES και για αυτό τον λόγο γίνεται τελικά και η επιλογή της καθώς είναι αυτή που ταιριάζει ώστε να χρησιμοποιηθεί σαν βάση για την υλοποίηση του λογισμικού

Κεφάλαιο 4
Ανάπτυξη Λογισμικού Automatic Risk Assessment Tool

Στην παρούσα μεταπτυχιακή διατριβή αναπτύσσεται το λογισμικό Automatic Risk Assessment Tool (ARAT), το οποίο έχει ως αντικείμενο την αυτόματη εκτίμηση επικινδυνότητας υπολογιστικών συστημάτων. Το λογισμικό ARAT εφαρμόζεται σε υπολογιστικά συστήματα ως σύνολο και υπολογίζει τόσο τα επιμέρους όσο και τα συνολικά επίπεδα επικινδυνότητας του μελετώμενου συστήματος. Τα χαρακτηριστικά του λογισμικού παρουσιάζονται στο παρόν κεφάλαιο.

4.1 Εισαγωγή

Η ανάπτυξη του λογισμικού γίνεται σε γλώσσα προγραμματισμού Python έκδοση 3.0. Η python επιλέχθηκε γιατί παρουσιάζει κάποια χαρακτηριστικά κατάλληλα για την υλοποίηση του A.R.A.T v1. Κατ' αρχάς, η Python ως γλώσσα προγραμματισμού είναι εύχρηστη όσον αφορά την εισαγωγή αρχείων (πχ. Json, cvs, xml). Επίσης είναι γλώσσα τύπου script, ενώ δεν απαιτείται η διενέργεια compiling. Έχει πληθώρα έτοιμων βιβλιοθηκών, έχει υψηλή υποστήριξη και είναι εύκολη στην εκμάθηση. Τέλος είναι γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως στην Τεχνητή Νοημοσύνη (A.I.). Στα μειονεκτήματά της συγκαταλέγονται η χαμηλή ταχύτητα εκτέλεσής της και η υψηλή χρήση μνήμης. Στο συγκεκριμένο όμως λογισμικό (A.R.A.T.) οι υπολογιστικές ανάγκες είναι σχετικά μικρές, οπότε η python κάλυψε με επάρκεια τις προγραμματιστικές απαιτήσεις.

4.2 Διάρθρωση του ARES σε διαδικασία έξι βημάτων

1. Εισαγωγή των CPE, αναγνώριση των αντίστοιχων CVE που με την σειρά τους αναγνωρίζουν αντίστοιχα CWE οπότε και προκύπτουν τα αντίστοιχα CAPEC.

2. Ταξινόμηση των CAPEC σύμφωνα με την επικινδυνότητα των CIA χαρακτηριστικών

3. Επιλογή του CAPEC σύμφωνα με το worst-case scenario

4. Έλεγχος αν τα προηγουμένως επιλεγμένα CAPEC και CPE ανήκουν στο ίδιο cluster

5. Αν η προηγούμενη απάντηση είναι θετική έλεγχος αν τα CAPEC και CPE ανήκουν στην ίδια κατηγορία. Αν αρνητική τότε επιστροφή στο βήμα 3

6. Αν η προηγούμενη απάντηση είναι θετική, γίνεται εκτίμηση της επικινδυνότητάς του κάθε CIA χαρακτηριστικού. Αν αρνητική επιστροφή στο βήμα 3

Για την αναγνώριση των παραγόντων κινδύνου του μοντέλου που είναι βασικά η πιθανότητα εμφάνισης και η επίπτωση, το ARES χρησιμοποιεί μια προσέγγιση προσανατολισμένη στις ευπάθειες . Κάθε CPE μπορεί να έχει παραπάνω από μια ευπάθειες , δηλαδή CVE, που με την σειρά τους περιέχουν βαθμολογία του CVSS που μπορεί να συσχετισθεί με μία αδυναμία (CWE) και μπορεί να γίνει αντικείμενο εκμετάλλευσης από ένα CAPEC.

Το ARES υπολογίζει την επικινδυνότητα για κάθε χαρακτηριστικό CIA ενός CPE βασιζόμενο στην επίπτωση (impact) και την πιθανότητα εμφάνισης (likelihood) του σχετιζόμενου CAPEC . Όμως καθώς ένα CPE μπορεί να έχει πολλαπλά CVE, το καθένα από τα οποία μπορούν να γίνουν αντικείμενα εκμετάλλευσης από πολλαπλά CAPEC κάτι που σημαίνει ότι το κάθε χαρακτηριστικό CIA μπορεί να αντιμετωπίζει πολλούς κίνδυνους. Ακολουθώντας το ARES η επικινδυνότητα υπολογίζεται σύμφωνα με το CAPEC που έχει την πιο υψηλή τιμή επικινδυνότητας για το χαρακτηριστικό CIA. Αυτό το worst-scenario CAPEC είναι αυτό με τις υψηλότερες τιμές επίπτωσης και πιθανότητας εμφάνισης για όλα τα χαρακτηριστικά CIA.

Η πιθανότητα εμφάνισης του CAPEC δίνεται από το ίδιο και είναι η ίδια για κάθε ένα από τα χαρακτηριστικά CIA. Σε περίπτωση που το CAPEC δεν παρέχει την πιθανότητα εμφάνισης, το ARES χρησιμοποιεί την πιθανότητα εμφάνισης του γονικού CAPEC . Η τιμή της επίπτωσης όμως μπορεί να είναι διαφορετική για κάθε CIA χαρακτηριστικό. Για να υπολογισθεί η τιμή της επίπτωσης το ARES συγκρίνει την τιμή που παρέχεται από το CAPEC με την τιμή που παρέχεται από το CVSS. Επιλέγεται η χαμηλότερη των δύο. Αν το CAPEC δεν παρέχει τιμή χρησιμοποιείται αυτή του γονικού CAPEC . Αν το CVSS δεν παρέχει τιμή, το ARES χρησιμοποιεί αυτή του CAPEC. Το αποτέλεσμα προκύπτει με τον τρόπο που απεικονίζεται στον πίνακα 3.

CVSS \ CAPEC	**-**	**Very Low**	**Low**	**Medium**	**High**	**Very high**
-	[Parent CAPEC]	Very Low	Low	Medium	High	Very High
None	None	None	None	None	None	None
Low	Low	Very Low	Low	Low	Low	Low
High	High	Very Low	Low	Medium	High	High

Πίνακας 4: Συντελεστής επικινδυνότητας για ένα CIA χαρακτηριστικό, ως συνάρτηση CAPEC και CVSS (MITRE4) (FIRST)

4.3 Κατηγοριοποίηση

Ένα ζήτημα που ανακύπτει κατά την εφαρμογή του ARES αλλά και άλλων μοντέλων, μεταξύ των οποίων και αυτό που αναπτύσσεται στην παρούσα εργασία, είναι το ζήτημα της κατηγοριοποίησης. Από την επεξεργασία των λιστών του MITRE, προκύπτει ότι εντοπίζονται πολλοί συσχετισμοί CPE με CVE και CAPEC. Ωστόσο, η σύνδεση που προκύπτει δεν είναι πάντα

λογική. Για παράδειγμα, από το ARES μπορεί να προκύψει σύνδεση ενός CPE που ουσιαστικά είναι software με ένα CAPEC το οποίο αφορά μόνο hardware. Ακόμα και αν από την επεξεργασία των λιστών προκύψει τέτοια σύνδεση, το εν λόγω CAPEC θα πρέπει να διαγραφεί από τις ευπάθειες του CPE, ως μη εφικτό. Η ταξινόμηση των διάφορων στοιχείων είναι σημαντική, ώστε η αποσύνδεση και κατ' επέκταση ο αποκλεισμός των μη σχετιζόμενων CPE και CAPEC να γίνεται με συστηματικό και όχι με εμπειρικό τρόπο.

Η χρήση συστήματος κατηγοριοποίησης δεν είναι απλά η κατηγοριοποίηση γνωστών ονομάτων, αλλά αναπαριστά την θεωρία σύμφωνα με την οποία αυτά τα ονόματα οργανώνονται καθορίζοντας πως διακρίνονται μεταξύ τους.

Πλεονεκτήματα εφαρμογής κατηγοριοποίησης (Forward, et al., 2008).

- Καλύτερη έρευνα λογισμικού. Μια κατηγοριοποίηση θα πρέπει να παρέχει ένα γενικό πλαίσιο για εμπειρικά περιεχόμενα στην ανάπτυξη λογισμικού καθώς και να διευκολύνει την εξερεύνηση της εφαρμοσιμότητας αυτών των αποτελεσμάτων. Η κατηγοριοποίηση μπορεί να χρησιμοποιηθεί για να προταθούν οι τύποι του λογισμικού που τα αποτελέσματα μπορούν να ελεγχθούν. Για παράδειγμα μια έρευνα σε χρήστες λογισμικού μπορεί να υποδιαιρεθεί σε συμμετέχοντες που εργάζονται με λογισμικό που κυριαρχούν τα δεδομένα σε σχέση με αυτούς που το λογισμικό είναι βασισμένο σε υπολογισμούς. Από αυτή την υποδιαίρεση ο ερευνητής θα μπορούσε να επιδείξει ότι μια συγκεκριμένη διεργασία του λογισμικού είναι πράγματι κατάλληλη και για τις δυο περιπτώσεις εφαρμογών ή μόνο στην μία από αυτές.

- Αντικείμενα είναι πιο εύκολα διαθέσιμα. Αντικείμενα όπως βιβλιοθήκες (libraries), πρόσθετα (plugins), μοτίβα λογισμικού και αλγόριθμοί γίνονται πιο εύκολα προσβάσιμα για επανάχρηση αν αντιστοιχισθούν σε κατηγορίες μέσα στην κατηγοριοποίηση λογισμικού.

- Αυξημένη χρήση μοντέλων και πλαισίων αναφοράς. Αν μοντέλα και πλαίσια αναφοράς αντιστοιχίζονται σε μια κατηγοριοποίηση ειδών λογισμικού, οι προγραμματιστές έχουν ένα καλύτερο σημείο εκκίνησης κατά την διαδικασία δόμησης νέου λογισμικού σε ένα συγκεκριμένο τομέα. Παρόμοια με τα επαναχρησιμοποιούμενα αντικείμενα που αναφέρθηκαν προηγουμένως μπορούν επίσης να αναγνωριστούν ελλείψεις σε πλαίσια και μοντέλα αναφοράς.

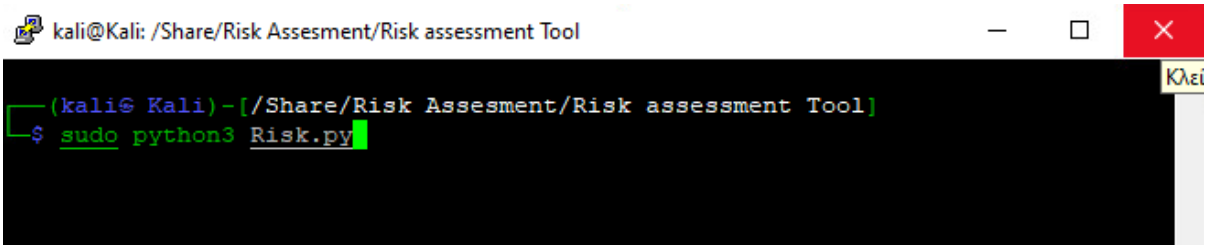
Η χρήση του συστήματος κατηγοριοποίησης σε συνδυασμό με την μεθοδολογία του ARES αποτελεί έναν ακόμη τρόπο διαχωρισμού των σχετικών CAPEC και CPE με αυτά των μη σχετικών.

4.4. Διαδικασία Υπολογισμών

Ακολουθεί περιγραφή της διαδικασίας υπολογισμού που εκτελεί το προτεινόμενο λογισμικό Automatic Risk Assessment Tool v1.0 (A.R.A.T. , με σκοπό να υλοποιήσει την αυτοματοποιημένη εκτίμηση του επίπεδου επικινδυνότητας του συνολικού υπολογιστικού συστήματος.

4.4.1 Εισαγωγή Δεδομένων

Ως μέθοδος εισαγωγής των δεδομένων χρησιμοποιείται το command-line interface. Η υποδομή που χρησιμοποιείται είναι ένα Virtual Machine (VM) που έχει εγκατεστημένη την τελευταία έκδοση του Kali Linux. Τα δεδομένα εισάγονται από την γραμμή εντολή σε περιβάλλον linux. Σε μελλοντική ανάπτυξη του λογισμικού είναι δυνατόν να υλοποιηθεί ως δικτυακή εφαρμογή εγκατεστημένη σε web server, στην οποία ο κώδικας της python θα εκτελείται μέσα από html κέλυφος. Ο τρόπος εισαγωγής των δεδομένων παρουσιάζεται στην Εικόνα 3.



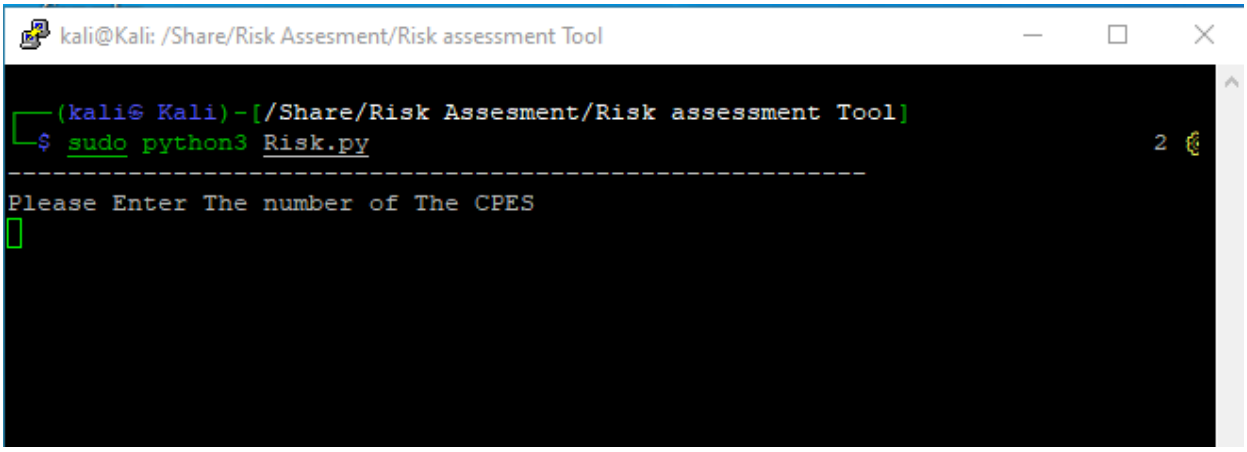
Εικόνα 3: Εισαγωγή δεδομένων στην command-line

Στη συνέχεια εκτελείται μέσω python ο κώδικας που βρίσκεται στο αρχείο **Risk.py** . Το αρχείο που περιέχει τον κώδικα συνδέεται με διάφορα άλλα αρχεία που περιέχουν κατάλληλα δεδομένα απαραίτητα για την εκτέλεση του αλγορίθμου εκτίμησης. Τα αρχεία αυτά είναι:

- το αρχείο **capec2cpe.xml**, το οποίο είναι αρχείο απαραίτητο για το biclustering που θα παρουσιασθεί σε επόμενη ενότητα
- τα αρχεία **cpes.csv** και **capecs.csv**, τα οποία περιέχουν δεδομένα απαραίτητα για την κατηγοριοποίηση

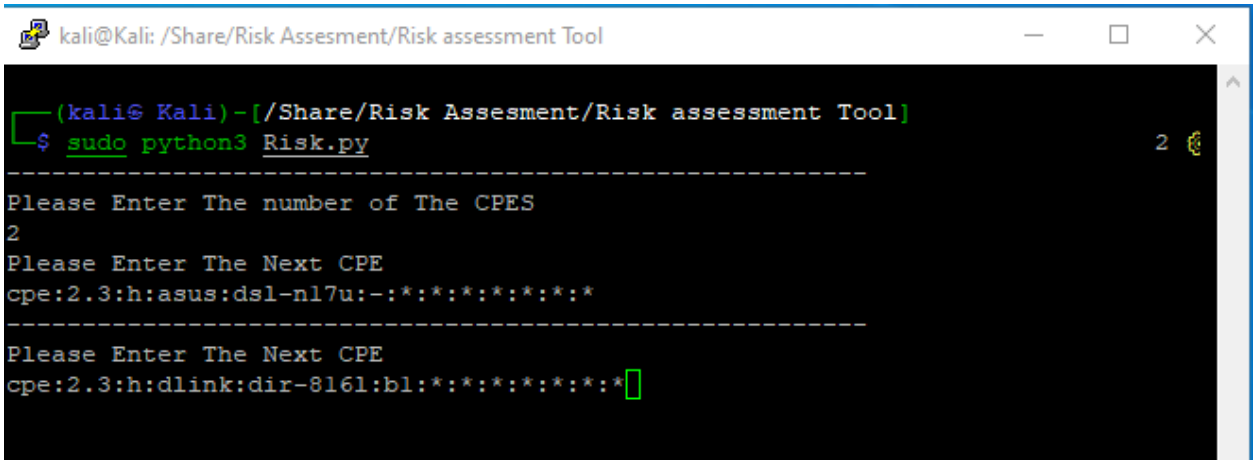
4.4.2. Εισαγωγή CPE αναγνωριστικών

Το επόμενο βήμα είναι η εισαγωγή των CPE αναγνωριστικών. Το λογισμικό A.R.A.T. v1.0 ζητάει να εισαχθεί ο αριθμός των CPE, δηλαδή των αγαθών (asset) για τα οποία πρόκειται να εφαρμοστεί η μεθοδολογία ARES (Εικόνα 4.2). Υπενθυμίζεται ότι κάθε CPE αντιστοιχεί σε ένα συγκεκριμένο τμήμα από το οποίο απαρτίζεται το μελετώμενο σύστημα (πχ. router, antivirus, μικροελεγκτής SCADA κλπ.)



Εικόνα 4 : Εισαγωγή πλήθους CPE

Με την εισαγωγή του πλήθους των CPE δημιουργείται μια λίστα από αυτά, εισάγοντάς τα ένα προς ένα. Για παράδειγμα εισάγοντας τον αριθμό 2, ζητείται ακολούθως να εισαχθούν 2 CPE αναγνωριστικά, όπως φαίνεται στην εικόνα 5 Σε περίπτωση λανθασμένης εισαγωγής των αναγνωριστικών δημιουργείται μήνυμα λάθους και το πρόγραμμα παύει να εκτελείται.



Εικόνα 5: Εισαγωγή CPE αναγνωριστικών

4.4.3. Εντοπισμός σχετικών CVE και CWE

Με την ολοκλήρωση των παραπάνω βημάτων, η εισαγωγή των δεδομένων που εισάγει ο χρήστης έχει ολοκληρωθεί. Σε αυτό το σημείο του αλγόριθμου γίνεται η διασύνδεση με τις βάσεις δεδομένων του NIST. Όπως έχει αναφερθεί, το NIST έχει εκτεταμένες βάσεις δεδομένων ο οποίες περιλαμβάνουν μεγάλο πλήθος στοιχείων λογισμικού και υλισμικού που έχουν κωδικοποιηθεί σε CPE. Μέσω των APIs, τα οποία διατίθενται ελεύθερα, εκτελούνται ερωτήματα (queries). Ως αποτέλεσμα, για κάθε CPE που έχει εισαχθεί στο λογισμικό επιστρέφονται οι ευπάθειές του (CVE), όπως φαίνεται στην Εικόνα 6.

```
----------------------------------------------------------
CPE Number 1
cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*
----------------------------------------------------------
CVE-2017-14698
CVSS =  9.8
CVE-2017-14699
CVSS =  6.5
CVE-2020-35219
CVSS =  9.8
```

Εικόνα 6: Τα CVE που αντιστοιχούν στο κάθε CPE, ύστερα από διασύνδεση του λογισμικού με τις βάσεις δεδομένων του NIST

Όπως φαίνεται στην εικόνα 4.4, για κάθε CPE εκτυπώνονται τα CVE (εδώ συγκεκριμένα για το πρώτο που εισήχθη), ενώ για κάθε ένα από τα CVE εκτυπώνεται η βαθμονόμηση - με κλίμακα έως το 10 - του CVSS. Στη συγκριμένη περίπτωση η βαθμολογία που προκύπτει μέχρι στιγμής είναι πολύ υψηλή, επιπέδου Critical, καθώς η συνολική βαθμολογία αυτή με την υψηλότερη τιμή (weakest link chain paradigm).

Το επόμενο βήμα του λογισμικού είναι η χρήση της API που δημιουργεί την σύνδεση με την βάση δεδομένων του NIST. Με την νέα διασύνδεση ανακτάται το CWE (αδυναμία) που αντιστοιχεί σε κάθε ευπάθεια CVE (Εικόνα 7).

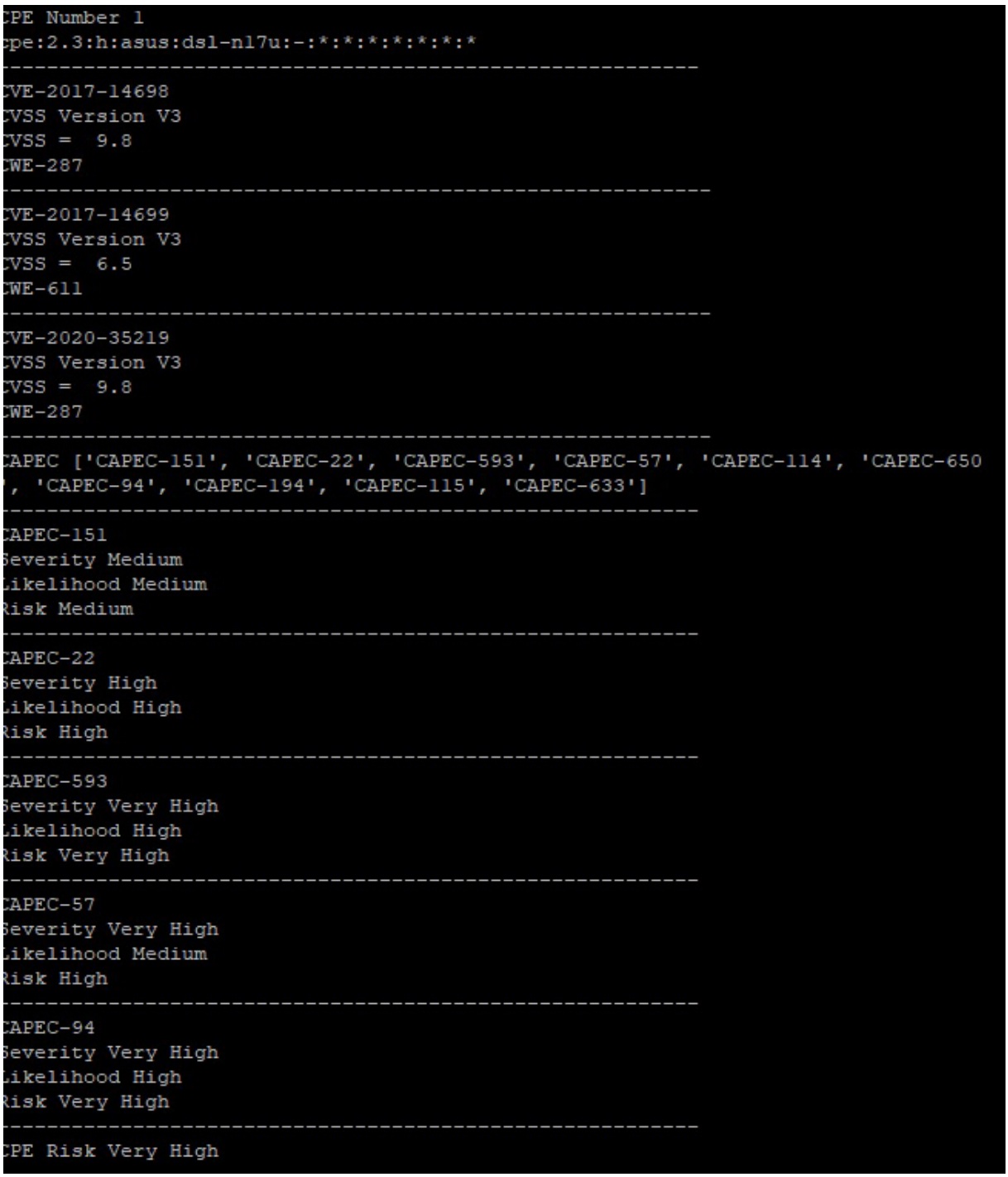
kali@Kali: /Share/Risk Assesment/Risk assessment Tool

```
----------------------------------------------------------
Please Enter The number of The CPES
1
Please Enter The Next CPE
cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*
----------------------------------------------------------
CPE Number 1
cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*
----------------------------------------------------------
CVE-2017-14698
CVSS Version V3
CVSS =  9.8
CWE-287
----------------------------------------------------------
CVE-2017-14699
CVSS Version V3
CVSS =  6.5
CWE-611
----------------------------------------------------------
CVE-2020-35219
CVSS Version V3
CVSS =  9.8
CWE-287
----------------------------------------------------------
```

Εικόνα 7: Αποτέλεσμα CWE για κάθε CPE

4.4.4. Εντοπισμός σχετικών CAPEC

Σε αυτό το σημείο πραγματοποιείται ακόμα μία διασύνδεση με τη βάση δεδομένων του MITRE. Ανακτάται, σε μορφή XML, όλη η βάση δεδομένων του MITRE και το αποτέλεσμα αυτής της διαδικασίας είναι η λίστα των CAPEC (μοτίβα επίθεσης) που αντιστοιχούν στο κάθε CWE. Τελικά, εφόσον τα CWE είναι ήδη συσχετισμένα με τα CPE, με τη διαδικασία αυτή λαμβάνεται η συσχέτιση των CPE με τα αντίστοιχα CAPEC. Με άλλα λόγια, μπορεί ο χρήστης, έχοντας ένα στοιχείο λογισμικού ή υλισμικού, ξέρει ποιο είναι το/τα μοτίβο/α επίθεσης που του αντιστοιχούν και το οποίο μοτίβο είναι αυτό που θα βαθμονομηθεί σε επόμενο βήμα. Στην εικόνα 8 παρουσιάζεται το αποτέλεσμα της συσχέτισης του μελετώμενου CPE με τα CAPEC.



Εικόνα 8: Αποτελέσματα CAPEC που συνδέονται με το μελετώμενο CPE

Μετά τη ολοκλήρωση των βημάτων του λογισμικού που έχουν ήδη περιγραφεί, το αποτέλεσμα που έχει εξαχθεί είναι:

- Ο εντοπισμός των CVE που σχετίζονται με το κάθε CPE

- Ο εντοπισμός των CWE που σχετίζονται με το κάθε CPE, μέσω της συσχέτισης CWE και CVE
- Ο εντοπισμός των CAPEC που σχετίζονται με το κάθε CPE, μέσω της συσχέτισης CAPEC και CWE
- Ο υπολογισμός των επιπέδου επικινδυνότητας (CVSS) για κάθε ένα από τα CAPEC που σχετίζονται με το κάθε CPE

Αυτά τα ενδιάμεσα αποτελέσματα είναι απαραίτητα για να εφαρμοστούν τα επόμενα δύο βήματα, το biclustering και η κατηγοριοποίηση, τα οποία οδηγούν στην εξαγωγή των τελικών αποτελεσμάτων.

4.4.5. Αντιστοίχιση CAPEC με CPE

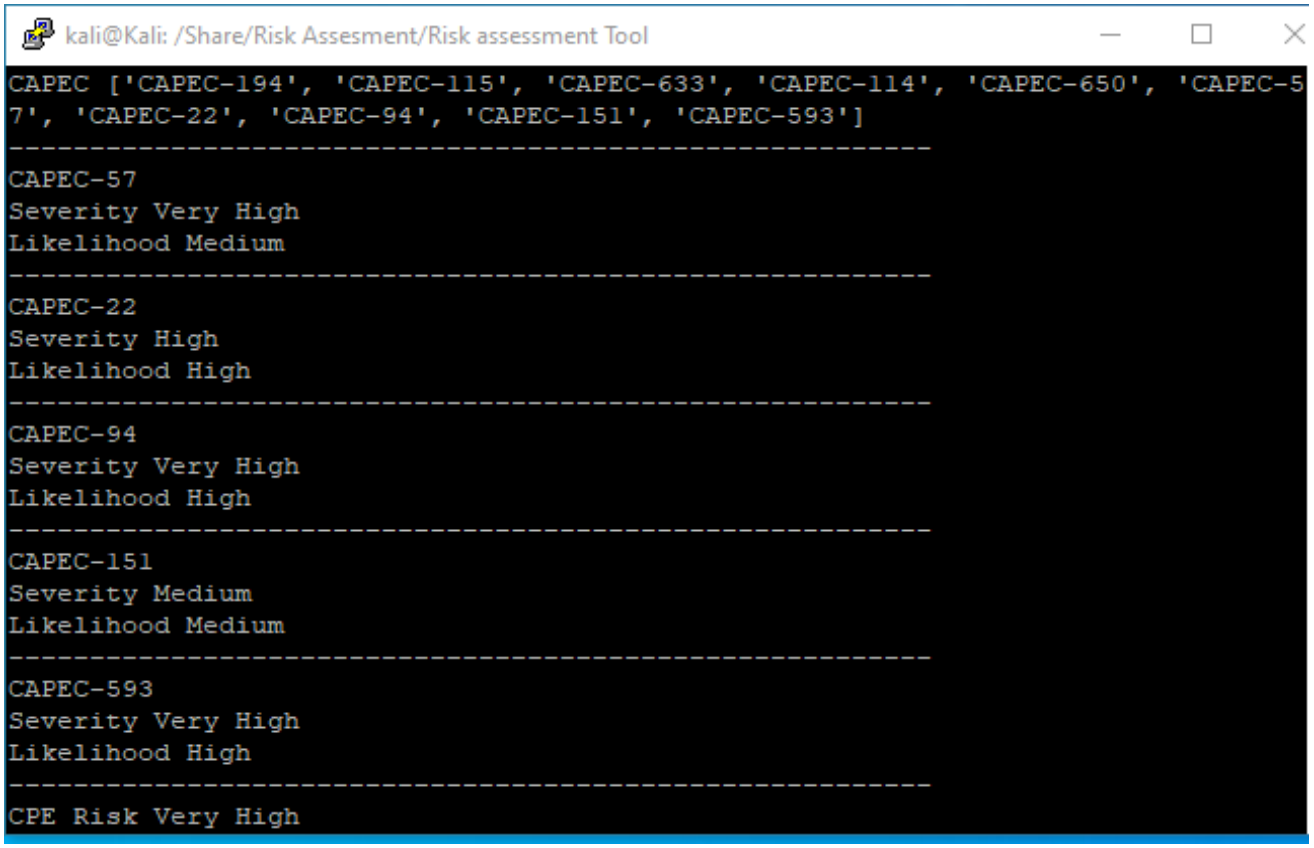
Το επόμενο ζητούμενο είναι ο προσδιορισμός της συνολικής επικινδυνότητας του μελετώμενου συστήματος, το οποίο, όπως έχει αναφερθεί και προηγουμένως, μπορεί να έχει παραπάνω από ένα CPE. Μέχρι στιγμής η όλη διαδικασία που έχει εφαρμοστεί είναι πλήρως αυτοματοποιημένη. Το λογισμικό συνδυάζει τα δεδομένα από τις βάσεις δεδομένων του MITRE και του NIST και εξάγει τα αποτελέσματα σχετικά με τα επίπεδα επικινδυνότητας. Η αυτοματοποιημένη διαδικασία ωστόσο είναι πιθανόν να μην οδηγεί σε σωστά αποτελέσματα, καθώς δεν λαμβάνει υπόψη ποιοτικά χαρακτηριστικά των CPE και των CAPEC. Για παράδειγμα, η διαδικασία μπορεί να αντιστοιχίσει ένα CAPEC που αφορά λογισμικό σε ένα CPE που αφορά υλισμικό, αντιστοίχιση η οποία δεν είναι σωστή, καθώς οι ευπάθειες του λογισμικού δεν επηρεάζουν το υλισμικό. Για την αντιμετώπιση αυτού του προβλήματος εφαρμόζεται το επόμενο βήμα, που είναι η αντιστοιχία CAPEC με CPE με εφαρμογή του biclustering.

Τα αποτελέσματα της εφαρμογής του biclustering προέκυψαν από την αξιοποίηση ιδιωτικής βιβλιοθήκης (https://github.com/asdimitriadis/cpe2capec). Πρόκειται για τον αλγόριθμο Univariate Marginal Distribution Algorithm (UMDA). Αυτός ο αλγόριθμος δημιουργεί ομάδες CPE και CAPEC με την μεγαλύτερη δυνατή συσχέτιση. Τα αποτελέσματα της εφαρμογής του περιλαμβάνονται σε ένα αρχείο τύπου xml (capec2cpe.xml) που έχει εξαχθεί από την ιδιωτική βιβλιοθήκη.

Συνοπτικά, ο τρόπος λειτουργίας αυτής της διαδικασίας (biclustering) είναι ο εξής: Τα CAPEC που έχουν προκύψει ως αποτέλεσμα από τα προηγούμενα βήματα ελέγχονται ένα προς ένα. Για κάθε CAPEC , το οποίο έχει προκύψει ως σχετιζόμενο με το μελετώμενο CPE, ελέγχεται εάν στο αρχείο **capec2cpe.xml** το CAPEC αυτό εμφανίζεται στην ίδια ομάδα με το δεδομένο CPE. Εάν αυτό είναι αληθές, τότε το CAPEC κρατείται στη λίστα. Εάν όχι, το CAPEC διαγράφεται, καθώς δεν υπάρχει λογική συσχέτιση του CAPEC με το CPE. Αυτή η διαδικασία biclustering διενεργείται για κάθε ένα CAPEC, μειώνοντας έτσι το πλήθος των CAPEC που θα εισαχθούν στη διαδικασία βαθμονόμησης.

4.4.6. Ταξινόμηση

Μετά την εφαρμογή του Biclustering και έχοντας μειώσει τα υποψήφια προς βαθμονόμηση CAPEC εκτελείται εκ νέου διαλογή μέσω της διαδικασίας της ταξινόμησης. Τα CAPEC και τα CPE ταξινομούνται με ανάλογο τρόπο και εντάσσονται σε όμοιες κατηγορίες. Με αυτόν τον τρόπο μπορεί να γίνει αντιπαραβολή μεταξύ τους, έτσι ώστε τα CAPEC που δεν ανήκουν στην ίδια κατηγορία με τα CPE να αφαιρούνται από την λίστα με τις συσχετίσεις CPE και CAPEC. Η διαδικασία που αναφέρθηκε υλοποιείται στο λογισμικό όπως φαίνεται στην εικόνα 9.



Εικόνα 9: Αποτελέσματα συσχέτισης CPE με CAPEC ύστερα από την εφαρμογή ταξινόμησης

4.4.7. Βαθμονόμηση – Εκτίμηση Επικινδυνότητας

Το τελευταίο βήμα που εκτελεί το προτεινόμενο λογισμικό είναι η εκτίμηση της συνολικής επικινδυνότητας του μελετώμενου υπολογιστικού συστήματος. Έχοντας ως δεδομένο τα αποτελέσματα του προηγούμενου βήματος, δηλαδή τη λίστα με τα CAPEC που αντιστοιχούν σε κάθε CPE, στο επόμενο βήμα το λογισμικό ανακτά μέσα από την βάση δεδομένων του MITRE αρχείο τύπου xml, στο οποίο περιέχονται οι πρόσφατοι ορισμοί των CAPEC μαζί με χαρακτηριστικά τους, δηλαδή την πιθανότητα εμφάνισης (Likelihood) και την σοβαρότητα επιπτώσεων (Severity). Αυτή η λίστα αντιπαραβάλλεται με την αναθεωρημένη λίστα των CAPEC που είχε προκύψει από την διαδικασία του biclustering και βαθμονομείται για το κάθε CAPEC. Το εργαλείο ARES χρησιμοποιεί ένα αλγόριθμό μου μεταφράζει την πιθανότητα εμφάνισης και την σοβαρότητα επιπτώσεων σε ρίσκο. Το προτεινόμενο λογισμικό υιοθετεί και κατ' επέκταση υλοποιεί την χρήση του αλγόριθμου του ARES για την εκτίμηση της συνολικής επικινδυνότητας. Ο πίνακας 5 παρουσιάζει το αποτέλεσμα της συνολικής επικινδυνότητας ανάλογα με το Severity και το Likelihood.

Impact / Likelyhood	None	Very Low	Low	Medium	High	Very high
Low	None	Low	Low	Medium	Medium	High
Medium	None	Low	Medium	Medium	High	High
High	None	Low	Medium	High	High	Very High

Πίνακας 5: Μοντέλο επικινδυνότητας (risk model) του ARES (Dimitriadis, et al., 2020)

4.5 Σημεία Αδυναμίας

Το προτεινόμενο λογισμικό παρουσιάζει κάποια σημεία αδυναμίας κατά την εκτέλεσή του, τα οποία οφείλονται κυρίως στο γεγονός της διασύνδεσης με πολλές βάσεις δεδομένων. Έχουν εντοπιστεί δύο σημεία αδυναμίας, τα οποία παρουσιάζονται στην συνέχεια, μαζί με τους προτεινόμενους τρόπους αντιμετώπισής τους.

4.5.1. Αδυναμία σχετιζόμενη με CWE

Κατά την εφαρμογή του αλγορίθμου και την εκτέλεση του λογισμικού παρατηρείται το φαινόμενο ότι κάποια CVE που έχουν εντοπιστεί δεν αντιστοιχίζονται με CWE, οπότε το αποτέλεσμα που παράγεται είναι είτε "NVD-CWE-Other" είτε "NVD-CWE-noinfo". Σε αυτήν την περίπτωση σταματάει η διαδικασία και συνεχίζει στο επόμενο CVE. Χρησιμοποιείται η βαθμονόμηση του αρχικού CVE, χωρίς την χρήση των επόμενων σταδίων (αντιστοίχιση με CWE και CAPEC). Το αρχικό CVE έχει ήδη βαθμονομηθεί με την χρήση του CVSS, από το οποίο προκύπτει η επικινδυνότητα. Η βαθμονόμηση του CVSS μετατρέπεται στις βαθμίδες Very Low, Low, Medium, High, Very High και μαζί με τα αποτελέσματα που θα προκύψουν από τα υπόλοιπα CAPEC λαμβάνεται υπόψη στον υπολογισμός της συνολικής επικινδυνότητας.

```
------------------------------------------------------------
CPE Number 1
cpe:2.3:a:microsoft:internet_information_server:7.5:*:*:*:*:*:*:*
------------------------------------------------------------
CVE-1999-0229
CVSS =  5.0
NVD-CWE-Other
CVE-2000-0115
CVSS =  5.0
NVD-CWE-Other
CVE-2007-0087
CVSS =  7.8
NVD-CWE-Other
CVE-2013-0941
CVSS =  2.1
CWE-310
CVE-2013-0942
CVSS =  4.3
CWE-79
CAPEC ['CAPEC-63', 'CAPEC-588', 'CAPEC-209', 'CAPEC-85', 'CAPEC-592', 'CAPEC-591']
------------------------------------------------------------
```

Εικόνα 10: Παράδειγμα αποτελέσματος NVD-CWE-Other

Παράδειγμα του προαναφερθέντος ζητήματος παρουσιάζεται στην εικόνα 10. Για την περίπτωση του CPE "**cpe:2.3:a:microsoft:internet_information_server:7.5:*:*:*:*:*:*:***", το λογισμικό εντοπίζει συσχέτιση με τα CVEs CVE-1999-0229, CVE-2000-0115, CVE-2007-0087, για τα οποία δεν εντοπίζεται συσχέτιση με CWE, οπότε εμφανίζεται ως αποτέλεσμα NVD-CWE-Other. Αυτό σημαίνει πως δεν εντοπίζεται και συσχετιζόμενο CAPEC, επομένως σε αυτή την περίπτωση χρησιμοποιείται το CVSS του CPE.

4.5.2. Αδυναμία Σχετιζόμενη με CAPEC

Μια ακόμα πιθανή περίπτωση είναι η εξής: ένα CVE μπορεί να έχει συσχετιζόμενα CWE, αλλά τα CAPEC που παράγει να μην έχουν τα απαραίτητα χαρακτηριστικά, δηλαδή likelihood (πιθανότητα εμφάνισης) ή/και severity (σοβαρότητα επιπτώσεων). Αυτό οφείλεται σε προβλήματα που παρουσιάζουν οι βάσεις δεδομένων NIST και MITRE καθώς είναι δυναμικές, και συνεχώς εμπλουτίζονται με νέα στοιχεία, νέες ευπάθειες και μοτίβα επίθεσης. Το προτεινόμενο λογισμικό είναι εξαρτώμενο από αυτή την δυναμικότητα, καθώς όλα τα ερωτήματα (queries) μέσω API ή xml αρχείου γίνονται σε πραγματικό χρόνο κατά την διάρκεια εκτέλεσης του.

Όταν εμφανιστεί αυτή η περίπτωση, χρησιμοποιούνται τα χαρακτηριστικά του γονικού CAPEC, εάν αυτό υπάρχει. Σε αντίθετη περίπτωση, ακολουθείται η διαδικασία που περιγράφεται στην §4.3.1.

Κεφάλαιο 5
Παράδειγμα Εφαρμογής

Κατά την εκπόνηση της παρούσας μεταπτυχιακής διατριβής χρησιμοποιείται συγκεκριμένο υπολογιστικό σύστημα, το οποίο χρησιμοποιείται ως δοκίμιο, ως πρότυπο για την εκτέλεση του λογισμικού για τον εντοπισμό και τη διόρθωση των λαθών του.

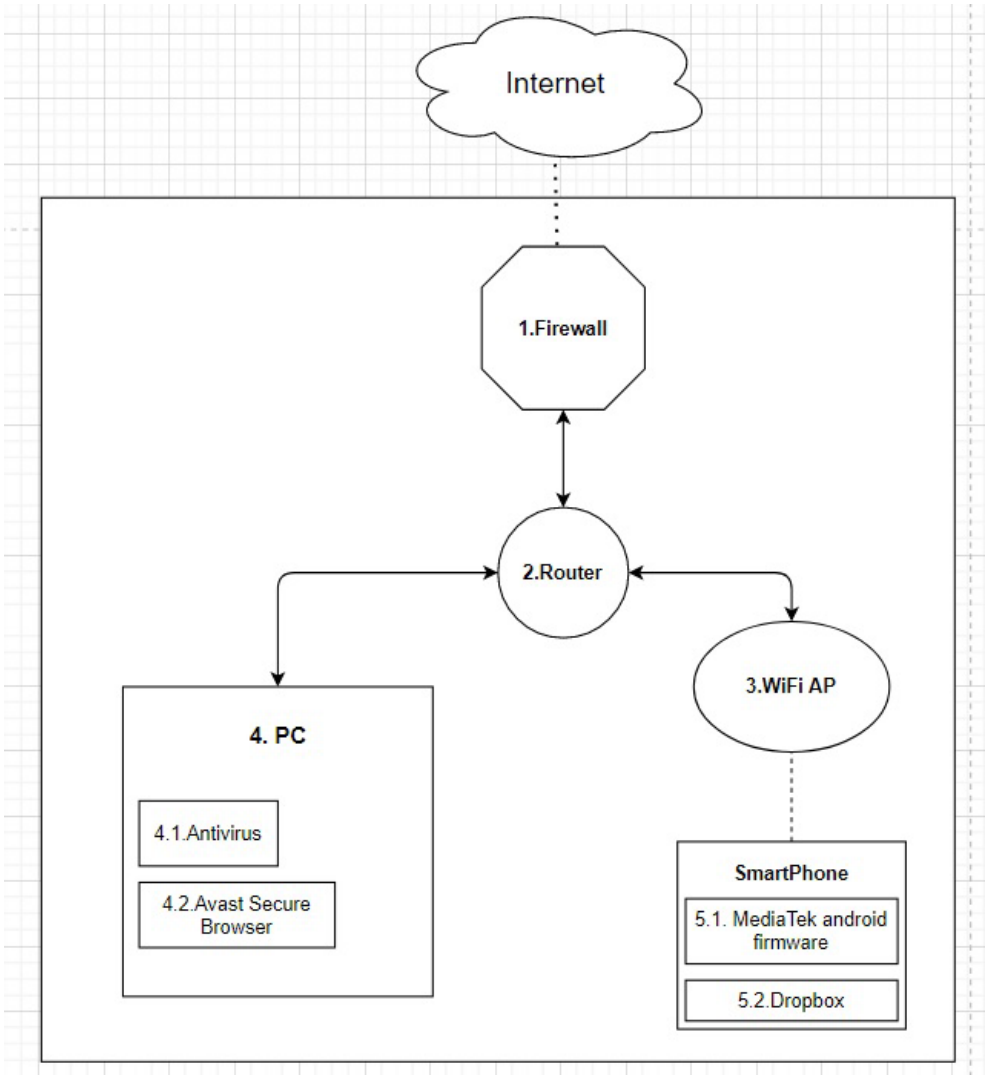
5.1 Πρότυπο Σύστημα

Το πρότυπο μελετώμενο σύστημα είναι ένα σύνθετο σύστημα που αποτελείται από μια δικτυακή υποδομή , μια υπολογιστική υποδομή και λογισμικό και συντίθεται από τα παρακάτω στοιχεία (Εικόνα 11).

- Δικτυακή υποδομή:
  - ένα τείχος προστασίας (Firewall)

- έναν δρομολογητή (router)
- ένα ασύρματο σημείο πρόσβασης (Wifi Access Point)

- Υπολογιστική υποδομή:
  - ένας προσωπικός υπολογιστής PC
  - ένα κινητό (Smartphone)
- Λογισμικό:
  - το αντι-ικο πρόγραμμα
  - ο δικτυακός φυλλομετρητής (Avast Secure Browser)
  - το λειτουργικό του κινητού (Mediatek Android FirmWare)
  - το εγκατεστημένο λογισμικό αποθήκευσης στο σύννεφο (DropBox)



Εικόνα 11: Σχηματική αναπαράσταση του πρότυπου συστήματος

Τα αναγνωριστικά CPE που αντιστοιχούν στο πρότυπο μελετώμενο σύστημα αποτυπώνονται στην παρακάτω λίστα:

- ASUS DSL-N17U:
  cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*:*
- A10 Networks Advanced Core Operating System (ACOS) Web Application Firewall (WAF) 4.1.1 Patch 3:
  cpe:2.3:a:a10networks:acos_web_application_firewall:4.1.1:p3:*:*:*:*:*:*:*
- D-Link DIR-816L B1:
  cpe:2.3:h:dlink:dir-816l:b1:*:*:*:*:*:*:*:*
- Microsoft Internet Information Services (IIS) 7.5:
  cpe:2.3:a:microsoft:internet_information_server:7.5:*:*:*:*:*:*:*:*
- Malwarebytes 3.3.1.2183 Premium Edition:
  cpe:2.3:a:malwarebytes:malwarebytes:3.3.1.2183:*:*:*:premium:*:*:*
- Avast Secure Browser 77.1.1831.91:
  cpe:2.3:a:avast:secure_browser:77.1.1831.91:*:*:*:*:*:*:*:*
- Mediatek MT8163 Firmware for Android:
  cpe:2.3:o:mediatek:mt8163_firmware:-:*:*:*:*:*:android:*:*
- Dropbox 98.2.2 for Android:
  cpe:2.3:a:dropbox:dropbox:98.2.2:*:*:*:*:*:android:*:*

Τα δεδομένα αυτά είναι τα δεδομένα τα οποία εισάγονται στο λογισμικό ARAT, ώστε να γίνουν οι δοκιμές. Η σειρά με την οποία θα εισαχθούν τα δεδομένα δεν έχει σημασία καθώς δεν επηρεάζει το τελικό αποτέλεσμα, αφού η επικινδυνότητα δεν δρα αθροιστικά αλλά μόνο ως μέγιστο όριο.

Η λογική που χρησιμοποιείται είναι αυτή του παραδείγματος της αλυσίδας. Σύμφωνα με αυτό, μία αλυσίδα (που μπορεί να θεωρηθεί το δοκίμιο ως αλυσίδα) είναι τόσο ισχυρή όσο ισχυρός είναι ο πιο αδύναμος κρίκος της. Επομένως, με το χειρότερο σενάριο (worst case scenario premise) υπολογίζεται το υψηλότερη τιμή επικινδυνότητάς που μπορεί να προκύψει. Η θεώρηση της αλυσίδας μπορεί να εφαρμοστεί στο πρότυπο μελετώμενο σύστημα, καθώς αυτό αποτελείται από διάφορα ξεχωριστά στοιχεία, έτσι αν για παράδειγμά ένα εκ των οποίων τεθεί εκτός λειτουργίας τότε τίθεται εκτός λειτουργίας όλο το σύστημα.

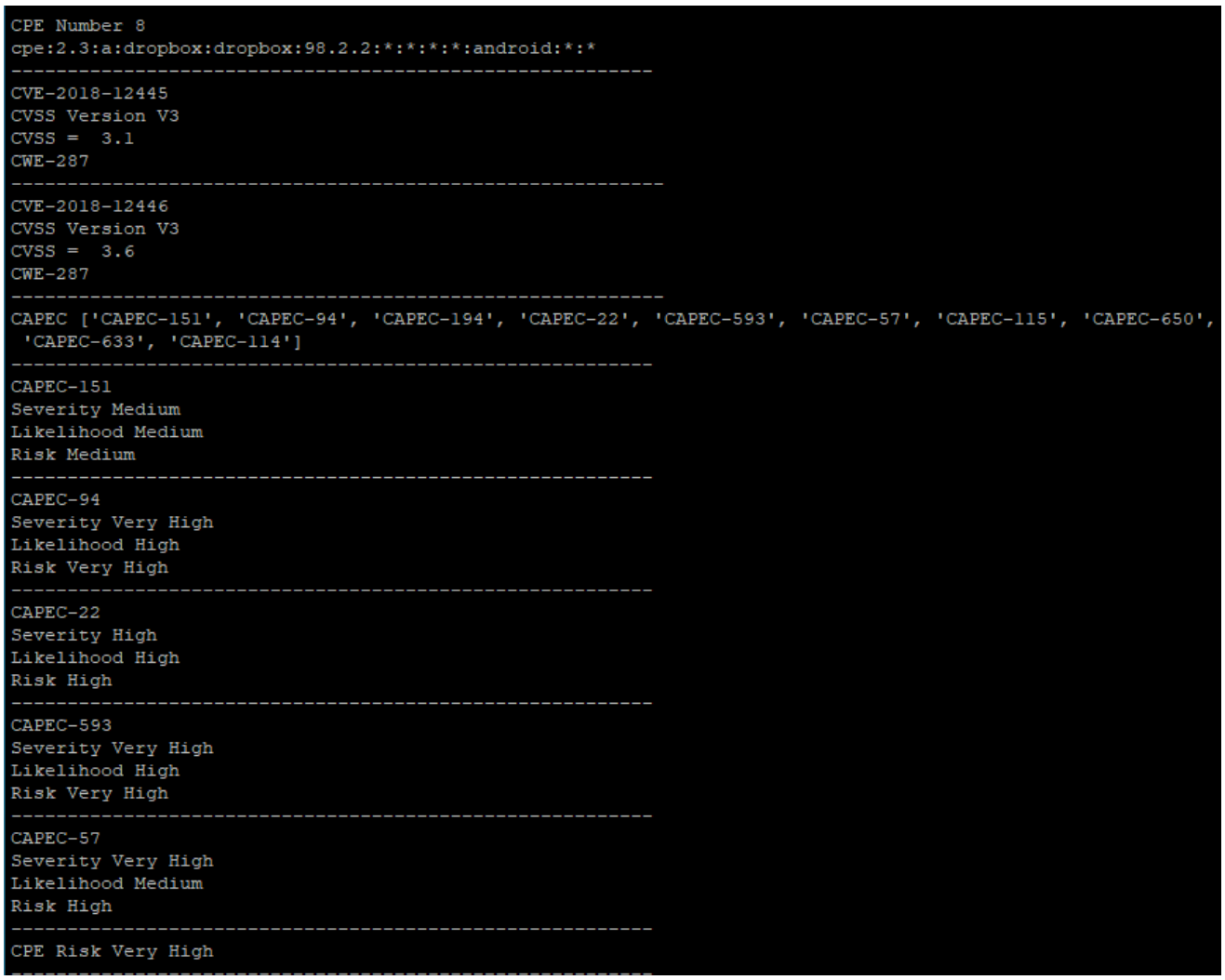
5.2 Εκτέλεση Λογισμικού

Το πρώτο βήμα στην εκτέλεση του προγράμματος είναι η εισαγωγή των δεδομένων μέσω της γραμμής εντολών (command line), όπως φαίνεται στην εικόνα 12.

```
┌──(kali㉿ Kali)-[/Share/Risk Assesment/Risk assessment Tool]
└─$ sudo python3 Risk.py
------------------------------------------------------
Please Enter The number of The CPES
8
Please Enter The Next CPE
cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:a:a10networks:acos_web_application_firewall:4.1.1:p3:*:*:*:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:h:dlink:dir-816l:b1:*:*:*:*:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:a:microsoft:internet_information_server:7.5:*:*:*:*:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:a:malwarebytes:malwarebytes:3.3.1.2183:*:*:*:premium:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:a:avast:secure_browser:77.1.1831.91:*:*:*:*:*:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:o:mediatek:mt8163_firmware:-:*:*:*:*:android:*:*
------------------------------------------------------
Please Enter The Next CPE
cpe:2.3:a:dropbox:dropbox:98.2.2:*:*:*:*:android:*:*
```

Εικόνα 12: Εισαγωγή δεδομένων στην command-line

Στο συγκεκριμένο παράδειγμα τα στοιχεία του συστήματος είναι 8 στο πλήθος, οπότε και τα CPE που εισάγονται είναι 8. Μετά την είσοδο του τελευταίου (8ου) CPE, το λογισμικό συνεχίζει με την εκτέλεση του αλγορίθμου εντοπισμού CAPEC, μέσω της χρήσης των CWE και CVE. Στην εικόνα 13 παρουσιάζεται το αποτέλεσμα για την τελευταία CPE.



Εικόνα 13: Αποτέλεσμα επικινδυνότητας για το 8ο CPE του συστήματος

Στο πρώτο μέρος των αποτελεσμάτων απεικονίζεται η ευπάθεια (CVE) που εντοπίζεται μαζί με την αδυναμία (CWE) που αντιστοιχεί σε αυτήν, ενώ βαθμονομείται σε κλίμακα CVSS και αναφέρεται και η έκδοση του CVSS που χρησιμοποιείται. Η χρήση της έκδοσης 3 του CVSS είναι η προκαθορισμένη επιλογή και εφαρμόζεται παντού, εκτός από τις περιπτώσεις που δεν έχει γίνει βαθμονόμηση σε αυτήν την έκδοση και επομένως χρησιμοποιείται η έκδοση 2.

Κατά την εκτέλεση του προγράμματος εντοπίζονται οι CAPEC που αντιστοιχούν στα προαναφερόμενα CWE οπότε και απεικονίζονται (Εικόνα 14).

```
CAPEC ['CAPEC-151', 'CAPEC-94', 'CAPEC-194', 'CAPEC-22', 'CAPEC-593', 'CAPEC-57', 'CAPEC-115', 'CAPEC-650',
 'CAPEC-633', 'CAPEC-114']
```

Εικόνα 14: Τα CAPEC που αντιστοιχούν στα CWE

Σε αυτό το σημείο εκτελείται η διαδικασία του biclustering μέσω της χρήσης του αρχείου **capec2cpe.xml** που έχει αντληθεί από το https://github.com/asdimitriadis/cpe2capec. Με το biclstering γίνεται το ξεκαθάρισμα των CAPEC που έχουν προκύψει από την προηγούμενη διαδικασία. Έτσι για παράδειγμα από 10 CAPEC που έχουν προκύψει για το 8ο CPE αφαιρούνται τα 5 και μένουν τα 5 που έχουν περάσει και από το φίλτρο της ταξινόμησης, το οποίο εφαρμόζεται αμέσως μετά το biclustering.

Για λόγους καταγραφής και αντιπαραβολής με το τελικό αποτέλεσμα, δημιουργούνται για κάθε CPE οι λίστες των CAPEC που προκύπτουν από τον αυτοματοποιημένο αλγόριθμο εξαγωγής CAPEC από τις βάσεις δεδομένων του NIST και του MITRE. Αυτό το βήμα υλοποιείται πριν τον διαχωρισμό πραγματοποιείται μέσω του biclustering και της ταξινόμησης. Τα αρχεία που προκύπτουν έχουν μορφή txt και παρατίθενται στους πίνακες 5 έως 9.

Αρχείο cpe1.txt	**Αρχείο cpe2.txt**
CAPEC – 194	CAPEC – 110
CAPEC – 151	CAPEC – 66
CAPEC – 22	CAPEC – 108
CAPEC – 633	CAPEC – 7
CAPEC – 650	
CAPEC – 114	
CAPEC – 115	
CAPEC – 57	
CAPEC – 94	
CAPEC – 593	

Πίνακας 6: Λίστα CAPEC για τα CPE 1 και CPE 2

Αρχείο cpe3.txt	Αρχείο cpe4.txt
CAPEC – 592	CAPEC – 592
CAPEC – 591	CAPEC – 591
CAPEC – 588	CAPEC – 588
CAPEC – 85	CAPEC – 85
CAPEC – 209	CAPEC – 209
CAPEC – 63	CAPEC – 63

Πίνακας 7: Λίστα CAPEC για τα CPE 3 και CPE 4

Αρχείο cpe5.txt	
CAPEC – 473	CAPEC – 28
CAPEC – 13	CAPEC – 24
CAPEC – 72	CAPEC – 45
CAPEC – 261	CAPEC – 42
CAPEC – 267	CAPEC – 110
CAPEC – 8	CAPEC – 67
CAPEC – 52	CAPEC – 53
CAPEC – 79	CAPEC – 88
CAPEC – 135	CAPEC – 3
CAPEC – 66	CAPEC – 9
CAPEC – 120	CAPEC – 10
CAPEC – 109	CAPEC – 46
CAPEC – 250	CAPEC – 80
CAPEC – 73	CAPEC – 43
CAPEC – 78	CAPEC – 71
CAPEC – 7	CAPEC – 47
CAPEC – 64	CAPEC – 153
CAPEC – 23	

Πίνακας 8: Λίστα CAPEC για το CPE 5

Αρχείο cpe6.txt	Αρχείο cpe7.txt
CAPEC – 592	CAPEC – 43
CAPEC – 591	CAPEC – 108
CAPEC – 588	CAPEC – 88
CAPEC – 85	CAPEC – 15
CAPEC – 209	
CAPEC – 63	

Πίνακας 9: Λίστα CAPEC για τα CPE 6 και CPE 7

Αρχείο cpe8.txt	
CAPEC – 194	CAPEC – 115
CAPEC – 151	CAPEC – 57
CAPEC – 22	CAPEC – 94
CAPEC – 633	CAPEC – 593
CAPEC – 650	
CAPEC – 114	

Πίνακας 10: Λίστα CAPEC για το CPE 8

Το επόμενο βήμα είναι η εκτέλεση του φίλτρου κατηγοριοποίησης. Το φίλτρο κατηγοριοποίησης υλοποιείται με την χρήση δύο αρχείων τύπου csv, τα οποία εν προκειμένω έχουν ονομαστεί **cpes.csv** και **capecs.csv.** Τα αρχεία αυτά αποτελούνται από δύο στήλες. Όσον αφορά το αρχείο cpes.csv (Πίνακας 10), στην πρώτη στήλη αναφέρονται τα CPE ενώ στη δεύτερη στήλη η κατηγορία λογισμικού και υλισμικού που αντιστοιχούν σε κάθε CPE. Αντίστοιχα, το αρχείο capecs.csv (Πίνακας 11), αντιστοιχίζει το κάθε CAPEC (1η στήλη) με την κατηγορία λογισμικού και υλισμικού (2η στήλη) που αυτό αντιστοιχεί. Το προτεινόμενο λογισμικό εντοπίζει μέσα στο πρώτο αρχείο το αναγνωριστικό του CPE και ελέγχει την κατηγορία στην οποία ανήκει. Στην συνέχεια εντοπίζει στο δεύτερο αρχείο το CAPEC και την κατηγορία του. Αν η κατηγορία του CAPEC ταυτίζεται με την κατηγορία του CPE, τότε το προτεινόμενο λογισμικό συνεχίζει να χρησιμοποιεί το εν λόγω CAPEC στη διαδικασία βαθμονόμησης του CPE. Σε διαφορετική περίπτωση, το CAPEC αφαιρείται από τη λίστα, καθώς θεωρείται μη σχετικό με το CPE και δεν συμμετέχει στη διαδικασία βαθμονόμησης.

CPE	Κατηγορία
cpe:2.3:h:asus:dsl-n17u:-:*:*:*:*:*:*:*	networking_communications
cpe:2.3:a:a10networks:acos_web_application_firewall:4.1.1:p3:*:*:*:*:*:*	networking_communications
cpe:2.3:h:dlink:dir-816l:b1:*:*:*:*:*:*:*	networking_communications
cpe:2.3:a:40icrosoft:internet_information_server:7.5:*:*:*:*:*:*:*	kernel_distributions
cpe:2.3:a:40icrosoft40es:40icrosoft40es:3.3.1.2183:*:*:*:premium:*:*:*	support_utilities
cpe:2.3:a:avast:secure_browser:77.1.1831.91:*:*:*:*:*:*:*	web_browsers
cpe:2.3:o:mediatek:mt8163_firmware:-:*:*:*:*:android:*:*	android
cpe:2.3:a:dropbox:dropbox:98.2.2:*:*:*:*:android:*:*	android
cpe:2.3:h:hp:s1000-e_vpn_firewall_appliance:jd272a:*:*:*:*:*:*:*	networking_communications
cpe:2.3:h:40icros:access_router:v200r002c01spc200:*:*:*:*:*:*:*	networking_communications
cpe:2.3:h:acexy:wireless-n_wifi_repeater:1.0:*:*:*:*:*:*:*	networking_communications
cpe:2.3:o:40icrosoft:windows_10:1803:*:*:*:*:*:x64:*	kernel_distributions
cpe:2.3:a:40icrosoft:windows_defender:-:*:*:*:*:*:*:*	support_utilities
cpe:2.3:a:microsoft:edge:-:*:*:*:*:*:*:*	web_browsers
cpe:2.3:o:google:android:8.0:*:*:*:*:*:*:*	android
cpe:2.3:a:samsung:members:3.9.10.11:*:*:*:*:*:*:*	android

Πίνακας 11: Δεδομένα αρχείου cpes.csv

CAPEC	Κατηγορία
CAPEC-13	Android
CAPEC-88	Android
CAPEC-22	kernel_distributions
CAPEC-63	kernel_distributions
CAPEC-57	networking_communications
CAPEC-94	networking_communications
CAPEC-108	Software
CAPEC-114	Software
CAPEC-151	Software
CAPEC-194	Software
CAPEC-633	Software
CAPEC-83	Software
CAPEC-115	web_browsers
CAPEC-209	web_browsers
CAPEC-588	web_browsers
CAPEC-591	web_browsers
CAPEC-592	web_browsers
CAPEC-593	web_browsers
CAPEC-650	web_browsers
CAPEC-85	web_browsers

Πίνακας 12: Μερικά Δεδομένα αρχείου capecs.csv

5.3 Αποτελέσματα

Στο τέλος και αφού ολοκληρωθεί η διαδικασία biclustering και ταξινόμησης, προκύπτουν 5 CAPEC σχετικά με το 1ο CPE. Αυτά τα 5 CAPEC βαθμονομούνται μέσω της βάσης δεδομένων του MITRE και για κάθε ένα από αυτό προκύπτει Severity (σοβαρότητα επιπτώσεων), Likelihood (Πιθανότητα εμφάνισης) και τέλος το Risk (Επικινδυνότητα). Εντοπίζεται το CAPEC με το υψηλότερο ρίσκο, που στη συγκεκριμένη περίπτωση είναι Very High (Πολύ Υψηλή Επικινδυνότητα). Άρα, η επικινδυνότητα για το 1ο CPE είναι Very High (Πολύ Υψηλή).

Με αντίστοιχη διαδικασία γίνεται ο υπολογισμός της επικινδυνότητας και για τα άλλα 7 CPE. Από την στιγμή όμως που έχει εντοπισθεί έστω και ένα CPE με πολύ υψηλή επικινδυνότητα, η συνολική βαθμολογία όλου του μελετώμενου συστήματος (δοκιμίου) είναι πολύ υψηλή, καθώς η τιμή της επικινδυνότητας από αυτό το CPE αποτελεί το ελάχιστο όριο. Στην εικόνα 15 απεικονίζεται το τελικό αποτέλεσμα του λογισμικού για το σύνολο του μελετώμενου συστήματος.

Εικόνα 15: Τελικό αποτέλεσμα

5.4 Παρατηρήσεις

Η εκτύπωση των αποτελεσμάτων γίνεται με τη χρήση 2 αρχείων τύπου csv, ενώ επίσης τυπώνονται και αρχεία κείμενου ίσα σε πλήθος με το πλήθος των CPE που έχουν εισαχθεί ως δεδομένα. Στα παραγόμενα αρχεία καταγράφονται τα CAPEC που έχουν υπολογισθεί. Στα αρχεία csv τυπώνονται οι επικινδυνότητες του κάθε ενός από τα CPE, ενώ στο δεύτερο αρχείο οι συνολικές επικινδυνότητες. Τα παραγόμενα αρχεία παρατίθενται στους πίνακες 12 έως 16.

Αρχείο cpe1.csv		Αρχείο cpe2.csv	
CAPEC – 57	High	CAPEC – 66	High
CAPEC – 22	High	CAPEC – 108	Medium
CAPEC – 593	Very High	CAPEC – 7	High
CAPEC – 151	Medium	CAPEC – 110	Very High
CAPEC – 94	Very High		

Πίνακας 13: Αποτελέσματα των CPE 1 και CPE 2

Αρχείο cpe3.csv		Αρχείο cpe4.csv	
CAPEC – 592	Very High	CAPEC – 592	Very High
CAPEC – 591	Very High	CAPEC – 591	Very High
CAPEC – 63	Very High	CAPEC – 63	Very High
CAPEC – 588	Very High	CAPEC – 588	Very High
CAPEC – 85	Medium	CAPEC – 85	Medium

Πίνακας 14: Αποτελέσματα των CPE 3 και CPE 4

Αρχείο cpe5.csv			
CAPEC – 46	High	CAPEC – 8	High
CAPEC – 24	High	CAPEC – 43	Medium
CAPEC – 73	High	CAPEC – 42	High
CAPEC – 9	High	CAPEC – 13	Very High
CAPEC – 67	Very High	CAPEC – 267	High
CAPEC – 109	Medium	CAPEC – 45	High
CAPEC – 23	Very High	CAPEC – 88	High
CAPEC – 10	High	CAPEC – 120	Medium
CAPEC – 66	High	CAPEC – 7	High
CAPEC – 135	High	CAPEC – 28	High
CAPEC – 79	High	CAPEC – 52	High
CAPEC – 78	High	CAPEC – 3	Medium
CAPEC – 53	High	CAPEC – 71	Medium
CAPEC – 80	High	CAPEC – 64	High
CAPEC – 72	High	CAPEC – 110	Very High
CAPEC – 47	Medium		

Πίνακας 15: Αποτελέσματα του CPE 5

Αρχείο cpe6.csv		Αρχείο cpe7.csv	
CAPEC – 592	Very High	CAPEC – 88	High
CAPEC – 591	Very High	CAPEC – 43	Medium
CAPEC – 63	Very High	CAPEC – 108	Medium
CAPEC – 588	Very High	CAPEC – 15	High
CAPEC – 85	Medium		

Πίνακας 16: Αποτελέσματα των CPE 6 και CPE 7

Αρχείο cpe8.csv	
CAPEC – 57	High
CAPEC – 22	High
CAPEC – 593	Very High
CAPEC – 151	Medium
CAPEC – 94	Very High

Πίνακας 16: Αποτελέσματα του CPE 8

Στις περισσότερες περιπτώσεις παρατηρείται ότι το αποτέλεσμα είναι έως 5 CAPEC, εκτός από την περίπτωση του 5ου CPE, το οποίο αντιστοιχεί σε αντι-ιικο πρόγραμμα. Λόγω της φύσης του λογισμικού, αυτό το αποτέλεσμα είναι αναμενόμενο, μιας και τα αντι-ιικά προγράμματα γίνονται πιο συχνά στόχος επιθέσεων. Όπως φαίνεται από τα αποτελέσματα, για όλα τα CPE υπάρχουν CAPEC που έχουν πολύ υψηλό ρίσκο, επομένως το τελικό αποτέλεσμα που προκύπτει είναι πολύ υψηλού ρίσκου.

5.5 Εναλλακτική προσέγγιση

Ο υπολογισμός της επικινδυνότητας θα μπορούσε να γίνει αθροιστικά δηλαδή να υπολογιζόταν το ρίσκο για το τοίχος προστασίας που είναι το πρώτο σημείο επαφής που έχει να κάνει με μοτίβο επίθεσης από το διαδίκτυο. Όπως διαπιστώνεται η επικινδυνότητα για αυτό το στοιχείο του δοκιμίου είναι Very High (Πολύ Υψηλή) επομένως, η αρχική στάθμη επικινδυνότητας είναι ήδη πολύ υψηλή και γίνεται κατανοητό ότι δεν μπορεί να ανέβει σε υψηλότερο επίπεδο. Εξάγεται το συμπέρασμα ότι και με την εναλλακτική αυτή προσέγγιση το τελικό αποτέλεσμα είναι το ίδιο με την αρχική προσέγγιση, ότι δηλαδή η ύπαρξη της πιο υψηλής στάθμης επικινδυνότητας μέσα σε

κάποιο στοιχείο του δοκιμίου ανεβάζει το συνολικό επίπεδο επικινδυνότητας του όλου δοκιμίου - συστήματος.

5.6 Συμπεράσματα

Το Λογισμικό που αναπτύχθηκε με το όνομα ARAT για συντομία, εκτιμά τα επίπεδα επικινδυνότητας υπολογιστικών συστημάτων και παράγει δύο τελικά αποτελέσματα: το επί μέρους επίπεδο επικινδυνότητας για κάθε CPE και το συνολικό επίπεδο επικινδυνότητας για όλο το δοκίμιο – σύστημα.

Όπως προαναφέρθηκε, είναι σύνηθες σε κάθε υπολογιστικό σύστημα να υπάρχουν επί μέρους αγαθά/στοιχεία (assets) με πολύ υψηλή λόγω της φύσης τους επικινδυνότητα. Τα αντι-ιικά προγράμματα είναι μια τέτοια περίπτωση, καθώς πολλές επιθέσεις έχουν ως στόχο την πρώτη γραμμή άμυνας, δηλαδή τα αντι-ιικά προγράμματα. Σύμφωνα με την μεθοδολογία του ARAT, εάν υπάρχει έστω και ένα αγαθό/στοιχείο (asset) με πολύ υψηλή επικινδυνότητα, τότε η συνολική επικινδυνότητα όλου του μελετώμενου συστήματος θα είναι πολύ υψηλή.

Αυτό είναι ένα πρώτο και πολύ βασικό συμπέρασμα της έρευνας που διενεργήθηκε στην παρούσα μεταπτυχιακή διατριβή. Εκ πρώτης όψεως, αυτό το συμπέρασμα φαίνεται τετριμμένο, καθώς ένα εργαλείο ή μεθοδολογία που εξάγει συνεχώς το ίδιο αποτέλεσμα είναι ίσως περιττό. Ωστόσο, τα αποτελέσματα του ARAT πρέπει να εξετάζονται από μια διαφορετική οπτική γωνία. Το σημαντικότερο ίσως αποτέλεσμα του εργαλείου ARAT είναι το επίπεδο επικινδυνότητας του κάθε asset ξεχωριστά. Από τη συγκριτική μελέτη των επιπέδων επικινδυνότητας των επί μέρους assets μπορούν να εξαχθούν χρήσιμα συμπεράσματα.

Μέσα από τα αρχεία που δημιουργούνται μπορεί κάποιος να εντοπίσει τα CAPEC με την πιο υψηλή βαθμολογία. Ακολουθώντας την αντίστροφη πορεία, το CAPEC αντιστοιχίζεται με κάποιο CPE, δηλαδή με κάποιο τμήμα του συστήματος. Με τα αποτελέσματα του ARAT μπορούν να εντοπιστούν τα CPE που έχουν το υψηλότερο επίπεδο επικινδυνότητας και στη συνέχεια να διεξαχθεί στοχευμένο project μείωσης της επικινδυνότητάς του. Για παράδειγμα, εάν το αντικείμενο αυτό είναι αντιικό πρόγραμμα, μπορεί να εγκατασταθεί κάποιο άλλο περισσότερο ασφαλές. Εάν είναι λογισμικό μπορεί να αντικατασταθεί με νεότερη έκδοση ή αν είναι υλισμικό

(hardware) να δρομολογηθεί η αντικατάστασή του, ή να γίνει ενημέρωση του υλικολογισμικού (firmware) του, ή να απομονωθεί η πρόσβαση σε αυτό το υλισμικό.

Ο δεύτερος τύπος αποτελεσμάτων που είναι το συνολικό μπορεί να χρησιμοποιηθεί για να επαληθευτεί η μείωση κινδύνου των επιμέρους τιμών επικινδυνότητας. Καθώς γίνεται επίτευξη μείωσης αρκετών επιμέρους τιμών επικινδυνότητας επιτυγχάνεται η μείωση της συνολικής. Φυσικά, η αντικατάσταση ενός επί μέρους στοιχείου, στα πλαίσια των δοκιμών για τη μείωση των επιπέδων επικινδυνότητας, έχει ως αποτέλεσμα την αλλαγή των υπαρχόντων CPE. Για παράδειγμα, στο δοκίμιο της §5.1, το τέταρτο CPE είναι ο Web Server της Microsoft IIS 7.5. Αυτή είναι μια παλιά έκδοση και η αλλαγή του Web Server σε νεότερη έκδοση αλλάζει το CPE δηλαδή γίνεται από cpe:2.3:a:microsoft:internet_information_server:7.5:*:*:*:*:*:*:* σε cpe:2.3:a:microsoft:internet_information_server:10.0:*:*:*:*:*:*:* . Επίσης, αλλάζουν και τα CAPEC που εντοπίζονται, με αποτέλεσμα να αλλάξουν όλα τα εξαγόμενα αποτελέσματα.

Το πλεονέκτημα της αλλαγής σε νεότερη έκδοση είναι, εκτός της προφανούς αύξησης της λειτουργικότητας λόγων των νέων δυνατοτήτων που προκύπτουν από μία καινούρια έκδοση λογισμικού, και η βελτίωση του συνολικού προφίλ επικινδυνότητας. όμως αυτό εμπεριέχει την πιθανότητα καθώς διενεργείται αυτή η αλλαγή η νέα έκδοση του λογισμικού (IIS 10.0) να έχει υψηλότερη επικινδυνότητα από το αρχικό (IIS 7.5). Στην συγκεκριμένη περίπτωση, εξετάστηκε πως αυτό δεν είναι κάτι που τελικά συμβαίνει.

Η χρήση των csv αρχείων που περιέχουν τις κατηγορίες των capec και των cpe δεν παρουσιάζουν την όλη συσχέτιση των δυο αυτών προτύπων καθώς κάθε κατηγορία έχει δοθεί με την όσο δυνατόν μεγαλύτερη συνάφεια χωρίς όμως να χαρακτηρίζεται ο βαθμός συνάφειας και στην ουσία είναι εντελώς επίπεδη αυτή η συσχέτιση. Είναι κάτι που θα μπορούσε να διερευνηθεί περισσότερο στο μέλλον και να ενσωματωθεί στο λογισμικό για να υπάρχει το βέλτιστο δυνατό αποτέλεσμα, πιθανώς με την χρήση συντελεστών συνάφειας που θα αποκαλύπτουν πόσο κοντά είναι οι πραγματικές κατηγορίες των capec και των cpe.

Το εργαλείο ARAT λόγω της γρήγορης απόκρισης του σε συνδυασμό με ένα λογισμικό εντοπισμού cpe και ενός patch manager θα μπορούσε να χρησιμοποιηθεί για την έγκαιρο εντοπισμό απειλών, αναζήτηση λύσης, και επίλυση προβλήματος σε πολύ μικρό χρόνο με αυτοματοποίηση. Η ροή μιας τέτοιας διαδικασίας θα ήταν η εξής: Το λογισμικό εντοπίζει τα αγαθά (assets) και μεταφράζει σε cpe. Σε δεύτερο χρόνο το ARAT υπολογίζει την επικινδυνότητα του κάθε CPE και βρίσκει για ποια CVE υπάρχει η μέγιστη τιμή. Σε τρίτο χρόνο το λογισμικό patch manager εντοπίζει τις απαραίτητες

ενημερώσεις που πρέπει να γίνουν και τις εκτελεί. Τέλος το ARAT. ξανά εκτελείται με τις αλλαγές που έχουν εκτελεσθεί και εισέρχεται σε νέο κύκλο βελτιστοποίησης.

Κλείνοντας, όπως φαίνεται από τη διενεργηθείσα μεταπτυχιακή διατριβή, το λογισμικό ARAT μπορεί δυνητικά να αποτελέσει ένα σημαντικό εργαλείο αντιμετώπισης διαδικτυακών απειλών, καθώς όχι μόνο υπολογίζεται το συνολικό επίπεδο επικινδυνότητας ενός υπολογιστικού συστήματος, αλλά εντοπίζει τα πλέον ευπαθή του σημεία, ώστε να μπορεί ο διαχειριστής του συστήματος να εξετάσει την αντικατάστασή τους, με στόχο την μείωση των επιπέδων επικινδυνότητας όλου του συστήματος.

Κεφάλαιο 6
Επίλογος

Στην παρούσα μεταπτυχιακή διατριβή εξετάστηκε το ζήτημα της επικινδυνότητας υπολογιστικών συστημάτων με αυτοματοποιημένο τρόπο. Αφού μελετήθηκε η σχετική βιβλιογραφία, αναπτύχθηκε το λογισμικό ARAT, το οποίο υπολογίζει τα επίπεδα επικινδυνότητας υπολογιστικών συστημάτων. Στη συνέχεια έγινε δοκιμή εφαρμογής του σε ένα τυχαίο σύστημα, από την οποία δοκιμή εξήχθησαν ενδιαφέροντα αποτελέσματα.

Το λογισμικό ARAT δεν πρέπει να αντιμετωπιστεί απλά ως ένα λογισμικό αυτοματοποιημένης εκτίμησης επικινδυνότητας προσανατολισμένο στο αποτέλεσμα, καθώς το αποτέλεσμα αυτό καθαυτό είναι λιγότερο σημαντικό από την ερμηνεία των ενδιάμεσων αποτελεσμάτων. Μπορεί επομένως το ARAT να εξελιχθεί σε ένα σημαντικό εργαλείο στα χέρια των διαχειριστών υπολογιστικών συστημάτων, οι οποίοι με τη χρήση του έχουν τη δυνατότητα να εντοπίζουν τα πλέον ευπαθή τμήματα των συστημάτων που διαχειρίζονται. Στη συνέχεια μπορούν να

προβαίνουν σε κατάλληλες ενέργειες ώστε να αντιμετωπίζουν τις εντοπισμένες ευπάθειες με τρόπο τέτοιο που να οδηγεί σε πιο ασφαλή συστήματα.

Για να γίνει το ARAT πιο προσιτό στο ευρύ κοινό, θα ήταν επιθυμητό να γίνουν ορισμένες βελτιώσεις. Μια πρώτη βελτίωση είναι η ανάπτυξή του σε περιβάλλον web interface, ώστε να είναι πιο εύκολα προσβάσιμο. Επίσης, θα πρέπει τα αποτελέσματα να δίνονται με πιο εποπτικό τρόπο, ώστε να εντοπίζονται εύκολα και με ακρίβεια τα ευπαθή σημεία. Τέλος, το λογισμικό ARAT μπορεί να προτείνει πιθανές εναλλακτικές επιλογές, με σκοπό τη βελτίωση των επιπέδων επικινδυνότητας του μελετώμενου υπολογιστικού συστήματος.

Βιβλιογραφία

Dimitriadis Athanasios [et al.] ARES: Automated Risk Estimation in Smart Sensor Environments [Article] // Sensors. - 2020. - Vol. 20.

Dimitriadis Athanasios Leveraging digital forensics and information sharing into prevention, incident response, and investigation of cyber threats (PhD Thesis). - Thessaloniki : [s.n.], 2022.

FIRST CVSS - Common Vulnerability Scoring System [Online]. - https://www.first.org/cvss/.

Forward Andrew and Lethbridge Timothy C. A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering [Conference] // Proceedings of the 2008 conference of the Centre for Advanced Studies on Collaborative Research, October 27-30, 2008,. - Richmond Hill, Orlando, Canada : [s.n.], 2008. - pp. 179-191.

Jakoubi S., Tjoa S. and Quirchmayr G. ROPE: A Methodology for Enabling the Risk-aware Modeling and Simulation of Business Processes [Conference] // Proceedings of Fifteenth European Conference on Information Systems. - St. Gallen, Switzerland : [s.n.], 7-9 June 2007. - pp. 1596-1607.

Joint Task Force Transformation Initiative Guide for Conducting Risk Assessments. - Gaithersburg, MD, USA : National Institute of Standards and Technology, 2012. - pp. p. NIST SP 800-30rl.

MITRE [Online] // MITRE Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description. - January 15, 2008. - https://capec.mitre.org/documents/documentation/CAPEC_Schema_Description_v1.3.pdf.

MITRE CAPEC - Common Attack Pattern Enumeration and Classification (CAPEC) [Online]. - https://capec.mitre.org.

MITRE CPE - Common Platform Enumeration [Online]. - https://cpe.mitre.org/.

MITRE CVE - Common Vulnerability Enumeration [Online]. - https://cve.mitre.org/.

MITRE CWE - Common Weakness Enumeration [Online]. - https://cwe.mitre.org/.

MITRE MITRE [Online]. - 05 05, 2022. - https://www.mitre.org/capabilities/cybersecurity/overview/cybersecurity-resources/standards.

NIST National Vulnerability Database NVD - Vulnerabilities [Online]. - https://nvd.nist.gov/vuln/full-listing.

Parmelle M.C. [et al.] Common Platform Enumeration: Name Matching Specification Version 2.3. - Gaithersburg, MD, USA : National Institute of Standards and Technology, 2011. - p. 28.

Rongrat K. and Senivongse T. Assessing Risk of Security Non-compliance of Banking Security Requirements Based on Attack Patterns [Article] // International Journal of Networked and Distributed Computing. - 2017. - 1 : Vol. 6. - pp. 1-10.

Schatz Daniel, Bashroush Rabih and Wall Julie Towards a Mor ds a More Representativ esentative Definition of Cyber Security e Definition of Cyber Security [Article] // Journal of Digital Forensics, Security and Law. - 2017. - 2 : Vol. 12. - 8.

Tjoa S., Jakoubi S. and Quirchmayr G. Enhancing Business Impact Analysis and Risk Assessment Applyiing a Risk-Aware Business Process Modeling and SImulation Methodology [Conference] // Proceedins of the 2008 Third International Conference on Availability, Reliability and Security. - Barcelona, Spain : [s.n.], 4-7 March 2008. - pp. 179-786.

Παράρτημα Α
Κώδικάς Λογισμικού

Α.1 Λογισμικό ARAT

Στο παρακάτω παρουσιάζεται ο κώδικας του λογισμικού που αναπτύχθηκε, χωρίς τα υποστηρικτικά αρχεία που απαιτούνται για την λειτουργία του

```
import os
cmd = 'rm -rf cpe??.txt'
cmd1 = 'rm -rf cpe??.csv'
os.system(cmd)
os.system(cmd1)
import requests
import pandas as pd
import xml.etree.ElementTree as ET
from urllib.request import urlopen
from io import BytesIO
from zipfile import ZipFile

zip_url = ' https://capec.mitre.org/data/xml/views/1000.xml.zip'
with urlopen(zip_url) as f:
    with BytesIO(f.read()) as b, ZipFile(b) as myzipfile:
        foofile = myzipfile.open('1000.xml')
        myzipfile.extractall()
file = ET.parse('1000.xml').getroot()
root = ET.parse('capec2cpe.xml').getroot()
AG = root.findall('AttackGroup')
dat = 0
total = 0
trisk = 0
klik = 0
input_data = []
lista=[]
listat=[[]]
lista0=[]
lista1=[]
lista2=[]
lista3=[]
lista4=[]
lista5=[]
lista6=[]
lista7=[]
lista8=[]
lista9=[]
lista10=[]
lista11=[]
lista12=[]
lista13=[]
lista14=[]
filename2="cpe.csv"
cwe=[]
cvss=[]
yz=0
df1 = pd.read_csv('CPEs.csv',delimiter=';')
cpes_list = [list(row)for row in df1.values]
df2 = pd.read_csv('CAPECs.csv',delimiter=';')
capecs_list = [list(row)for row in df2.values]
for ag1 in range(0,len(AG)):
 for ag2 in range(0,len(AG[ag1])):
  for ag3 in range(0,len(AG[ag1][ag2])):
   if ag1==0:
     lista0.append(AG[ag1][ag2][ag3].text)
   elif ag1==1:
     lista1.append(AG[ag1][ag2][ag3].text)
   elif ag1==2:
     lista2.append(AG[ag1][ag2][ag3].text)
   elif ag1==3:
     lista3.append(AG[ag1][ag2][ag3].text)
   elif ag1==4:
     lista4.append(AG[ag1][ag2][ag3].text)
   elif ag1==5:
     lista5.append(AG[ag1][ag2][ag3].text)
   elif ag1==6:
     lista6.append(AG[ag1][ag2][ag3].text)
   elif ag1==7:
     lista7.append(AG[ag1][ag2][ag3].text)
   elif ag1==8:
     lista8.append(AG[ag1][ag2][ag3].text)
```

```
    elif ag1==9:
      lista9.append(AG[ag1][ag2][ag3].text)
    elif ag1==10:
      lista10.append(AG[ag1][ag2][ag3].text)
    elif ag1==11:
      lista11.append(AG[ag1][ag2][ag3].text)
    elif ag1==12:
      lista12.append(AG[ag1][ag2][ag3].text)
    elif ag1==13:
      lista13.append(AG[ag1][ag2][ag3].text)
    elif ag1==14:
      lista14.append(AG[ag1][ag2][ag3].text)
listat.append(lista0)
listat.append(lista1)
listat.append(lista2)
listat.append(lista3)
listat.append(lista4)
listat.append(lista5)
listat.append(lista6)
listat.append(lista7)
listat.append(lista8)
listat.append(lista9)
listat.append(lista10)
listat.append(lista11)
listat.append(lista12)
listat.append(lista13)
listat.append(lista14)
print("------------------------------------------------------------")
k = input("Please Enter The number of The CPES\n")
for ik in range(0,int(k)):
      g = input("Please Enter The Next CPE\n")
      input_data.append(g)
      print("------------------------------------------------------------")
with open(filename2, 'w')as g:
 for i in range(0, int(k)):
      lines=[]
      risk1=[]
      lines3=[]
      tank1=[]
      print("CPE Number", i + 1)
      print(input_data[i])
      print("------------------------------------------------------------")
      url = "https://services.nvd.nist.gov/rest/json/cpes/1.0?cpeMatchString=" +
input_data[i] + "&addOns=cves&includeDeprecated=true"
      response = requests.get(url)
      j = 0
      for j in range(len(response.json()["result"]["cpes"][0]["vulnerabilities"])):
       data = response.json()["result"]["cpes"][0]["vulnerabilities"][j]
       url2 = "https://services.nvd.nist.gov/rest/json/cves/1.0?keyword=" + data
       response2 = requests.get(url2)
       if response2.status_code == 200:
         print(data)
         try:
            data2 =
response2.json()["result"]["CVE_Items"][0]["impact"]["baseMetricV3"]["cvssV3"]["base
Score"]
            print("CVSS Version V3")
         except KeyError:
            data2 =
response2.json()["result"]["CVE_Items"][0]["impact"]["baseMetricV2"]["cvssV2"]["base
Score"]
            print("CVSS Version V2")
         data3 =
response2.json()["result"]["CVE_Items"][0]["cve"]["problemtype"]["problemtype_data"]
[0]["description"][0]["value"]
         data3_ = data3.replace('CWE-','')
         capec=[]
         print("CVSS = ",data2)
         print(data3)
         print("------------------------------------------------------------")
         if data3[0]=="C":
```

```
            url3 = "https://cve.circl.lu/api/capec/" + data3_
            response3 = requests.get(url3)
            for k in range(len(response3.json())):
             data5 = response3.json()[k]["id"]
             if data5 != 0:
              data6 = 'CAPEC-' + data5
              capec.append(data6)
             else:
              print(data2,data6)
         else:
            print("Error 403 " + data)
            print()
        capec1=set(capec)
        capec.clear()
        capec2=list(dict.fromkeys(capec1))
        print("CAPEC",capec2)
        for ic in range(0,len(capec2)):
          for id in range(1,15):
           for ie in range(0,len(listat[id])):
            if capec2[ic]==listat[id][ie]:
             for ig in range(0,len(listat[id])):
              if listat[id][ig]==input_data[i]:
               lista.append(capec2[ic])
        capec2.clear()
        listak=set(lista)
        listal=list(dict.fromkeys(listak))
        lines.extend(listal)
        lines1=set(lines)
        lista.clear()
        listak.clear()
        listal.clear()
        lines2=list(dict.fromkeys(lines1))
        lines3.extend(lines2)
        for est1 in range(0,len(cpes_list)):
         for est2 in range(0,len(cpes_list[est1])):
          if input_data[i]==cpes_list[est1][est2]:
           print(est1,est2)
        for ist1 in range(0,len(capecs_list)):
         for ist2 in range(0,len(capecs_list[ist1])):
          for ist3 in range(0,len(lines3)):
           if capecs_list[ist1][ist2]==lines3[ist3]:
            print(capecs_list[ist1][1])


        tr1=0
        klak = 0
        name = "cpe"+str(i+1)
        filename = "%s.txt"%name
        with open(filename, 'w') as f:
         for line in lines3:
          f.write(line)
          f.write('\n')
          for q in range(0,len(file[0])):
            tank = "CAPEC-" + file[0][q].get('ID')
            if tank == line:
             if file[0][q].findall('.//{http://capec.mitre.org/capec-
3}Likelihood_Of_Attack') is not None:

                                    klak = klak + 1
            else:
                                     print("error")
            if file[0][q].find('.//{http://capec.mitre.org/capec-3}Typical_Severity')
is not None:
             klik = klik+1
             if file[0][q].find('.//{http://capec.mitre.org/capec-
3}Likelihood_Of_Attack') is not None:
              likelihood = file[0][q].find('.//{http://capec.mitre.org/capec-
3}Likelihood_Of_Attack').text
              print("------------------------------------------------------------")
              if file[0][q].find('.//{http://capec.mitre.org/capec-
3}Typical_Severity') is not None:
```

```
                severity = file[0][q].find(".//{http://capec.mitre.org/capec-
3}Typical_Severity").text
                if severity=="Low":
                  print(tank)
                  tank1.append(tank)
                  print("Severity Low")
                  sev=1
                  if likelihood=="Low":
                   print("Likelihood Low")
                   like=1
                  elif likelihood=="Medium":
                   print("Likelihood Medium")
                   like=2
                  elif likelihood=="High":
                   print("Likelihood High")
                   like=3
                  elif likelihood=="Very High":
                   print("Likelihood Very High")
                   like=4

                elif severity=="Medium":
                  print(tank)
                  tank1.append(tank)
                  print("Severity Medium")
                  sev=2
                  if likelihood=="Low":
                   print("Likelihood Low")
                   like=1
                  elif likelihood=="Medium":
                   print("Likelihood Medium")
                   like=2
                  elif likelihood=="High":
                   print("Likelihood High")
                   like=3
                  elif likelihood=="Very High":
                   print("Likelihood Very High")
                   like=4

                elif severity=="High":
                  print(tank)
                  tank1.append(tank)
                  print("Severity High")
                  sev=3
                  if likelihood=="Low":
                   print("Likelihood Low")
                   like=1
                  elif likelihood=="Medium":
                   print("Likelihood Medium")
                   like=2
                  elif likelihood=="High":
                   print("Likelihood High")
                   like=3
                  elif likelihood=="Very High":
                   print("Likelihood Very High")
                   like=4
                elif severity=="Very High":
                  print(tank)
                  tank1.append(tank)
                  print("Severity Very High")
                  sev=4
                  if likelihood=="Low":
                   print("Likelihood Low")
                   like=1
                  elif likelihood=="Medium":
                   print("Likelihood Medium")
                   like=2
                  elif likelihood=="High":
                   print("Likelihood High")
                   like=3
                  elif likelihood=="Very High":
                   print("Likelihood Very High")
```

```
                like=4
            if likelihood == "High" and severity == "High":
                                                    risk = "High"
                                                    r1=3
            elif likelihood == "Low" and severity == "Low":
                                                    risk = "Low"
                                                    r1=1
            elif likelihood == "Medium" and severity == "Medium":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Low" and severity == "Medium":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Low" and severity == "High":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Medium" and severity == "Very Low":
                                                    risk = "Low"
                                                    r1=1
            elif likelihood == "Medium" and severity == "Low":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Medium" and severity == "High":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Medium" and severity == "Very High":
                                                    risk = "High"
                                                    r1=3
            elif likelihood == "High" and severity == "Very Low":
                                                    risk = "Low"
                                                    r1=1
            elif likelihood == "High" and severity == "Low":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "High" and severity == "Medium":
                                                    risk = "High"
                                                    r1=3
            elif likelihood == "High" and severity == "Very High":
                                                    risk = "Very High"
                                                    r1=4
            elif likelihood == "Medium" and severity == "Low":
                                                    risk = "Medium"
                                                    r1=2
            elif likelihood == "Low" and severity == "Very High":
                                                    risk = "Medium"
                                                    r1=2

            tr1=r1
            print("Risk",risk)
            risk1.append(risk)
            name1 = "cpe"+str(i+1)
            filename1 = "%s.csv"%name1
        if trisk<tr1:
            trisk = tr1
print("-------------------------------------------------------------")
if trisk==1:
  print("CPE Risk Low")
  g.write(input_data[i])
  g.write(',')
  g.write("CPE Risk  Low")
  g.write("\n")
elif trisk==2:
  print("CPE Risk Medium")
  g.write(input_data[i])
  g.write(',')
  g.write("CPE Risk Medium")
  g.write("\n")
elif trisk==3:
  print("CPE Risk High")
  g.write(input_data[i])
  g.write(',')
  g.write("CPE Risk High")
```

```
            g.write("\n")
          elif trisk==4:
            print("CPE Risk Very High")
            g.write(input_data[i])
            g.write(',')
            g.write("CPE Risk Very High")
            g.write("\n")
          print("-------------------------------------------------------------")
          if total <trisk:
            total = trisk
            print(tank1,risk1)
          with open(filename1, 'w') as f:
            for tank2 in range(0,len(tank1)):
             f.write(tank1[tank2])
             f.write(',')
             f.write(risk1[tank2])
             f.write('\n')
if total==1:
          print("Total Risk Low")
elif total==2:
          print("Total Risk Medium")
elif total==3:
          print("Total Risk High")
elif total==4:
          print("Total Risk Very High")
```