Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Οντολογίας Ασφάλειας για την Ανάλυση και Διαχείριση Επικινδυνότητας IoT Συσκευών και Συστημάτων Κρίσιμων Υποδομών - Development of a Cybersecurity Ontology for the Analysis and Management or Risk for IoT and Critical Infrastructure Systems
Ονοματεπώνυμο Φοιτητή	ΑΔΑΜΑΝΤΙΟΣ – ΜΑΡΙΟΣ ΜΠΕΡΖΟΒΙΤΗΣ
Πατρώνυμο	ΑΡΙΣΤΕΙΔΗΣ
Αριθμός Μητρώου	ΜΠΚΣΑ19017
Επιβλέπων	ΠΑΝΑΓΙΩΤΗΣ ΚΟΤΖΑΝΙΚΟΛΑΟΥ, ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης - Ιούνιος 2022

Τριμελής Εξεταστική Επιτροπή

Παναγιώτης Κοτζανικολάου Αναπληρωτής Καθηγητής	Δημήτριος Αποστόλου Καθηγητής	Μιχαήλ Ψαράκης Αναπληρωτής Καθηγητής

Θέλω να ευχαριστήσω πολύ
όσους με στήριξαν για την εκπόνηση της μεταπτυχιακής διατριβής
και την ολοκλήρωση του Προγράμματος Μεταπτυχιακών Σπουδών

Περίληψη

Οι σύγχρονοι οργανισμοί και επιχειρήσεις, επεκτείνουν διαρκώς τα πληροφοριακά τους συστήματα τόσο σε ψηφιακό όσο και σε φυσικό επίπεδο. Παρατηρούνται μάλιστα, αλλεπάλληλες προσθήκες σε εξοπλισμό, που εκτείνονται από καθημερινούς σταθμούς εργασίας και δικτυακό εξοπλισμό έως Internet of Things (IOT) συσκευές και Supervisory Control and Data Acquisition (SCADA) συστήματα, σύμφωνα πάντα με τις ανάγκες των οργανισμών και επιχειρήσεων. Από τη σκοπιά της κυβερνοασφάλειας, οι διαρκείς αυτές αλλαγές, οξύνουν την ανάγκη στοιχειοθέτησης του μεγάλου και πολύπλοκου όγκου πληροφορίας, καθώς και την αποδοτική ανάλυση της, για την εξαγωγή των κατάλληλων συμπερασμάτων. Ο στόχος της εργασίας αυτής, είναι η ανάπτυξη μιας μεθοδολογίας που θα συμβάλει σε αυτήν την κατεύθυνση, με την υποβοήθηση ενός εργαλείου ανοιχτού κώδικα που υποστηρίζει την αναπαράσταση δημόσιων καταλόγων και βάσεων δεδομένων αδυναμιών όπως: Common Vulnerabilities and Exposures (CVE), Common Platform Enumeration (CPE) τα οποία καταγράφονται και ανανεώνονται από το National Institute of Technology (NIST) και οι κατάλογοι Common Weakness Enumeration (CWE) και Common Attack Pattern Enumeration and Classification (CAPEC) που παρέχονται από το MITRE, σε μορφή διασυνδεδεμένου γράφου. Στόχος αυτής της υλοποίησης αποτελεί η στήριξη υπαρχόντων μεθοδολογιών ανάλυσης και διαχείρισης επικινδυνότητας, μέσω της παροχής ενός εκτεταμένου χάρτη πληροφορίας που μπορεί να παρέχει αυτοματοποίηση στις φάσεις της αναγνώρισης απειλών και αδυναμιών ασφαλείας σε ένα πληροφοριακό σύστημα. Παράλληλα, παρουσιάζονται μελέτες περίπτωσης για την εξαγωγή πολύπλοκης πληροφορίας, σχετικά με υπάρχοντα προφίλ κακόβουλων χρηστών που παρέχει η βιβλιοθήκη Threat Agent Library της Intel.

Abstract

Modern companies and organizations continuously upgrade their informational systems in the digital and the physical level. These upgrades vary -depending on the needs of organizations-, ranging from daily used workstations to network equipment, Internet of Things (IOT) devices and Supervisory Control and Data Acquisition (SCADA) systems. From the cybersecurity perspective, these lasting changes, exacerbate the need for efficient construction, analysis, and knowledge extraction, derived from a vast amount of complex data. The goal of this thesis is the development of a methodology which can contribute to this direction, supported by an open-source cybersecurity tool that represents an interconnected graph, which contains vulnerability and asset databases such as Common Vulnerabilities and Exposures (CVE), Common Platform Enumeration (CPE) which are recorded and updated by the National Institute of Technology (NIST) and catalogues such as Common Weakness Enumeration (CWE) and Common Attack Pattern Enumeration and Classification (CAPEC) which are provided by MITRE. The goal of this implementation is to support existing risk assessment methodologies, by providing a useful and large information map which can automate risk assessment's phases such as the threat and vulnerability reconnaissance of an informational system. To validate the use of this tool and methodology, we demonstrate use case scenarios, we extract additional connections and unveil additional relationships. This way, we extend existing knowledge of known security catalogues and databases and create an initial methodology to connect this static information, to temporal objects like the malicious user profiles provided by Intel's Threat Agent Library.

Πίνακας Περιεχομένων

1 Εισαγωγή

Στον 21ο αιώνα η επιστήμη και η τεχνολογία παρέχουν σύγχρονες λύσεις που οδηγούν σε συχνές προσθήκες και επεκτάσεις πληροφοριακών συστημάτων, η πολυπλοκότητα και η δια-συνδεσιμότητα των οποίων αυξάνεται εκθετικά. Η αύξηση αυτή, επηρεάζει ιδιαίτερα ψηφιακά και φυσικά συστήματα των οποίων η άνοδος, οφείλεται σε μεγάλο βαθμό στην είσοδο των IOT (Διαδίκτυο των Πραγμάτων) συσκευών που χρησιμοποιούνται σε επαγγελματικούς και προσωπικούς χώρους. Στην εποχή των δεδομένων, η ερευνητική κοινότητα της κυβερνοασφάλειας αναγνωρίζει τις αλλαγές αυτές, ως πιθανές πηγές κινδύνου και επιδιώκει ενεργά την δημιουργία ενός πλαισίου αποδοτικής αποθήκευσης και μελέτης σχετικών πληροφοριών.

Το εύρος έρευνας και εφαρμογής της κυβερνοασφάλειας, ποικίλει ανάλογα με τις ανάγκες των επιστημόνων και των οργανισμών. Υπάρχουν ωστόσο τρία βασικά σημεία που καλείται η επιστήμη αυτή να ελέγξει και εν τέλει να προστατέψει αποδοτικά.

Κάθε πληροφοριακό σύστημα ενός οργανισμού, ψηφιακό (δηλαδή άυλο) και φυσικό (δηλαδή με υλική υπόσταση) μπορεί να έχει μικρότερη ή μεγαλύτερη αξία για τον οργανισμό αυτό. Έτσι, το μελετάμε ως αγαθό προς προστασία, σε σχέση με την εμπιστευτικότητα, δηλαδή την μη εξουσιοδοτημένη πρόσβαση σε αυτό, την ακεραιότητα, δηλαδή την μη εξουσιοδοτημένη μορφοποίηση, επεξεργασία, και οποιαδήποτε αλλαγή φυσική και ψηφιακή αυτού καθώς και την διαθεσιμότητά του, δηλαδή την εξουσιοδοτημένη σε αυτό πρόσβαση ανά πάσα ώρα και στιγμή.

Αν αναλογιστούμε δε, το πλήθος των αγαθών σε έναν οργανισμό, φυσικών, ψηφιακών, που ανήκουν ή μη στο φάσμα των δια-συνδεδεμένων συσκευών (IOT), θα καταλάβουμε τη σημασία της ύπαρξης ενός αποδοτικού πλαισίου για την προστασία τους, με καθορισμένες πολιτικές, δηλαδή κανόνες, γενικές αρχές, δεσμεύσεις κοκ. για την επίτευξη του στόχου της ασφάλειας καθώς και καθορισμένες διαδικασίες, δηλαδή αυστηρά καθορισμένα βήματα για την επίτευξη των πολιτικών και τη συμμόρφωση του οργανισμού σε αυτές.

Με μια τέτοια λογική και ανάγκη, οι επιστήμονες της κυβερνοασφάλειας, οι δημόσιοι και ιδιωτικοί οργανισμοί, ανέπτυξαν λεπτομερείς πολιτικές και διαδικασίες για τη διαχείριση των απειλών -των περιστάσεων και ενεργειών δηλαδή που μπορούν να επηρεάσουν δυσμενώς τον οργανισμό- και των ευπαθειών και αδυναμιών που μπορεί να υπάρχουν στα αγαθά αλλά ακόμη και σε πολιτικές, διαδικασίες και εσωτερικές λειτουργίες ή ελέγχους του οργανισμού. Εν τέλει στόχο έχουν να μπορέσουν να αναλύσουν και να προστατευθούν από τους διάφορους κινδύνους που στην ουσία αποτελούν τον συνδυασμό των απειλών, των αδυναμιών αλλά και των επιπτώσεων που η εκμετάλλευση τους μπορεί να έχει για τον οργανισμό.

Έτσι γεννήθηκε η ανάλυση και η διαχείριση της επικινδυνότητας για τα πληροφοριακά συστήματα ώστε πέρα από τον εντοπισμό των σημείων που χρήζουν προσοχής, να καθοριστεί ένα πλαίσιο αποφάσεων για την βελτίωση της ψηφιακής ασφάλειας του οργανισμού. Η διαχείριση κινδύνων επιτρέπει την επιλογή των κατάλληλων μέτρων προστασίας με 1) αντιμετώπιση του κινδύνου 2) αποδοχή του κινδύνου 3) μεταφορά του κινδύνου.

Σαφώς και στην εποχή των γρήγορων εξελίξεων και αλλαγών, η ανάλυση και διαχείρισης της επικινδυνότητας παρότι αναγκαία δεν επαρκεί με τις κλασσικές μεθόδους της. Χρειάζεται η περαιτέρω εξέλιξή της, με ολοένα και πιο αυτοματοποιημένο αλλά ταυτόχρονα εμπεριστατωμένο τρόπο. Το κλειδί είναι η αξιοποίηση της δημόσιας πληροφορίας, και η ενορχήστρωσή της στις υπάρχουσες μεθόδους μας.

Στο διαδίκτυο υπάρχουν εκατομμύρια τέτοιες πληροφορίες οι οποίες βρίσκονται σε ανοιχτές πηγές σε δομημένη (με οργανωμένους πίνακες, κατηγορίες κοκ.), ημιδομημένη (με εν μέρει την προαναφερθείσα οργάνωση) και μη δομημένη (δηλαδή όπως τα εξάγουμε από

συσκευές, μοντέλα δεδομένων κοκ.) μορφή. Κάθε μορφή πληροφοριών παρέχει διαφορετικές προκλήσεις από την ανάγνωση μέχρι την κατεργασία, την δημιουργία καινούργιων συσχετίσεων και την αποθήκευση τους.

Γνωστοί οργανισμοί όπως ο NIST και ο MITRE διατηρούν το National Vulnerability Database (NVD) και διάφορους καταλόγους αδυναμιών όπως το Common Platform Enumeration (CPE) για την κατηγοριοποίηση συστημάτων τεχνολογίας, λογισμικού, υλισμικού και λειτουργικών συστημάτων το οποίο βασίζεται σε γενικευμένα -συντακτικά- αναγνωριστικά (Uniform Resource Identifiers -URI-) [1] [2]. Το Common Weakness Enumeration (CWE) για την κατηγοριοποίηση των ευπαθειών βάσει συγκεκριμένων χαρακτηριστικών., και το Common Attack Pattern Enumeration and Classification (CAPEC) για την ταξινόμηση γνωστών επιθέσεων που μπορούν να χρησιμοποιηθούν από κακόβουλους ή εξουσιοδοτημένους χρήστες για να εκμεταλλευτούν τις ευπάθειες ενός πληροφοριακού συστήματος με σκοπό είτε κακόβουλες ενέργειες είτε την ενίσχυση της ψηφιακής άμυνας του Π.Σ. και πίνακες αδυναμιών όπως το Adversary Tactics Techniques and Common Knowledge (ATT&CK) και το D3FEND [3] [4] [5]. Τέλος, διατηρούν το Common Vulnerabilities and Exposures (CVE) [6] που είναι το δημόσιο σύστημα καταγραφών ευπαθειών ψηφιακής ασφάλειας, ενώ κάθε εγγραφή του, έχει μοναδικό αναγνωριστικό που του δίνεται από πιστοποιημένους φορείς (CNAs).

Οι παραπάνω βάσεις δεδομένων, κατάλογοι και πίνακες ακολουθούν συγκεκριμένα σχήματα που τις οριοθετεί στις δομημένες πηγές δεδομένων. Ωστόσο, ενώ εμπεριέχουν τεράστιο όγκο πληροφορίας παρατηρείται ιδιαίτερη έλλειψη πληρότητας και διασυνδεσιμότητας μεταξύ τους.

Τα προβλήματα αυτά μπορούν να λυθούν μέσω της συνδεσιμότητας και της εξαγωγής γνώσεων. Κάθε επιστήμονας ή ερευνητής προσπαθεί να βρει συνδέσεις μεταξύ των όσων μελετά και προσπαθεί να εξάγει τα κατάλληλα συμπεράσματα, ώστε να τον/την οδηγήσουν στο επόμενο βήμα, όπου θα ακολουθήσει εκ νέου την ίδια λογική. Η μεθοδολογία αυτή είναι αέναη, και έχει συντελέσει την εξέλιξη όλων των επιστημών. Μια από τις σύγχρονες λύσεις που αναπτύχθηκαν πάνω σε αυτή τη μεθοδολογία, είναι οι βάσεις δεδομένων υπό τη μορφή γράφων (ΒΔΓ), πάνω στις οποίες παρουσιάζεται ιδιαίτερη δημοτικότητα τελευταία στο χώρο της κυβερνοασφάλειας [7].

Η ΒΔΓ είναι Βάση Δεδομένων που έχει σχεδιαστεί με στόχο να χειρίζεται σχέσεις μεταξύ δεδομένων με την ίδια σημασία που χειρίζεται τα δεδομένα αυτά καθ' αυτά. Επιδιώκει να σχηματοποιεί τα δεδομένα με μοναδικό τρόπο υπό τη μορφή γράφου (κόμβοι και ακμές) χωρίς δηλαδή αυτά να υπόκεινται σε μια προκαθορισμένη μοντελοποίηση [8] [9].

Όπως αναφέρθηκε παραπάνω τα δομημένα ανοιχτά δεδομένα που παρέχουν ο NIST και ο MITRE παρουσιάζουν ορισμένες ελλείψεις και σε πλαίσιο εγγραφών και σε πλαίσιο διασυνδεσιμότητας των διαφόρων συλλογών δεδομένων που διαθέτουν. Λόγω του πολύ εκτενούς σχήματος που τηρούν αυτές οι συλλογές είναι δύσκολη η καταγραφή όλων των διαθέσιμων μεταβλητών για κάθε εγγραφή. Ένα ανοιχτό πρόβλημα αυτή τη στιγμή λοιπόν αποτελεί το πως μπορεί να χρησιμοποιηθεί το πλήρες κομμάτι της συλλογής για την δημιουργία ενός αυτόματου συστήματος συμπλήρωσης των ελλειπών εγγραφών. Ένα δεύτερο πρόβλημα που προκύπτει είναι η δημιουργία περισσότερων διασυνδέσεων μεταξύ των διαφορετικών συλλογών.

Η λύση του παραπάνω προβλήματος αποτελείται από τρία σκέλη:

- Λήψη τοπικού αντιγράφου δεδομένων: Οι πηγές δεδομένων ανανεώνονται καθημερινά οπότε είναι απαραίτητη η δημιουργία ενός μηχανισμού λήψης όλων των πηγών σε σταθερά διαστήματα για την κατάλληλη διατήρηση του τοπικού αντιγράφου. Για την επίτευξη του στόχου αυτού ενδείκνυται η χρήση μεθόδων parsing, δηλαδή ελέγχου των ψηφιακών σημείων που υπάρχουν τα δεδομένα για αλλαγές και ενημερώσεις με σκοπό την αποδοτική τους λήψη, και η χρήση των διάφορων API, δηλαδή ειδικών διεπαφών που επιτρέπουν στα προγράμματα να μιλούν μεταξύ τους, να συνδέονται και να ανταλλάσσουν δεδομένα.
- Αναγνώριση των ελλείψεων σε εγγραφές και διασυνδέσεις: Για κάθε εγγραφή κάθε βάσης, καταλόγου ή πίνακα πρέπει να πραγματοποιηθεί έλεγχος σχετικά με την πληρότητα των συμπληρωμένων μεταβλητών σε σχέση με τις προκαθορισμένες

μεταβλητές του ανάλογου σχήματος. Επίσης οι διάφορες συλλογές έχουν καταγεγραμμένες διασυνδέσεις στις εγγραφές τους οι οποίες είναι ελλιπείς. Για την επίτευξη του στόχου αυτού, μπορεί να χρησιμοποιηθεί στατιστική ανάλυση, αλλά και αλγόριθμος που μπορεί να μαθαίνει από συλλογές δεδομένων και να κάνει προβλέψεις για τις ελλείψεις σε νέα δεδομένα, δηλαδή ένας αλγόριθμος μηχανικής μάθησης

- Δημιουργία καινούργιων διασυνδέσεων: Με την ανάλυση των εγγραφών και την αναγνώριση των όποιων ελλείψεων τους, προχωράμε στον εκ νέου σχεδιασμό και την μοντελοποίηση του τρόπου διασύνδεσης των δεδομένων – πληροφοριών, και στην υλοποίηση του στις υπάρχουσες εγγραφές. Θέτουμε σαν στόχο την πληρέστερη αναπαράσταση και διασυνδεσιμότητα των εγγραφών μας, ώστε η εξαγωγή των γνώσεων να είναι η βέλτιστη δυνατή για τους χρήστες και κατ' επέκταση η χρήση των εξαγώμενων πληροφοριών και γνώσεων, να είναι η αποδοτικότερη σε ένα συνολικό πλαίσιο Ανάλυσης και Διαχείρισης Επικινδυνότητας ενός πληροφοριακού συστήματος. Για παράδειγμα μέσω της αξιοποίησης των ΒΔΓ και της θεωρίας γράφων, της μελέτης και των σχέσεων τους και την δημιουργία νέων τρόπων αναπαράστασης των πρωτογενών δεδομένων μέσω των γράφων.

1.1 Παραδοτέα της Εργασίας

Στην ενότητα αυτή αναφέρονται συνοπτικά τα παραδοτέα της διπλωματικής εργασίας, δηλαδή όλο το σχετικό υλικό που συμπεριλαμβάνεται και παραδίδεται στο πανεπιστήμιο και την αρμόδια επιτροπή εξέτασης. Πιο συγκεκριμένα:

- Το παρόν κείμενο της πτυχιακής εργασίας, το οποίο περιλαμβάνει, την μελέτη και γραπτή αποτύπωση της εργασίας, τα αποτελέσματα και συμπεράσματά της, και την σχετική βιβλιογραφία.
- Το λογισμικό «GraphKer». Όλα τα απαραίτητα στοιχεία λογισμικού, για την εκτέλεση του «GraphKer» και το σχετικό βοηθητικό έγγραφο περιγραφής της εγκατάστασης και παραμετροποίησης των απαιτούμενων βοηθητικών λογισμικών και της εκτέλεσης του «GraphKer».
- Την παρουσίαση της διπλωματικής εργασίας σε μορφή PowerPoint Presentation, που εμπεριέχει σύντομη παρουσίαση της μελέτης και οπτικό υλικό με φωτογραφίες και βίντεο παραδειγμάτων εκτέλεσης του λογισμικού «GraphKer».
- Το πρωτότυπο υλικό των βίντεο που θα αξιοποιηθούν κατά την παρουσίαση αλλά και που καταγράφηκαν για παραδείγματα εκτέλεσης του λογισμικού «GraphKer».

1.2 Δομή της Εργασίας

Η παρούσα διπλωματική εργασία χωρίζεται σε επτά βασικές ενότητες. Την Εισαγωγή όπου τίθεται το πρόβλημα υπό μελέτη και οι στόχοι της παρούσας εργασίας. Στη συνέχεια ακολουθεί η «επισκόπηση του χώρου» όπου γίνεται μια μελέτη βάσεων δεδομένων αδυναμιών, γενικότερων χρήσεων των βάσεων δεδομένων γράφων και ειδικών χρήσεων τους εφαρμοσμένες πάνω στην κυβερνοασφάλεια. Το επόμενο κεφάλαιο που ακολουθεί ονομάζεται «Τεχνολογίες, Πηγές Δεδομένων και Αρχιτεκτονική Λογισμικού» και εμπεριέχει την ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν στην υλοποίηση που συνοδεύει την παρούσα εργασία, στις Πηγές Δεδομένων που χρησιμοποιήθηκαν με λεπτομέρειες για τις εγγραφές που λήφθηκαν και τέλος η αρχιτεκτονική του συνολικού λογισμικού. Ακολουθούν τα κεφάλαια «Υλοποίηση και Πειράματα» όπου περιγράφονται οι δοκιμές που πραγματοποιήθηκαν πάνω στην αναπτυσσόμενη υλοποίηση και το κεφάλαιο «Συμπεράσματα» όπου παρουσιάζεται η συνολική εικόνα της εργασίας ως προς τους αρχικούς στόχους, την υλοποίηση και παρέχει μελλοντικές κατευθύνσεις για έρευνα. Τέλος ακολουθούν τα κεφάλαια «Βιβλιογραφικές Πηγές» και «Οδηγίες Χρήσεις GraphKer, Πίνακες και εξτρά υλικό».

2 Επισκόπηση του Χώρου

Στο πλαίσιο του κεφαλαίου 2 πραγματοποιείται μια αναλυτική επισκόπηση των:

- Γνωστών πηγών πληροφοριών ευπαθειών και των καταλόγων αδυναμιών που διατίθενται από τον NIST και τον MITRE.
- Γνωστών μεθοδολογιών Βάσεων Δεδομένων Γράφων
- Εφαρμογών Βάσεων Δεδομένων Γράφων σε πληροφορίες σχετικές με αδυναμίες και ευπάθειες.

2.1 Βάσεις Δεδομένων Αδυναμιών και Κατάλογοι Απειλών

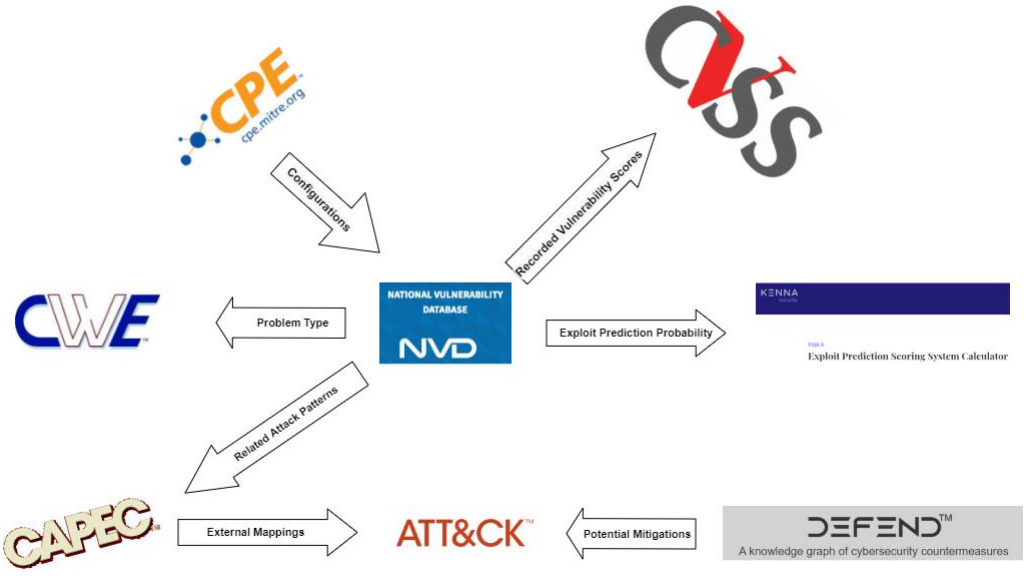
Για να δημιουργήσουμε μια ισχυρή και αξιόπιστη οντολογία ασφάλειας, χρησιμοποιούμε μια σειρά ταξινομιών, καταλόγων και μοντέλων που κατασκευάστηκαν και συντηρούνται από τη NIST, τον MITRE και τη FIRST. Αυτή η προσέγγιση μας δίνει δυνατότητες όπως το να τροφοδοτούμε μοντέλα υπολογισμού ρίσκου που χρησιμοποιούν το μοτίβο συσκευής-απειλής-ευπάθειας αυτόματα από νόμιμες πηγές δεδομένων όπως είναι οι κατάλογοι CPE, CAPEC, CWE και η βάση δεδομένων που εμπεριέχει τα CVE. Προχωρώντας ένα βήμα παραπέρα, χρησιμοποιόντας το πλαίσιο ATT&CK που προσομοιώνει τη συμπεριφορά των κακόβουλων χρηστών σε διάφορα στάδια επιθέσεων στον κυβερνοχώρο και το D3FEND που προσομοιώνει τα ανάλογα αμυντικά μέτρα. Τα παραπάνω πρότυπα μαζί με τις σχέσεις τους αναλύονται περαιτέρω παρακάτω:

- Το MITRE Adversarial Tactics, Techniques and Common Knowledge (ATT&CK) [5] είναι ένα πλαίσιο που προσομοιώνει τη συμπεριφορά κακόβουλων χρηστών στον κυβερνοχώρο στα διάφορα στάδια του κύκλου ζωής μιας επίθεσης. Το μοντέλο συμπεριφοράς ATT&CK αποτελείται ουσιαστικά από τρία βασικά στοιχεία:
    - Τις τακτικές, οι οποίες αντιπροσωπεύουν βραχυπρόθεσμους τακτικούς στόχους κατά τη διάρκεια μιας επίθεσης.
    - Τεχνικές, οι οποίες αντιπροσωπεύουν τα μέσα μέσω των οποίων οι κακόβουλοι χρήστες επιτυγχάνουν τους τακτικούς στόχους.
    - Τεκμηριωμένα δεδομένα συμπεριφορών κακόβουλων χρηστών κατά την εκτέλεση μιας τεχνικής μαζί με σχετικές πληροφορίες.
- Το Πλαίσιο D3FEND αντιμετωπίζει την ανάγκη για ένα μοντέλο που μπορεί να προσδιορίζει επακριβώς τα στοιχεία και τις δυνατότητες αντιμέτρων για την ασφάλεια στον κυβερνοχώρο. Επιπλέον, είναι απαραίτητο οι επαγγελματίες να γνωρίζουν όχι μόνο ποιες απειλές ισχυρίζεται ότι αντιμετωπίζει μια ικανότητα, αλλά συγκεκριμένα πώς αντιμετωπίζονται αυτές οι απειλές σε τεχνικό επίπεδο και υπό ποιες συνθήκες θα λειτουργούσε η λύση. Το D3FEND, επίσης αποτελεί έναν γράφο γνώσης. Ο γράφος περιέχει σημασιολογικά αυστηρούς τύπους και σχέσεις που ορίζουν τόσο τις βασικές έννοιες στον τομέα αντιμέτρων της κυβερνοασφάλειας όσο και τις σχέσεις που είναι απαραίτητες για τη σύνδεση αυτών των εννοιών μεταξύ τους. Οι εγγραφές D3FEND κατηγοριοποιούνται σε τακτικές και τεχνικές χρησιμοποιώντας παρόμοια άποψη με τις εγγραφές ATT&CK, οι τεχνικές D3FEND συνδέονται και στοχεύουν στον μετριασμό των τεχνικών ATT&CK
- Το Common Platform Enumeration (CPE) είναι ένα δομημένο σχήμα ονοματοδοσίας για υλικό υπολογιστών, λειτουργικά συστήματα, και λογισμικά . Η χρήση καταγεγραμμένων προϊόντων CPE διευκολύνει τη χαρτογράφηση των διαφόρων στοιχείων που βρίσκονται στα συστήματα, ενώ διάφοροι σαρωτές που χρησιμοποιούνται για την απαρίθμηση χρησιμοποιούν τον κατάλογο CPE στα αποτελέσματα τους. Τελευταίο αλλά εξίσου σημαντικό για κάθε προϊόν που καταγράφεται στον κατάλογο CPE, οι συνδεδεμένες εγγραφές ενδέχεται να βρίσκονται στους καταλόγους CVE και CWE.

- Ο κατάλογος Common Weakness Enumeration (CWE) αποτελεί έναν κατάλογο αδυναμιών λογισμικού και υλικού. Οι αδυναμίες μπορεί να είναι ελαττώματα ή σφάλματα στη σχεδίαση λογισμικού και υλικού, σε πλαίσιο αρχιτεκτονικής, κώδικα ή ακόμα και υλοποίησης που έχουν τη δυνατότητα να καταστήσουν τα συστήματα, τα δίκτυα και το υλικό ευάλωτα σε επιθέσεις. Ο κατάλογος CWE υποστηρίζεται τόσο από προγραμματιστές όσο και από επαγγελματίες στο χώρο της ασφάλειας. Εκπαιδεύοντας τους αρχιτέκτονες, τους σχεδιαστές, τους προγραμματιστές και τους αγοραστές λογισμικού και υλικού εξοπλισμού σχετικά με τον τρόπο εξάλειψης των πιο συνηθισμένων αδυναμιών πριν από την παράδοση των προϊόντων, τα τρωτά σημεία μπορούν να εξαλειφθούν στην πηγή τους.
- Ο κατάλογος Common Vulnerabilities and Exposures (CVE) περιλαμβάνει δημοσίως αναγνωρισμένες ευπάθειες. Ως ευπάθεια ορίζεται μια «αδυναμία στην υπολογιστική λογική (π.χ. κώδικας) που εντοπίζεται σε στοιχεία λογισμικού και υλικού που, σε περίπτωση εκμετάλλευσης της, έχει αρνητικό αντίκτυπο στην εμπιστευτικότητα, την ακεραιότητα ή τη διαθεσιμότητα». Κάθε καταχώριση CVE, περιέχει έναν αριθμό αναγνώρισης, μια περιγραφή και τουλάχιστον μία αναφορά. Πρόσθετες πληροφορίες μπορούν να περιλαμβάνουν πληροφορίες επιδιόρθωσης, βαθμολογίες σοβαρότητας και αξιολογήσεις επιπτώσεων σύμφωνα με το Κοινό Σύστημα Βαθμολόγησης Ευπαθειών (CVSS).
- Το Common Vulnerability Scoring System (CVSS) [10]είναι ένα σύστημα μέσω του οποίου ορίζονται τα χαρακτηριστικά μιας ευπάθειας και παράγεται μια βαθμολογία που αντικατοπτρίζει τη σοβαρότητα και τον αντίκτυπό της. Οι βαθμολογίες μπορούν στη συνέχεια να μεταφραστούν σε μια ποιοτική αναπαράσταση για να βοηθήσουν τους οργανισμούς να αξιολογήσουν σωστά και να ιεραρχήσουν τις διαδικασίες διαχείρισης ευπαθειών.
- Το Exploit Prediction Scoring System (EPSS) [11] είναι ένα πλαίσιο που βασίζεται σε ανοιχτά δεδομένα για την αξιολόγηση ευπαθειών, στο πλαίσιο του υπολογισμού της πιθανότητας εκμετάλλευσης τους, εντός των πρώτων δώδεκα μηνών μετά τη δημόσια αποκάλυψη τους. Αυτό το σύστημα βαθμολόγησης έχει σχεδιαστεί για να είναι απλό και ευέλικτο, ενώ παρέχει ακριβείς εκτιμήσεις σχετικά με την εκμετάλλευση ευπαθειών. Επιπλέον, η υλοποίηση του προορίζεται για επεκτασιμότητα, επομένως μπορεί να ενημερώνεται όσο περισσότερα και πιο ποιοτικά δεδομένα γίνονται διαθέσιμα. Σε αυτό το πλαίσιο επιτρέπει ήδη στους χρήστες είτε να αναζητήσουν την πιθανότητα εκμετάλλευσης για καταγεγραμμένα CVE είτε να δημιουργήσουν μια προσαρμοσμένη ευπάθεια με τη ρύθμιση των αντίστοιχων χαρακτηριστικών χειροκίνητα.
- Ο κατάλογος Common Attack Pattern Enumeration and Classification (CAPEC) αποτελεί μια δημόσια διαθέσιμη πηγή κοινών μοτίβων επιθέσεων που βοηθά έγκυρους χρήστες συστημάτων και ερευνητές ασφαλείας να κατανοήσουν πώς οι κακόβουλοι χρήστες εκμεταλλεύονται τις αδυναμίες που μπορούν να βρεθούν στον κυβερνοχώρο. Τα μοτίβα επίθεσης είναι ουσιαστικά περιγραφές κοινών χαρακτηριστικών και μεθόδων που χρησιμοποιούνται από κακόβουλους χρήστες για την εκμετάλλευση γνωστών αδυναμιών στον κυβερνοχώρο, επίσης σε ορισμένες περιπτώσεις εμπεριέχουν συγκεκριμένα βήματα και προκλήσεις που ενδέχεται να προκύψουν. Προέρχονται από την έννοια των μοτίβων σχεδίασης που εφαρμόζονται σε ένα καταστροφικό και όχι εποικοδομητικό πλαίσιο και παράγονται από εις βάθος ανάλυση συγκεκριμένων παραδειγμάτων εκμετάλλευσης του πραγματικού κόσμου. Κάθε μοτίβο επίθεσης συλλαμβάνει τη γνώση σχετικά με το πώς σχεδιάζονται και εκτελούνται συγκεκριμένα μέρη μιας επίθεσης και παρέχει καθοδήγηση σχετικά με τρόπους μετριασμού της αποτελεσματικότητας της επίθεσης. Τα μοτίβα επίθεσης βοηθούν όσους αναπτύσσουν, διαχειρίζονται ή υποστηρίζουν εφαρμογές στον κυβερνοχώρο να κατανοήσουν καλύτερα τα συγκεκριμένα στοιχεία μιας επίθεσης και πώς να την αποτρέψουν.

Δεδομένου ότι όλα αυτά τα προϊόντα δημοσιεύονται και συντηρούνται από τον MITRE ή από τον NIST, έχουν σχεδιαστεί ώστε να συνδυάζονται και να αλληλοεπεκτείνονται. Για παράδειγμα, διάφορες τεχνικές που απαριθμούνται στο ATT&CK συνδέονται με συγκεκριμένα αναγνωριστικά CAPEC, επιπλέον τα απαριθμημένα μοτίβα επιθέσεων CAPEC καταγράφονται μαζί με τις σχετικές αδυναμίες τους, αυτή η προσέγγιση προσφέρει μια άμεση σύνδεση και μια ισχυρή σχέση με τον κατάλογο CWE. Τέλος, οι εγγραφές CWE αναλογούν σε πολλαπλές

ευπάθειες (CVE) οι οποίες αναλογούν σε συγκεκριμένο υλικό εξοπλισμό, λειτουργικά συστήματα ή εφαργογές. Οι σχέσεις μεταξύ των υπαρχουσών ταξονομιών απεικονίζονται στην εικόνα 1.



Εικόνα 1 Σχέσεις ανάμεσα σε υπάρχοντες ταξονομίες απειλών

2.2 Βάσεις Δεδομένων Γράφων

Η απλή ιδέα πίσω από αυτήν την τεχνολογία, τα αμέτρητα πεδία εφαρμογών, η ευχρηστία, η αλληλεπίδραση με τεχνολογίες όπως η Μηχανική Μάθηση και η Τεχνητή Νοημοσύνη που χρησιμοποιώντας τις ΒΔΓ, μπορούν να απογειώσουν την εξαγωγή δεδομένων για πληθώρα περιπτώσεων, συμβάλουν στην αύξηση της χρήσης των ΒΔΓ ολοένα και περισσότερο τόσο για ερευνητική όσο και για επιχειρηματική δραστηριότητα. Αξίζει να σημειωθεί, πως σύμφωνα με την πλατφόρμα Dimensions που παρουσιάζει αναλυτικά δεδομένα σχετικά με επιστημονικές δημοσιεύσεις, η χρήση της τεχνολογίας γράφων στην τεχνητή νοημοσύνη αυξάνεται κατακόρυφα. Μάλιστα την τελευταία δεκαετία η ποσοστιαία αύξηση των ερευνητικών άρθρων γύρω από αυτήν αυξήθηκε πάνω από 700%.

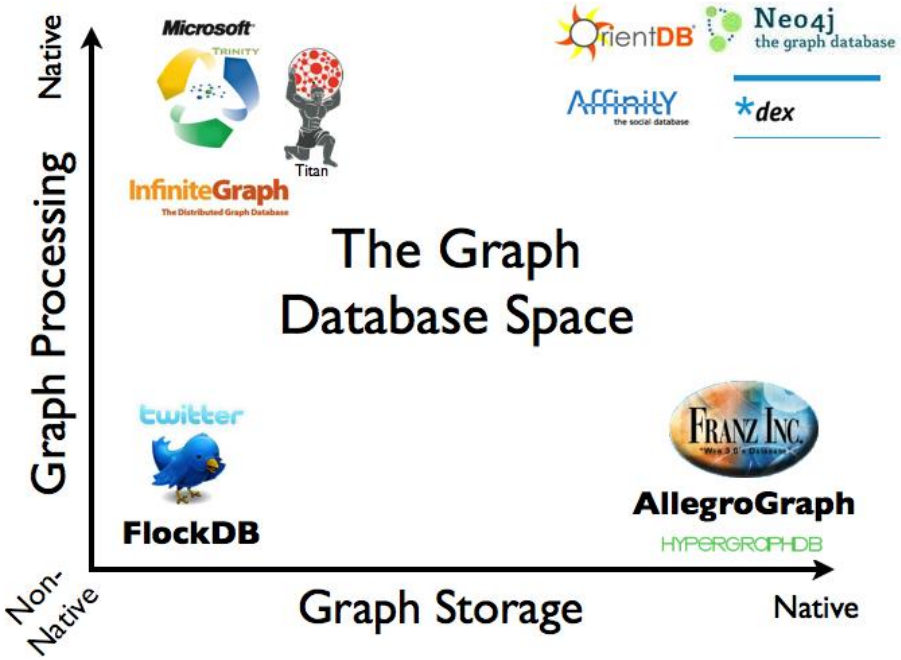
2.2.1 Τεχνολογία

Ένα σύστημα διαχείρισης βάσεων δεδομένων υπό τη μορφή γράφων, δεν είναι τίποτα παραπάνω από μια βάση δεδομένων που υποστηρίζει τις λειτουργίες της Εγγραφής, Ανάγνωσης, Ενημέρωσης και Διαγραφής (**Create, Read, Update, Delete**) παράγοντας σχήμα **υπό μορφή γράφου**. Οι ΒΔΓ είναι έτσι ανεπτυγμένες, ώστε να λειτουργούν αρμονικά με τα συστήματα επεξεργασίας δοσοληψιών (OLTP), λόγω της βελτιστοποιημένης λειτουργίας τους στην εκτέλεση δοσοληψιών και της ακέραιας – δίχως λάθη – συνεχούς λειτουργίας τους.

Εν αντιθέσει με τα άλλα συστήματα διαχείρισης ΒΔ, στις ΒΔΓ οι οντότητες αναπαρίστανται **με κόμβους, οι σχέσεις με βέλη ενώ δεν υπάρχουν πίνακες με γραμμές και στήλες.** Κάθε κόμβος μιας ΒΔΓ αποτελεί **μια ξεχωριστή εγγραφή** όπως θα βλέπαμε σε ένα σχεσιακό σύστημα, ενώ τα βέλη που δείχνουν τις σχέσεις μεταξύ των κόμβων θα μπορούσαμε να τα **συγκρίνουμε με τα ξένα κλειδιά.** Κάθε κόμβος **έχει μια ετικέτα**, βάσει και της οποίας ομαδοποιείται με τους υπόλοιπους και φυσικά **έχει και τα χαρακτηριστικά του** που είναι στην ουσία ζευγάρια κλειδιών με τιμές που μπορεί να έχει **τόσο ο κόμβος όσο και μια σχέση** [12] [13]**.**

Μελετώντας τις ΒΔΓ, αξίζει να εστιάσουμε σε τρία σημεία:

- **Αποθήκευση:** Κάποιες ΒΔΓ χρησιμοποιούν απευθείας την αποθήκευση σε γράφους, η οποία είναι μια αρκετά αποδοτική μέθοδος για την αποθήκευση και διαχείριση των γράφων. Ωστόσο, άλλες, επεξεργάζονται τα δεδομένα αποθηκεύοντας τα σε σχεσιακές ή άλλες βάσεις δεδομένων, με την επιπρόσθετη δουλειά της εξαγωγής των δεδομένων σε γράφους που θα παρουσιαστούν στον χρήστη [14].
- **Μηχανή Επεξεργασίας:** Οι ΒΔΓ διαφέρουν και σε επίπεδο μηχανής επεξεργασίας, καθώς το ότι ο τελικός χρήστης χρησιμοποιεί τις λειτουργίες Create, Read, Update, Delete σε γράφους, δεν σημαίνει ότι η ΒΔΓ επεξεργάζεται με αυτόν τον τρόπο τα δεδομένα. Οι ΒΔΓ που χρησιμοποιούν αποθήκευση απευθείας σε γράφους, δεν χρησιμοποιούν το κλασσικό ευρετήριο όπως για παράδειγμα οι σχεσιακές ΒΔ. Αντιθέτως στη βάση δεδομένων δείχνεται απευθείας η σύνδεση μεταξύ των κόμβων, με την επίδοση προφανώς να είναι σημαντικά διαφορετική.
- **Σχέσεις μεταξύ Οντοτήτων:** Στις ΒΔΓ οι σχέσεις γράφονται στην βάση, ως τρίτη ξεχωριστή οντότητα, εν αντιθέσει με άλλα συστήματα ΒΔ όπου οι συνδέσεις μεταξύ των οντοτήτων χρησιμοποιούν επιμέρους ιδιότητες στις οντότητες, ξένα κλειδιά, ή επεξεργασίες όπως του map-reduce. Κατ' επέκταση το τελικό σχήμα που έχει η βάση είναι πιο απλό στην δομή του, και πιο σύνθετο στις δυνατότητες του, από αυτό που παράγει μια σχεσιακή ΒΔ ή οποιαδήποτε άλλη. Ωστόσο σημειώνουμε πως επειδή ακριβώς κάθε σχέση αναπαρίστανται ως ξεχωριστή οντότητα, η διαχείριση μεγάλων όγκων δεδομένων ενώ θα εξάγει γράφους φιλικούς προς τους χρήστες, είναι μια διαδικασία πιο αργή από αυτήν σε μια σχεσιακή βάση, όπου η δομή είναι προκαθορισμένη, με ευρετήριο, πίνακες κοκ. και με μεγαλύτερη κατανάλωση μνήμης.



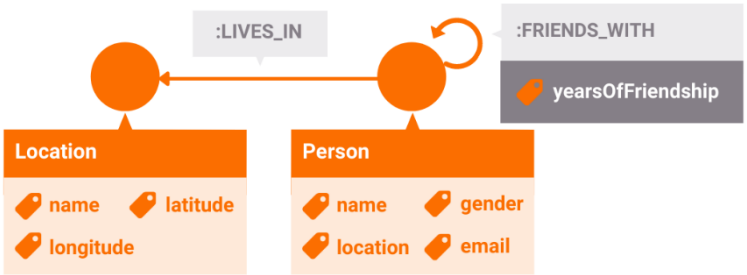
Εικόνα 2: Γνωστές πλατφόρμες ΒΔΓ σύμφωνα με την υποστήριξη της απευθείας επεξεργασίας, αποθήκευσης των δεδομένων με τη μορφή γράφων.

Όπως είναι λογικό το κάθε σύστημα διαχείρισης βάσεων δεδομένων εφαρμόζει καλύτερα στην ανάλογη περίπτωση. Οι ΒΔΓ είναι η βέλτιστη λύση όταν:

1. **Έχουμε πολλαπλές συνδέσεις – σχέσεις μεταξύ των δεδομένων.** Στην περίπτωση που τα δεδομένα συνδέονται μεταξύ τους, έχουν διαφορετικές σχέσεις και θέλουμε να μελετήσουμε τις σχέσεις αυτές, να εξάγουμε συμπεράσματα και αναλυτικά δεδομένα, τότε

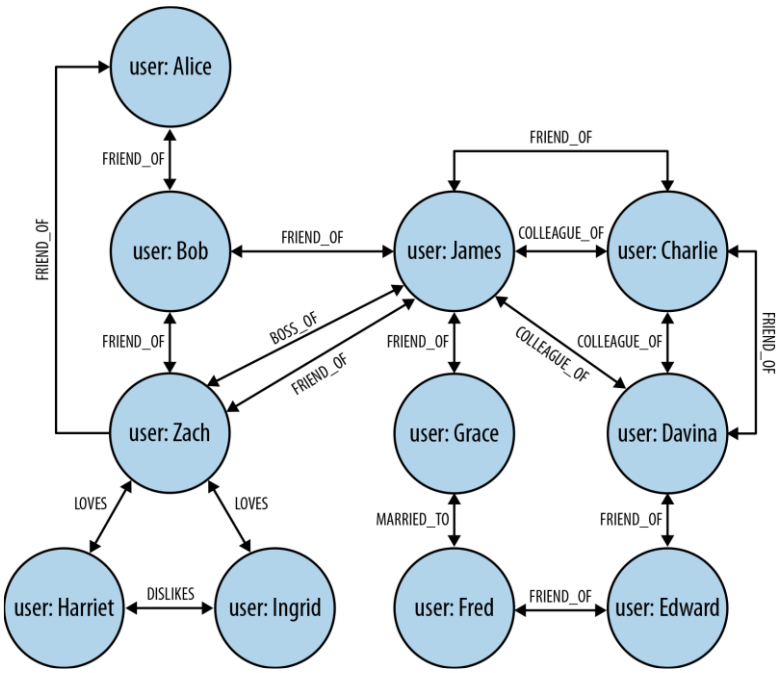
οι ΒΔΓ είναι ο πιο αποδοτικός τρόπος για την αποθήκευση και επεξεργασία των δεδομένων αυτών. Για παράδειγμα, έστω το σχήμα της Εικόνας 3. Έχουμε τους κόμβους Person, Location με τις ιδιότητες τους ο καθένας και τις σχέσεις Lives_In και Friends_With. Στην περίπτωση που θέλουμε να αποθηκεύσουμε ένα αξιοσημείωτου μεγέθους dataset με αυτά τα δεδομένα, χωρίς όμως να αποθηκεύσουμε τις σχέσεις ή στην περίπτωση που η σχέση για παράδειγμα Friends_With δεν εμπίπτει σε πολλές εγγραφές, τότε αποδοτικότερη θα ήταν η χρήση μιας σχεσιακής βάσης δεδομένων ή μιας άλλης NoSQL ΒΔ. Αντιθέτως, αν θέλουμε να εξάγουμε συμπεράσματα σύμφωνα με τις σχέσεις Friends_With μεταξύ των κόμβων Person τότε θα χρησιμοποιήσουμε μια ΒΔΓ.

2. **Θέλουμε να εξάγουμε αναλυτικά δεδομένα.** Η προσπέλαση των αποθηκευμένων στοιχείων σε μια ΒΔΓ ιδίως στην περίπτωση ύπαρξης πολλαπλών σχέσεων είναι σημαντικά ταχύτερη από αυτήν άλλων συστημάτων διαχείρισης ΒΔ. Δεν χρησιμοποιούνται JOIN λειτουργίες. Αντιθέτως στην περίπτωση που δεν θέλουμε να εξάγουμε σύνθετα αναλυτικά δεδομένα, αλλά κυρίως να εγγράφουμε στην ΒΔ, τότε μια ΒΔΓ δεν είναι η αποδοτικότερη λύση.
3. **Τα δεδομένα χρειάζονται συνεχείς αλλαγές.** Όταν τα δεδομένα της ΒΔ χρειάζονται συνεχείς αλλαγές, για παράδειγμα στις σχέσεις μεταξύ των οντοτήτων, τότε η αποδοτική λύση της χρήσης μιας ΒΔΓ, μας διευκολύνει στην ενημέρωση των σχέσεων μεταξύ των κόμβων του γράφου. Αυτό συμβαίνει καθώς στις ΒΔΓ η προσαρμογή του σχήματος είναι απλούστερη, γιατί η εστίαση γίνεται στα δεδομένα και όχι στο σχήμα αυτό καθ' αυτό όπως για παράδειγμα στις σχεσιακές ΒΔ. Αξίζει να σημειωθεί πως αν οι αλλαγές αφορούν συγκεκριμένα χαρακτηριστικά των κόμβων τότε δεν γίνεται γρηγορότερα η ενημέρωση στις ΒΔΓ, αλλά στις σχεσιακές ΒΔ. Δηλαδή η απόδοση των ΒΔΓ θα φανεί στις αλλαγές επί του ίδιου του σχήματος, για παράδειγμα στο που συνδέονται οι κόμβοι. Έστω το παράδειγμα της Εικόνας 3. Αν χρειαστεί να προσθέσουμε μια σχέση (για παράδειγμα Relative_With από Person σε Person) ή να αλλάξουμε υπάρχουσες (σε μεγάλη κλίμακα) τότε εφόσον χρησιμοποιούμε μια ΒΔΓ η όλη διαδικασία θα γίνει γρήγορα. Αντ' αυτού αν θέλουμε να αλλάξουμε το πεδίο Location στον Person τότε η διαδικασία θα γίνει γρηγορότερα σε μια σχεσιακή ΒΔ.



Εικόνα 3: Σχήμα σε ΒΔΓ για χρήση σε παραδείγματα.

Το παρακάτω σχήμα (Εικόνα 4) περιέχει πολλαπλές σχέσεις μεταξύ των οντοτήτων. Παρατηρώντας τις σχέσεις καταλαβαίνουμε πως είναι ευμετάβλητες (βλ. Friend_Of, Colleague_Of κλπ.), και ίσως χρειαστεί να ενημερώνονται συχνά. Γι' αυτό ακριβώς η χρήση μιας ΒΔΓ που θα αποθηκεύσει και θα επεξεργαστεί τα δεδομένα απευθείας υπό τη μορφή γράφων είναι η πιο αποδοτική λύση για τα παραπάνω δεδομένα.



Εικόνα 4: *Ευμετάβλητες σχέσεις μεταξύ των οντοτήτων.*

Σε σχήμα αντίστοιχο της Εικόνας 4, στο βιβλίο «Neo4j in Action» οι επιστήμονες Partner & Vukotic [15] πραγματοποίησαν ένα πείραμα για την αποδοτικότητα της πλατφόρμας Neo4j ως native ΒΔΓ σε σχέση με μια σχεσιακή ΒΔ. Το αντικείμενο του πειράματος ήταν η μέτρηση του χρόνου εκτέλεσης για την εύρεση διασυνδεδεμένων στοιχείων. Πιο συγκεκριμένα, το ερώτημα στη βάση έψαχνε στο κοινωνικό δίκτυο των φίλων, τους φίλους των φίλων. Η εκτέλεση αυτή αφορούσε τα επίπεδα 2 έως 5 δηλαδή (User1)-[:Friend_With]->(User2)-[:Friend_With]->(User3) κοκ. Το Dataset περιείχε 1.000.000 εγγραφές Users. Στον παρακάτω πίνακα θα δείτε τα αποτελέσματα, **τα οποία καταδεικνύουν τη σημαντική διαφορά στον χρόνο εκτέλεσης της σχεσιακής ΒΔ με την native ΒΔΓ.**

Βάθος	Χρόνος Εκτέλεσης Σχεσιακής ΒΔ (s)	Χρόνος Εκτέλεσης Neo4j – Native ΒΔΓ (s)	# Εγγραφών
2	0.016	0.01	~2500
3	30.267	0.168	~110000
4	1543.505	1.359	~600000
5	Δεν Τερμάτισε	2.132	~800000

Παρόλα αυτά, αν δημιουργήσουμε ερωτήματα με διαφορετικό τρόπο, **μπορούμε να αυξήσουμε σημαντικά τον χρόνο εκτέλεσης στις ΒΔΓ.** Τα ερωτήματα **που δεν έχουν σαφές σημείο εκκίνησης** (με τη μορφή συγκεκριμένου κόμβου ή κατηγορίας κόμβων κλπ.) οδηγούν τη ΒΔΓ να προσπελάσει όλους τους γράφους της βάσης! Προφανώς μια ΒΔΓ δεν είναι βελτιστοποιημένη για τέτοιου τύπου αναζητήσεις, όπου προτιμότερο θα ήταν να χρησιμοποιούσαμε ένα NoSQL σύστημα διαχείρισης ΒΔ (βλ. Apache Spark με Map Reduce).

Έπειτα, **στην περίπτωση που χρησιμοποιούμε την βάση για αναζητήσεις, ενημερώσεις συγκεκριμένων κλειδιών – τιμών, δηλαδή δεν αξιοποιούμε σχέσεις μεταξύ οντοτήτων,** αλλά αναζητούμε στοιχεία βάσει μιας σχέσης κλειδιού – τιμής (βλ. Αναζήτηση πολίτη μέσω ΑΜΚΑ) τότε οι ΒΔΓ θα δυσχεράνουν την εκτέλεση του ερωτήματος. Χαρακτηριστικό παράδειγμα είναι η περίπτωση που οι συνήθεις αναζητήσεις μας σε μια ΒΔΓ επιστρέφουν έναν

και μοναδικό κόμβο, χωρίς σχέσεις με άλλους κοκ. Τότε θα έχουμε χρησιμοποιήσει λάθος σύστημα διαχείρισης ΒΔ για την περίπτωση χρήσης μας.

Τέλος, **στην περίπτωση που επιθυμούμε να αποθηκεύσουμε τιμές μεγάλου μεγέθους στις εγγραφές μας,** (βλ. μεγάλα αλφαριθμητικά, κείμενα κοκ.) θα πρέπει είτε να τα αποθηκεύσουμε στην ΒΔΓ, με την βοήθεια ειδικών αναγνωριστικών που θα χρησιμοποιούνται για την εύκολη και σύντομη αναζήτηση των κόμβων, είτε να μην χρησιμοποιήσουμε μια ΒΔΓ. Η ΒΔΓ δεν είναι η βέλτιστη μορφή για να αναζητήσει τους κόμβους και τις οντότητες βάσει αναγνωριστικών που περιέχουν για παράδειγμα κάποιο μεγάλο κείμενο. Για παράδειγμα αν έχουμε το σχήμα της Εικόνας 3 και προσθέσουμε στον Person το χαρακτηριστικό «Resume» τότε προφανώς η ΒΔΓ θα καθυστερήσει σημαντικά αν ψάξουμε τους κόμβους όπου στο χαρακτηριστικό του Resume περιέχεται ένα συγκεκριμένο αλφαριθμητικό. Μια διαφορετική λύση σε αυτό θα ήταν η αλλαγή στο dataset όπου το Resume θα είχε συγκεκριμένα πεδία (βλ. Skills, Contact_Number, Intro_Text), θα συνδεόταν με τον (Person)-[:Has_Resume]->(Resume) και θα μπορούσαμε να κάνουμε μια αναζήτηση για παράδειγμα στο πεδίο Skills του Resume. Τότε θα ήταν βέλτιστη η χρήση της ΒΔΓ.

Κάθε νέο τεχνολογικό επίτευγμα, κάθε νέα πλατφόρμα, έχει συγκεκριμένους τρόπους χρήσεις και εμπίπτει σε διαφορετικές περιπτώσεις χρήσης. Η εξειδικευμένη χρήση σε συνδυασμό με την εκμετάλλευση των καινοτομιών κάθε πλατφόρμας, οδηγεί τους επιστήμονες και τους εργαζόμενους στην αποδοτικότερη αξιοποίησή τους και εν τέλει στην βέλτιστη επίτευξη των επιθυμητών αποτελεσμάτων. Υπό αυτήν την έννοια για λόγους που θα αναφερθούν παρακάτω, θα αναλύσουμε και θα αξιοποιήσουμε το σύστημα διαχείρισης ΒΔΓ Neo4j για την δική μας περίπτωση χρήσης.

2.2.2 Συνήθεις περιπτώσεις χρήσης

Όπως προαναφέραμε, οι δύο στόχοι της χρήσης των ΒΔΓ είναι η εύρεση της συνδεσιμότητας μεταξύ οντοτήτων και η εξαγωγή γνώσης και αναλυτικών δεδομένων. Σε αυτήν την ενότητα θα μελετήσουμε τις κυριότερες περιπτώσεις χρήσης για τις οποίες οι ΒΔΓ είναι η πλέον αποδοτική λύση [16].

Κοινωνικά Δίκτυα

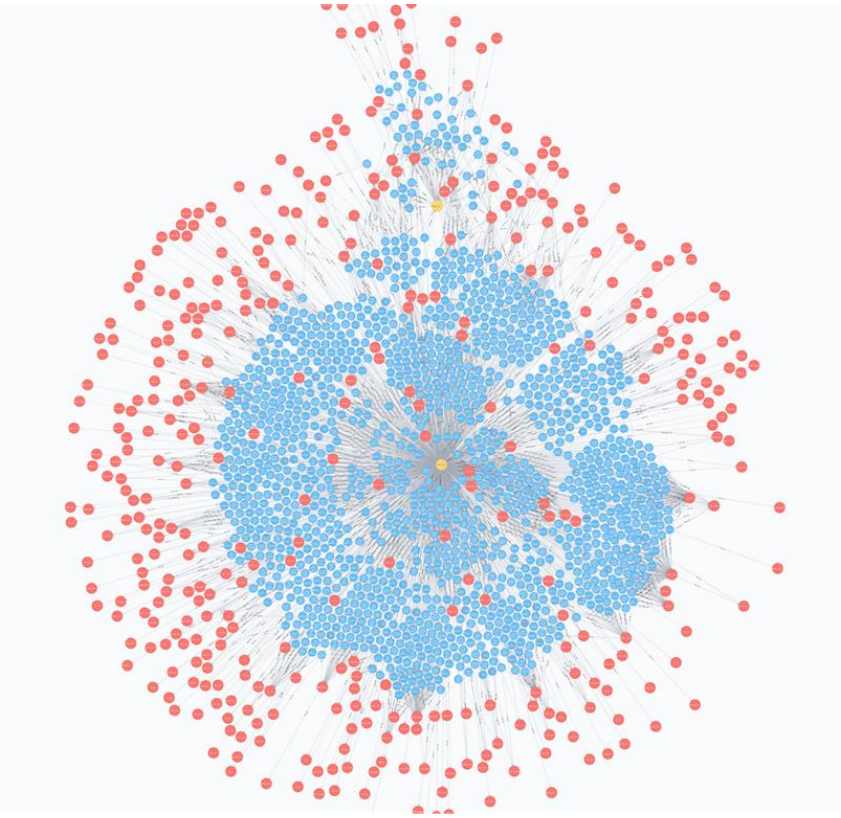
Οι κοινωνικές σχέσεις, η συμπεριφορά των ανθρώπων, αποτελεί ένα πεδίο μελέτης για τους επιστήμονες τόσο των κοινωνικών επιστημών όσο και των θετικών. Στο βιβλίο τους «Συνδεδεμένοι» οι επιστήμονες Νικόλας Χρηστάκης και James Fowler αποδεικνύουν πως χωρίς να γνωρίζεις τίποτα για έναν άνθρωπο μπορείς να προβλέψεις την συμπεριφορά του, τις συνήθειες του, τις επιθυμίες του κοκ. μέσω της μελέτης των ανθρώπων με τους οποίους έχει επαφές, δηλαδή μέσω αυτών που είναι συνδεδεμένοι μαζί του [17]. Στην πραγματικότητα μιλάμε για μια επιστημονική αναπαράσταση της λαϊκής ρήσης «Δείξε μου τον φίλο σου, να σου πω ποιος είσαι». Τα κοινωνικά δίκτυα και η τεχνολογία πίσω τους, είναι το πιο χαρακτηριστικό παράδειγμα καθώς επιτρέπει στους οργανισμούς, να αξιοποιήσουν τα αναλυτικά δεδομένα για να προβλέψουν τη συμπεριφορά ενός χρήστη, και κατ' επέκταση να αξιοποιήσουν την πληροφορία για την προώθηση του προϊόντος ή της υπηρεσίας τους. Φυσικά δεν μιλάμε μόνο για οργανισμούς που αποσκοπούν στο κέρδος, αλλά για πληθώρα υποκειμένων που προσπαθούν να αξιοποιήσουν τα αναλυτικά δεδομένα, για ίδιον όφελος, με πιο χαρακτηριστικό το σκάνδαλο της Cambridge Analytica, για την πρόβλεψη της συμπεριφοράς του κοινού και την άσκηση επιρροής στις εκλογές διάφορων χωρών.

Το Facebook χρησιμοποιώντας τον όρο κοινωνικός γράφος (social graph) εννοεί, πως πίσω από την πλατφόρμα λειτουργούν μοντέλα και βάσεις δεδομένων γράφων, καθώς έτσι θα γίνει η βέλτιστη παρουσίαση των σχέσεων μεταξύ των οντοτήτων (βλ. (Person)-[:Friend_With]→(Person)). Στα κοινωνικά δίκτυα μπορούμε να αναγνωρίσουμε τις άμεσες ή έμμεσες σχέσεις μεταξύ ανθρώπων, ομάδων και κατ' επέκταση των ενδιαφερόντων τους για παράδειγμα τις σελίδες στο Facebook που ακολουθούν. Μέσω της γνώσης αυτής -που σημειωτέων αποτελεί ένα από τα λεγόμενα μεγάλα δεδομένα λόγω του όγκου της-, έχουμε την δυνατότητα φυσικά να επηρεάσουμε και συμπεριφορές. Λόγω της ευελιξίας των ΒΔΓ στην

αναπαράσταση των οντοτήτων, με εύκολο τρόπο αναπαρίστανται συνδέσεις πέραν του Friend_With, όπως :Likes, :Attends κοκ.

Αντίστοιχα και στο Twitter, οι ΒΔΓ μπορούν να αναπαραστίσουν πληθώρα οντοτήτων πέραν των χρηστών όπως Tweet, Hashtag, Country κλπ. και σχέσεις όπως :Tweeted, :Retweeted, :Mentioned :FromCountry :UsedHashtag κλπ. όπου με αυτόν τον τρόπο λαμβάνουμε έναν σημαντικό όγκο αναλυτικών δεδομένων. Με αυτόν τον τρόπο βγαίνουν τα Twitter Graphs που χρησιμοποιούνται ευρέως για να καταλάβουμε το τι συζητούν οι χρήστες στο Twitter, ποιοι χρήστες έχουν τα περισσότερα Retweets, και πως μπορεί να πολωθεί μια συζήτηση για παράδειγμα με διαφορετικά hashtags. Μάλιστα το Neo4j συνδέεται εύκολα με το API του Twitter για να γίνει η διαδικασία αποθήκευσης στη ΒΔΓ και αναπαράστασης των δεδομένων πολύ γρήγορη.

Όπως βλέπουμε στο παρακάτω σχήμα (Εικόνα 5) μέσω του Neo4j και του Twitter Search API, δημοσιογράφοι του NBC απέδειξαν πως δρουν οι ψεύτικοι λογαριασμοί στο Twitter που προέρχονται από την Ρωσία και στόχο είχαν να επηρεάσουν την κοινή γνώμη των Η.Π.Α. στις προεδρικές εκλογές του 2016. Στους κίτρινους κόμβους φαίνονται οι δύο χρήστες, οι οποίοι δημοσίευαν το πρωτότυπο περιεχόμενο ενώ από αυτούς παρήχθησαν χιλιάδες Tweets (μπλε κόμβοι) με συγκεκριμένα hashtags (κόκκινοι κόμβοι).



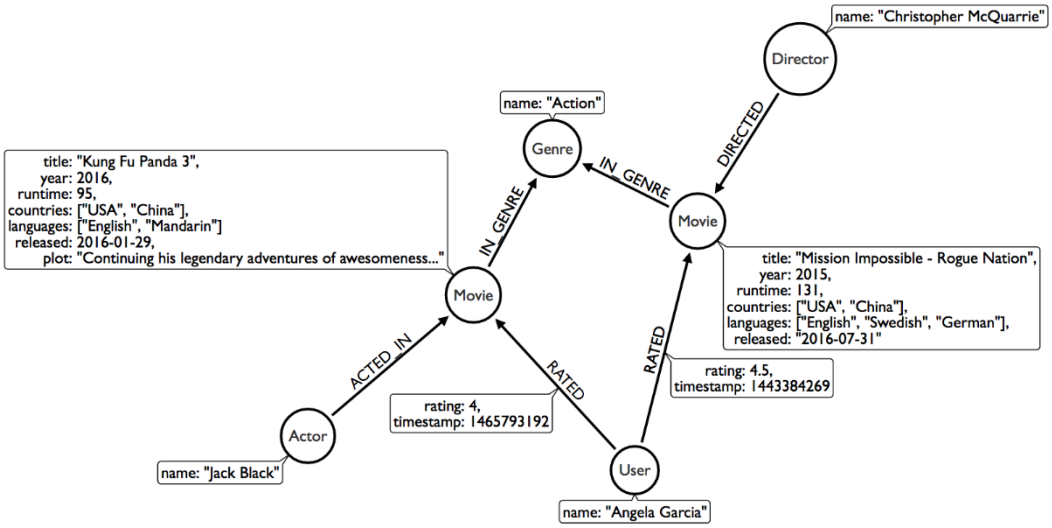
Εικόνα 5: Ψεύτικοι Λογαριασμοί στο Twitter [18].

Συστήματα Προτάσεων

Με ποιον τρόπο το Netflix θα προτείνει στον χρήστη μια ταινία ή σειρά που μπορεί να του αρέσει; Τα αποτελεσματικά συστήματα προτάσεων είναι χαρακτηριστικό παράδειγμα της χρήσης των ΒΔΓ. Η λογική πίσω από αυτά είναι απλή. Αρκεί να αναπτύξουμε σχέσεις μεταξύ ανθρώπων και πραγμάτων -ταινίες, προϊόντα, υπηρεσίες κλπ.-. Κάθε φορά που ο χρήστης προχωράει σε μια ενέργεια για παράδειγμα αναζητεί ή αγοράζει ένα προϊόν ή παρακολουθεί μια ταινία, δημιουργεί μια σχέση στη ΒΔΓ μεταξύ των δύο αυτών πλευρών. Με αυτόν τον τρόπο η ΒΔΓ αξιοποιεί το βάθος των συνδέσεων της δεύτερης πλευράς ώστε να προτείνει στον χρήστη κάτι που μπορεί όντως να τον ενδιαφέρει [19]. Παράλληλα, έχει τη δυνατότητα να ομαδοποιήσει χρήστες και

πράγματα με σκοπό την πρόβλεψη συμπεριφοράς για την βελτιστοποίηση των προτάσεων. Η δυνατότητα αυτή γίνεται και αντίστροφα. Όπου η ΒΔΓ, ξεκινά από το δεύτερο σκέλος δηλαδή το προϊόν, την ταινία κοκ. και καταλήγει στον χρήστη ή την ομάδα χρηστών βάσει του προφίλ τους. Η βελτιστοποίηση των συστημάτων προτάσεων δεν εξαρτάται μόνο από τις συνδέσεις αυτές καθ' αυτές μεταξύ δύο οντοτήτων, αλλά και από την ποιότητα αυτών των σχέσεων. Για παράδειγμα σε ένα κοινωνικό δίκτυο ενώ ένας άνθρωπος εν δυνάμει μπορεί να γίνει φίλος με άλλους δύο, το σύστημα θα φέρει σε προτεραιότητα αυτόν με τους περισσότερους κοινούς φίλους ή με έναν συνδυασμό κοινών φίλων και κοινών ενδιαφερόντων. Αντίστοιχα, σε ένα σύστημα προτάσεων ταινιών αν ο χρήστης δει τέσσερις κωμωδίες και μια περιπέτεια, το σύστημα θα του προτείνει πρώτα άλλη μια κωμωδία και σε δεύτερο επίπεδο μια περιπέτεια.

Στο παρακάτω σχήμα (Εικόνα 6) παρουσιάζεται ο τρόπος μοντελοποίησης ενός συστήματος προτάσεων σε ΒΔΓ. Ο χρήστης έχει βαθμολογήσει την ταινία Mission Impossible με 4,5. Η ταινία αυτή είναι Action και έχει σκηνοθετηθεί από τον Christopher McQuarrie. Αν προσθέταμε άλλες δύο ταινίες του ίδιου σκηνοθέτη με τη μια να είναι Action και την άλλη Drama. Το σύστημα θα πρότεινε στον χρήστη να δει την Action γιατί ήδη είδε άλλη μια της ίδιας κατηγορίας, παρόλο που και οι δύο ανήκουν στον σκηνοθέτη του οποίου την ταινία ήδη παρακολούθησε. Αν κάνουμε το σχήμα πιο πολύπλοκο και προσθέσουμε μια σχέση Viewed. Ο χρήστης, λοιπόν, είδε δύο ταινίες έστω Drama, αλλά βαθμολόγησε (Rated) με καλό βαθμό την Action. Το σύστημα προτάσεων θα επιλέξει να προτείνει άλλη μια Action σε παραπάνω θέση από τις Drama, γιατί παρόλο που είδε δύο ταινίες Drama, βαθμολόγησε με καλό βαθμό την Action.



Εικόνα 6: Σύστημα προτάσεων υλοποιημένο σε ΒΔΓ.

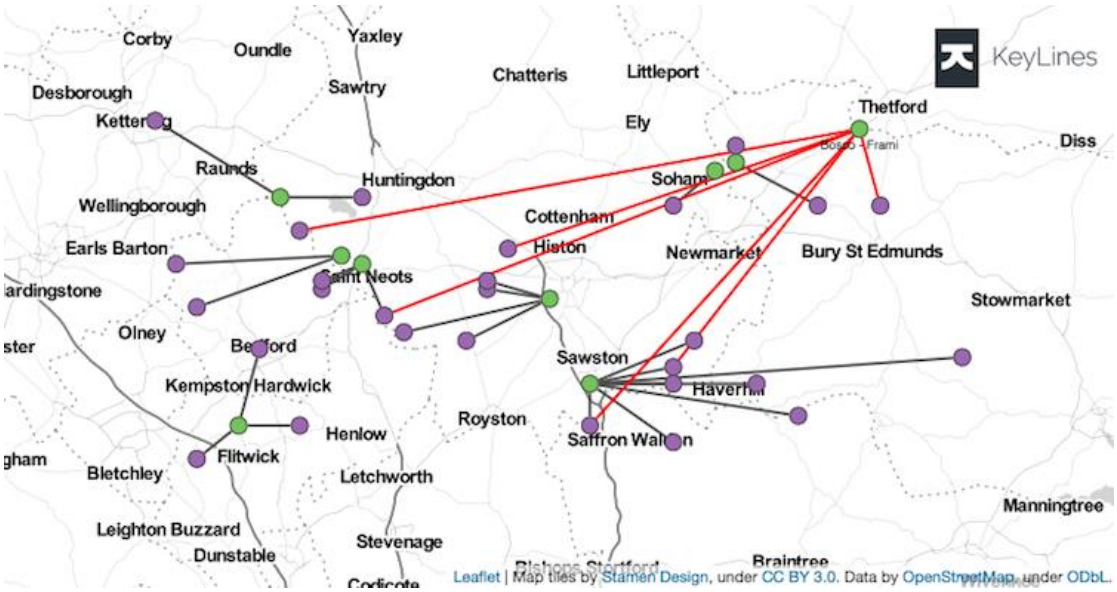
Γεωγραφικά Δεδομένα

Τα γεωγραφικά δεδομένα είναι η κατεξοχήν περίπτωση χρήσης των γράφων. Ο Euler έλυσε το πρόβλημα των επτά γεφυρών του Κένισμπεργκ μέσω του μαθηματικού θεωρήματος που στη συνέχεια αποτέλεσε βάση της θεωρίας γράφων. Οι εφαρμογές των γεωγραφικών δεδομένων στις ΒΔΓ ποικίλουν, από τον υπολογισμό διαδρομών μεταξύ περιοχών σε ένα δίκτυο (οδικό, σιδηροδρομικό, θαλάσσιο κλπ.) έως τις χωρικές διεργασίες που μπορεί να υπάρχουν σαν δυνατότητες σε εφαρμογές, όπως η εύρεση σημείων ενδιαφέροντος σε μια συγκεκριμένη περιοχή, η εύρεση του κέντρου μιας περιοχής κλπ [20].

Η αποθήκευση και επεξεργασία των γεωγραφικών δεδομένων απαιτεί συγκεκριμένη χρήση των ΒΔΓ. Συγκεκριμένα μπορεί να απαιτεί κατευθυνόμενες σχέσεις, με βάρη ή χωρίς, συγκεκριμένη ευρετηριοποίηση (βλ. R-Trees) ώστε αναπαρασταθούν καλύτερα οι ιδιότητες μέσω των δεντρικών δομών δεδομένων. Από τη στιγμή, λοιπόν, που τα δεδομένα οργανώνονται σε δέντρα, είναι τα πλέον κατάλληλα για την αποθήκευση και επεξεργασία σε μια ΒΔΓ. Η ευελιξία

στο σχήμα που δίνουν οι ΒΔΓ, επιτρέπει φυσικά την σύνθετη επεξεργασία των γεωγραφικών δεδομένων.

Στο παρακάτω σχήμα (Εικόνα 7) βλέπουμε παραπάνω μια αναπαράσταση γεωγραφικών δεδομένων σε γράφο στο Neo4j. Στους μωβ κόμβους φαίνονται οι άνθρωποι ενώ στους πράσινους τα σημεία επισκευής αυτοκινήτων. Στις κόκκινες ακμές φαίνονται οι λάθος επιλογές που έχουν κάνει οι άνθρωποι λόγω απόστασης – κόστους ενώ στις μαύρες ακμές οι σωστές επιλογές.



Εικόνα 7: Αναπαράσταση γεωγραφικών δεδομένων σε γράφο στο Neo4j.

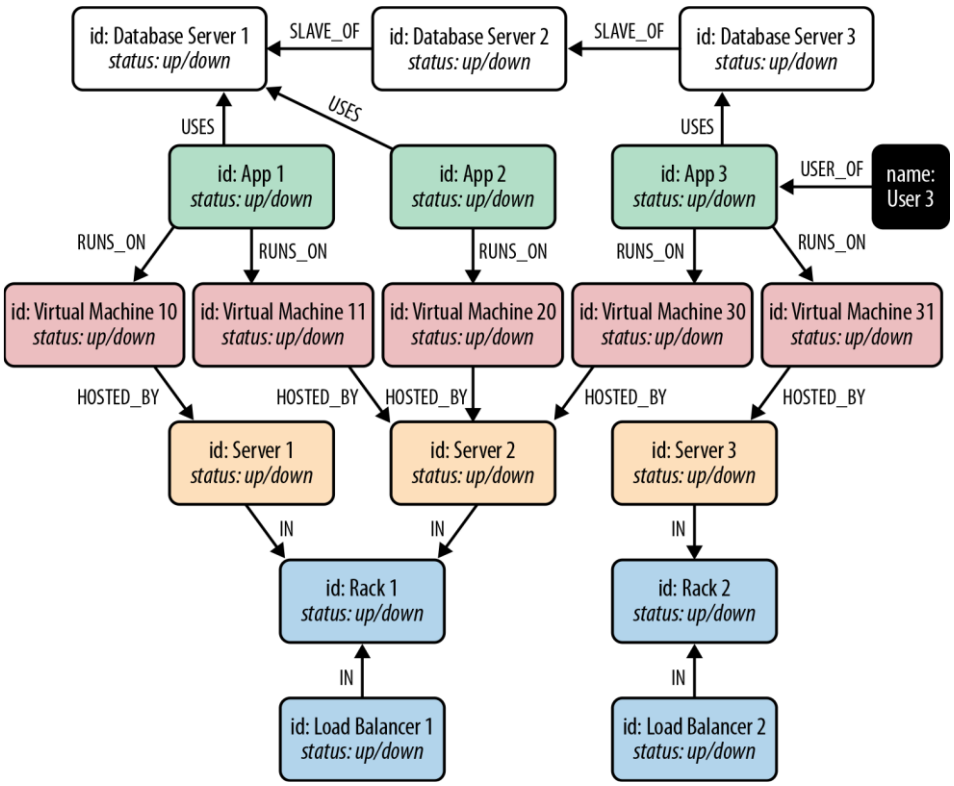
Δίκτυα και Κέντρα Δεδομένων

Στην Εικόνα 8 βλέπουμε την αναπαράσταση των ψηφιακών και φυσικών αγαθών ενός κέντρου δεδομένων σε μια ΒΔΓ. Γενικότερα όλες οι μορφές δικτύωσης στην πραγματικότητα είναι δομές γράφου. Επομένως η χρήση των ΒΔΓ είναι ο πλέον αποδοτικός τρόπος για την αποθήκευση, μελέτη και επεξεργασία τους. Στην περίπτωση των δικτύων επικοινωνιών και των αγαθών των κέντρων δεδομένων μπορούμε να παρατηρήσουμε δύο ιδιαίτερα κοινά σημεία. Η αναπαράσταση ενός δικτύου σε γράφο, συμβάλει στην καλύτερη κατηγοριοποίηση των αγαθών, την οπτικοποίηση της εγκατάστασης τους και των συνδέσεων τους [21].

- Από ποια στοιχεία του δικτύου, εφαρμογές, υπηρεσίες, εικονικές μηχανές, φυσικά αγαθά εξαρτώνται οι πελάτες (Top-Down Analysis).
- Ποιες εφαρμογές, υπηρεσίες, χρήστες του δικτύου, θα επηρεαστούν σε πιθανή δυσλειτουργία ενός στοιχείου του δικτύου. (Bottom-Up Analysis).

Στην πραγματικότητα, μιλάμε για μια ανάλυση επιπτώσεων, η οποία θα γίνει πολύ πιο εύκολα με την χρήση των ΒΔΓ. Σε μεγαλύτερη κλίμακα, χρησιμοποιούμε τις ΒΔΓ ως συμπλήρωμα στην ανάλυση και διαχείριση επικινδυνότητας των πληροφοριακών συστημάτων, γιατί πολύ απλά συμβάλουν στην βέλτιστη εξαγωγή γνώσης, μέσω της οπτικοποίησης των δεδομένων στους γράφους και μέσω των συνδέσεων μεταξύ των αγαθών του πληροφοριακού συστήματος που μας δείχνουν. Φυσικά η χρήση μπορεί να ποικίλει, για παράδειγμα σε επιχειρησιακό επίπεδο να παρατηρούμε την ορθή λειτουργία του δικτύου μιας επιχείρησης, μιας μονάδας τηλεπικοινωνιών κοκ. ή σε λοιπά εργαλεία ανάλυσης της λειτουργίας του δικτύου και του κέντρου δεδομένων.

Όπως βλέπουμε στο παρακάτω σχήμα (Εικόνα 8) τόσο τα φυσικά όσο και τα ψηφιακά αγαθά ενός κέντρου δεδομένων μπορούν να αναπαρασταθούν υπό μορφή γράφων σε ΒΔΓ, για την βέλτιστη κατανόηση και μελέτη των δεδομένων.



Εικόνα 8: Φυσικά και ψηφιακά αγαθά ενός κέντρου δεδομένων.

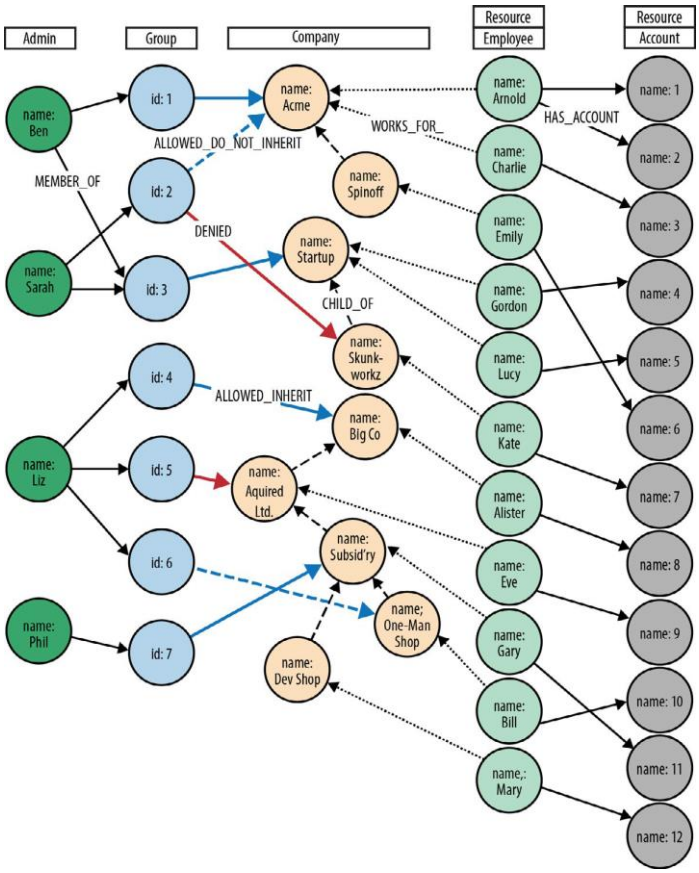
Αυθεντικοποίηση και Συστήματα Ελέγχου Πρόσβασης

Οι λειτουργίες αυτές αποθηκεύουν πληροφορίες για τις διάφορες ομάδες μιας υποδομής (χρήστες, διαχειριστές κλπ.) και για τους πόρους (αρχεία, φυσικά και ψηφιακά αγαθά κλπ.), μαζί με τους κανόνες λειτουργίας σε σχέση με την πρόσβαση των ομάδων στους πόρους [22]. Οι κανόνες αυτοί καταδεικνύουν την πρόσβαση και τα δικαιώματα του χρήστη πάνω στον πόρο όπως για παράδειγμα σε ένα αρχείο η ανάγνωση και η επεξεργασία ή σε ένα χωρικό σημείο του οργανισμού την πρόσβαση σε αυτό. Πριν την χρήση των ΒΔΓ οι δομές δεδομένων πάνω στις οποίες υλοποιούνταν τα συστήματα ελέγχου πρόσβασης, ήταν συνήθως ιεραρχικές και υλοποιημένες σε directory υπηρεσίες ή υποεφαρμογές εντός του λογισμικού Data Management του οργανισμού. Ωστόσο, από τη στιγμή που το οργανωτικό μοντέλο των οργανισμών σε πολλές περιπτώσεις έπαψε να είναι ιεραρχικό, άρα περισσότεροι χρήστες να έχουν μεγαλύτερη πρόσβαση σε δεδομένα (βλ. μια περίπτωση που ο υφιστάμενος λόγω εξειδίκευσης έχει πρόσβαση σε λογισμικό που δεν έχει ο προϊστάμενος), τότε η ιεραρχική δομή δεδομένων και τα παλαιού τύπου μοντέλα δεν ήταν ο βέλτιστος τρόπος. Πιο συγκεκριμένα, οι σχεσιακές ΒΔ που κάλυπταν συνήθως τέτοιες ανάγκες, λόγω των πολλαπλών joins δεν ήταν ο αποδοτικότερος τρόπος λειτουργίας και επεξεργασίας των δεδομένων αυτών, κάτι που έβαζε εμπόδια στην ίδια λειτουργία του οργανισμού (με πιο χαρακτηριστικό παράδειγμα τον χρόνο εκτέλεσης σε πιο περίπλοκα ερωτήματα). Αντιθέτως, οι ΒΔΓ λόγω των χαρακτηριστικών τους, είναι ο πλέον αποδοτικός τρόπος που προσφέρει απόλυτη ευελιξία τόσο για ιεραρχικά όσο και μη ιεραρχικά μοντέλα οργάνωσης κατά συνέπεια και συστημάτων ελέγχου πρόσβασης και αυθεντικοποίησης των χρηστών.

Όπως και στην περίπτωση χρήσης των δικτύων και των κέντρων δεδομένων, έτσι και εδώ έχουμε δύο λύσεις που μπορούμε να χρησιμοποιήσουμε για να αναλύσουμε τα δεδομένα:

- Ποιους πόρους (προϊόντα, φυσικά και ψηφιακά αγαθά, υπηρεσίες, εφαρμογές, χώρους κλπ.) μπορεί να διαχειριστεί ένας admin και αντίστοιχα ένας χρήστης να έχει πρόσβαση. (Top-Down Analysis).
- Δεδομένου ενός πόρου, ποιοι admins μπορούν να κάνουν αλλαγές και ποιοι χρήστες μπορούν να έχουν πρόσβαση σε αυτόν (Bottom-Up Analysis).

Στο παρακάτω σχήμα (Εικόνα 9) βλέπουμε την αναπαράσταση σε ΒΔΓ ενός συστήματος ελέγχου πρόσβασης. Παρατηρούμε την ευκολία για την χρήση τόσο Top-Down Analysis όσο και Bottom-Up Analysis στην διαχείριση αλλά και την πρόσβαση στους πόρους.



Εικόνα 9: Αναπαράσταση σε ΒΔΓ ενός συστήματος ελέγχου πρόσβασης.

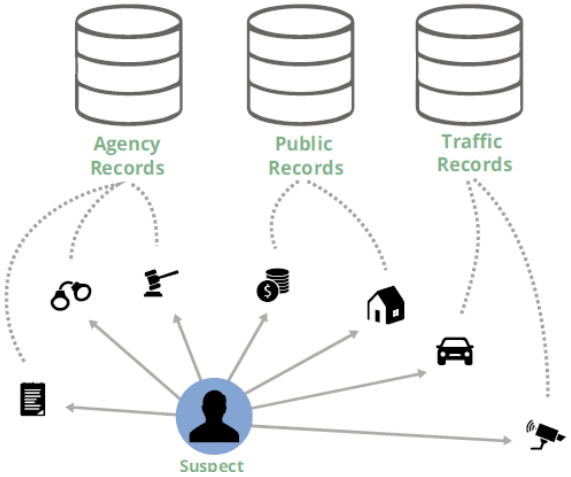
Δημόσιο και Διακυβέρνηση

Τα Συνδεδεμένα Δεδομένα σε Έρευνες Εγκλημάτων

Οι έρευνες εγκλημάτων από τις αρμόδιες αρχές, είναι χαρακτηριστικό παράδειγμα της ανάλυσης των συνδέσεων μεταξύ των δεδομένων, καθώς αποτελούν την βασική μεθοδολογία εύρεσης και ανάλυσης των υπόπτων στις υποθέσεις. Ένας ύποπτος συχνά παρουσιάζεται σε διαφορετικές βάσεις δεδομένων, δημόσιες και μη (για παράδειγμα ένας ύποπτος έχει ΑΜΚΑ, μπορεί να είναι στην ΒΔ μιας τράπεζας, και ενός γυμναστηρίου που είναι εγγεγραμμένος). Η σύνδεση των δεδομένων αυτών, είναι σημαντικός παράγοντας για την επίλυση της υπόθεσης. Αν αναλογιστούμε τη σημασία του χρόνου στην επίλυση μιας υπόθεσης εγκλήματος ή στην προστασία για ζητήματα εθνικής ασφάλειας, τότε καταλαβαίνουμε πως οι ΒΔΓ δημιουργούν το ευνοϊκό πεδίο για τις αρμόδιες υπηρεσίες [23].

(Εικόνα 10) Για την ενοποίηση των αποτελεσμάτων των επιμέρους ΒΔ προφανώς μεσολαβούν δικαστικά εντάλματα, αιτήματα στους οργανισμούς κλπ. Η χρήση των ΒΔΓ από την στιγμή που οι επιμέρους ΒΔ δώσουν τα κατάλληλα στοιχεία, θα μειώσουν σημαντικά τον χρόνο

επίλυσης της υπόθεσης, δεδομένου ότι άμεσα και με τον τρόπο που φαίνεται στην εικόνα, θα παρουσιάσουν τα αποτελέσματα πια ενοποιημένα, στον χρήστη.



Εικόνα 10: Αποδοτική χρήση των ΒΔΓ σε δικαστικές υποθέσεις.

Εξυπηρέτηση Πολιτών

Με την χρήση των ΒΔΓ και των ενοποιημένων αποτελεσμάτων στις ΒΔ, η εξυπηρέτηση των πολιτών από τις δημόσιες υπηρεσίες είναι και πιο εύκολη και πιο γρήγορη. Έστω ότι σε ένα δημόσιο σύστημα υπάρχουν πολλές και διαφορετικές ΒΔ, άλλες για την ασφάλιση, άλλες για την εφορία κοκ. Το πρώτο στάδιο είναι η ενοποίηση των ΒΔ, σε μια, η οποία θα έχει όλες τις πληροφορίες για κάθε πολίτη και την πρόσβαση που έχει στις διάφορες δημόσιες υπηρεσίες. Το στάδιο αυτό βελτιώνεται, αν χρησιμοποιήσουμε μια ΒΔΓ, όπου δεδομένης της ευελιξίας στο σχήμα που μπορεί να φτιαχτεί, μπορούμε να αναπαραστήσουμε όλες τις πιθανές σχέσεις του πολίτη με την οποιαδήποτε δημόσια υπηρεσία. Με αυτόν τον τρόπο 1) Ο πολίτης θα μπορεί να εξυπηρετηθεί πιο εύκολα και γρήγορα 2) Το κράτος θα έχει όλη την εποπτεία σε ένα σύστημα, κοινό για κάθε υπηρεσία και έτσι θα εξοικονομήσει πόρους και φυσικά θα λαμβάνει τα βέλτιστα αναλυτικά δεδομένα.

Εφοδιαστική Αλυσίδα Δημόσιων Υπηρεσιών

Πάλι λόγω της ευελιξίας στην δημιουργία του σχήματος της ΒΔΓ και της οπτικοποίησης των δεδομένων έχουμε άλλη μια περίπτωση χρήσης, πιο συγκεκριμένα την εφοδιαστική αλυσίδα, τόσο δημόσια όσο φυσικά και των οργανισμών. Εν προκειμένω μια δημόσια υπηρεσία όπως ένα νοσοκομείο ή ένα κέντρο υγείας μπορεί να μελετά και να επεξεργάζεται άμεσα τα αποθέματα προμηθειών του, να ενημερώνει τις αρμόδιες υπηρεσίες για παραγγελίες κοκ. με την κυρίαρχη διαφορά πως λόγω ευελιξίας σχήματος μπορεί ενόσω είναι στην ίδια ΒΔΓ με άλλες δημόσιες υπηρεσίες να προσθαφαιρεί πόρους που χρειάζεται που μπορεί φυσικά να μην εμπίπτουν στις άλλες υπηρεσίες. Για παράδειγμα ένα δημοτικό κέντρο υγείας στην Αθήνα έχει άλλες ανάγκες από ένα δημοτικό κέντρο υγείας σε μια επαρχιακή περιοχή, κάτι που φυσικά εξαρτάται από την κοινωνική ζωή, την εργασία της πλειοψηφίας των πολιτών κοκ. Τα δύο κέντρα υγείας βρίσκονται σε μια ενοποιημένη ΒΔΓ και το καθένα προσθαφαιρεί με τη μορφή οντοτήτων και σχέσεων τους πόρους που έχει ανάγκη.

Αντίστοιχο παράδειγμα μπορεί να υπάρξει και σε στρατιωτικές μονάδες. Μάλιστα οι ΗΠΑ, χρησιμοποίησαν το Neo4j για την υποστήριξη της εφοδιαστικής αλυσίδας του στρατού τους. Προσπάθησαν να αντιμετωπίσουν τα πολλά προβλήματα σε αλλαγές πόρων που χρειάζονται, δυσκολίες στη διαχείριση των δεδομένων, καθυστερήσεις στην επικοινωνία, μεγάλα κόστη κλπ. με μια σύγχρονη προσέγγιση του συστήματος διαχείρισης εφοδιαστικής αλυσίδας. Δεδομένου ότι οι ΗΠΑ εμπλέκονται στρατιωτικά με μια σειρά χωρών παγκοσμίως, η διοίκηση του στρατού χρησιμοποίησε το Neo4j ως τον τρόπο για την γρήγορη επεξεργασία και ανάλυση συνδεδεμένων δεδομένων σε: 1) Καιρικές συνθήκες, κλιματικές αλλαγές ανά περιοχή, με παράλληλες στατιστικές μελέτες περιθωρίου λάθους. 2) Πολυδιάστατη ανάλυση κόστους – οφέλους σε πόρους του

στρατού. 3) Ενημέρωση για τους διαθέσιμους πόρους, επισκευή και συντήρηση των αγαθών. 4) Ανάλυση σεναρίων για την προσαρμογή της εφοδιαστικής αλυσίδας σε περίπτωση κρίσης ή εμπλοκής σε πόλεμο.

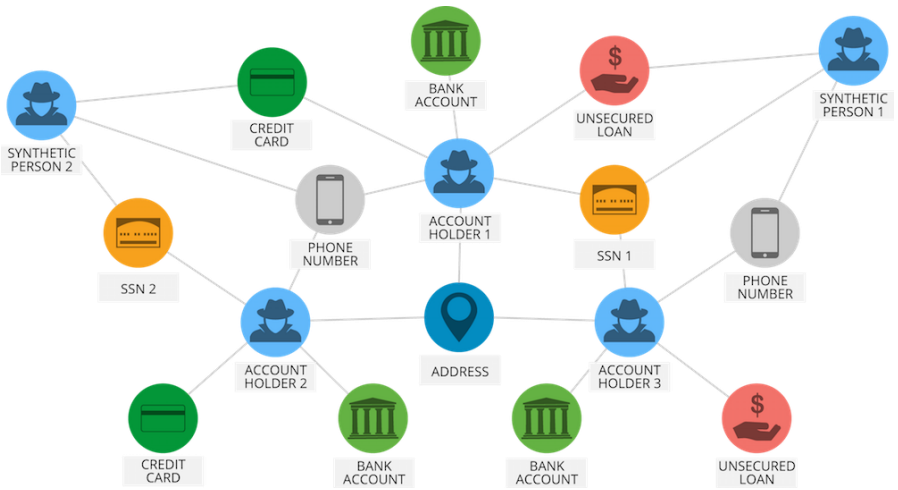
Συνολικά, η χρήση των ΒΔΓ στον δημόσιο τομέα, στα πλαίσια του ψηφιακού μετασχηματισμού όλων των κρατών -ανεξαρτήτως σε ποιο στάδιο βρίσκονται- αποτελεί μια υπαρκτή λύση σε μια σειρά προβλημάτων που βασίζονται κυρίως στον τρόπο οργάνωσης και ανάλυσης των δεδομένων. Είτε αυτά αφορούν αρχές προστασίας, είτε τις δημόσιες υπηρεσίες είτε την εξυπηρέτηση των πολιτών, ο στόχος μείωσης της πολυπλοκότητας της ανάλυσης και διαχείρισης τους θα επιτευχθεί μέσω των πιο απλών λύσεων όπως η τεχνολογία γράφων, με τη σύγχρονη προσέγγιση της, τις ΒΔΓ και όχι από την ύπαρξη τοπικών προσεγγίσεων, που δεν προσφέρουν καμία ευελιξία, ούτε και φιλικά προς το χρήστη -και τον διαχειριστή- περιβάλλοντα.

Οικονομικές Απάτες

Οι τράπεζες και οι ασφαλιστικοί όμιλοι χάνουν σημαντικά ποσά κάθε χρόνο λόγω οικονομικών απατών. Οι παραδοσιακές μέθοδοι ανίχνευσης της απάτης συνήθως αποτυγχάνουν να ελαχιστοποιήσουν τις απώλειες καθώς ακολουθούν μεθοδολογίες που είναι ευάλωτες σε false positive & false negative αποτελέσματα. Από την άλλη πλευρά όσοι επιδιώκουν να κάνουν τέτοιες απάτες γνωρίζουν συνήθως τις μεθόδους που ακολουθούν οι αρμόδιες υπηρεσίες και προσαρμόζονται κατάλληλα για να τις παρακάμψουν [24] [25] [26]. Η τεχνολογία γράφων και οι ΒΔΓ ανοίγουν νέες επιλογές και μεθοδολογίες για την ανίχνευση και επίλυση οικονομικών απατών ακόμη και σε πολύ περίπλοκες περιπτώσεις. Προφανώς καμιά μέθοδος δεν είναι απόλυτα και καθολικά αποτελεσματική αφού είναι μια διαρκής αλληλεπίδραση με τους οικονομικούς εγκληματίες οι οποίοι θα προσαρμοστούν, θα κάνουν πιο πολύπλοκες κινήσεις κοκ. Στην πραγματικότητα η λύση που προσφέρει η τεχνολογία γράφων είναι οι σχέσεις που θα δημιουργήσει μεταξύ των οντοτήτων και φυσικά η εύκολη προς τον χρήστη εκτύπωση του γράφου. Μάλιστα το Neo4j έχει δημοσιεύσει σχετικό White Paper όπου μεταξύ άλλων αναφέρει μια πιο συγκεκριμένη περίπτωση χρήσης για την ανάλυση πιθανής οικονομικής απάτης και μια μεθοδολογία που ακολούθησε πραγματική επενδυτική εταιρία χωρίς φυσικά να αναφέρονται τα πραγματικά στοιχεία.

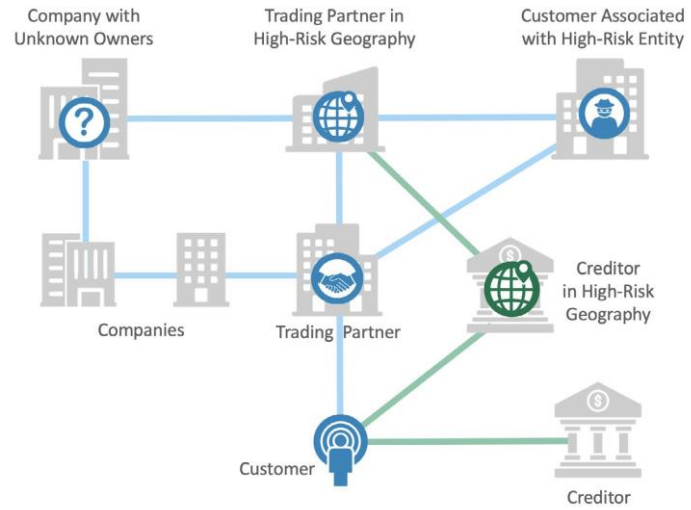
Έστω ότι μια εταιρία επενδύσεων, συγκεντρώνει έναν μεγάλο όγκο δεδομένων για έναν πελάτη που πρέπει να αναλυθούν, πριν γίνει αποδεκτή μια σημαντική συναλλαγή. Χωρίς τη χρήση των ΒΔΓ ο αναλυτής θα έπρεπε να εισάγει σε μια σχεσιακή ΒΔ όλα τα δεδομένα και να τρέξει επαναλαμβανόμενα ερωτήματα για να βρει τις σχέσεις μεταξύ τους. Αντιθέτως, μέσω του Neo4j η διαδικασία είναι σημαντικά αυτοματοποιημένη και απλούστερη και φυσικά στο τέλος θα αναπαρασταθεί πλήρως ο γράφος στον αναλυτή. Έτσι η πιθανότητα να χάσει τις λεπτομέρειες, τις σχέσεις μεταξύ συναλλαγών, οντοτήτων, προσώπων που είναι πιθανώς αβάσιμες και δείχνουν ότι υπάρχει θέμα απάτης, μειώνεται σημαντικά. Μάλιστα ανώτερα στελέχη – αναλυτές του οργανισμού, δήλωσαν πως ο χρόνος ανάλυσης των συναλλαγών και των δεδομένων μειώθηκε στο μισό .

Στην παρακάτω εικόνα (11) βλέπουμε ένα πολύπλοκο σχήμα – μεθοδολογία οικονομικής απάτης όπου στο κέντρο της βρίσκονται τρία φυσικά πρόσωπα. Μέσω των γράφων βρίσκουμε τις κινήσεις – σχέσεις τους με τράπεζες, διευθύνσεις, κινητά τηλέφωνα, πρόσωπα «βιτρίνες» για τις παράνομες δουλειές κλπ. Με την αναπαράσταση υπό μορφή γράφου μπορούμε να κατανοήσουμε πως λειτουργεί ένα ολόκληρο κύκλωμα τέτοιων ενεργειών, άρα και να λύσουμε μια πιθανή υπόθεση.



Εικόνα 11: Χρήση των ΒΔΓ για εξιχνίαση οικονομικών εγκλημάτων

Όπως βλέπουμε και στο παρακάτω σχήμα (Εικόνα 12) Στο ίδιο μήκος κινείται και η χρήση των ΒΔΓ ενάντια στο ξέπλυμα χρήματος με τα Anti-Money Laundering μοντέλα να μεταφέρονται σε native ΒΔΓ. Για τον αντίστοιχο λόγο της ανάγκης εύρεσης όλων των συνδέσεων και χρηματικών κινήσεων των υπόπτων και την αναπαράσταση των δεδομένων σε μορφή ευνοϊκή προς τον χρήστη – ελεγκτή, διαδόθηκε η χρήση των ΒΔΓ (και από διεθνείς οργανισμούς όπως ο European Financial Intelligence Unit). Για το ξέπλυμα χρήματος σημειώνεται πως η χρήση των ΒΔΓ είναι αποδοτική για την πλειοψηφία των μορφών που αυτό μπορεί να έχει, δηλαδή, την μέθοδο του μυρμηγκιού (smurfing), το λαθρεμπόριο νομισμάτων (Bulk Cash Smuggling), την αγοραπωλησία πολύτιμων αντικειμένων, εταιριών, ομόλογων, την αλλεπάλληλη μεταφορά χρημάτων, την ίδρυση και πτώχευση ασφαλιστικών εταιριών κλπ.



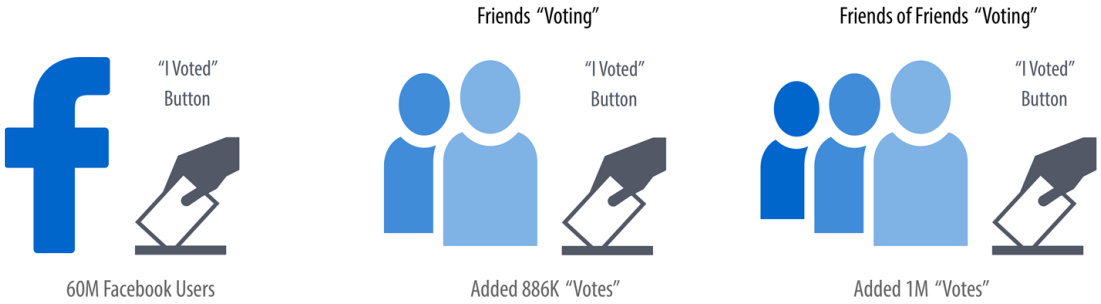
Εικόνα 12: Χρήση των ΒΔΓ για την εξιχνίαση εγκλημάτων Money-Laundering

Μηχανική Μάθηση

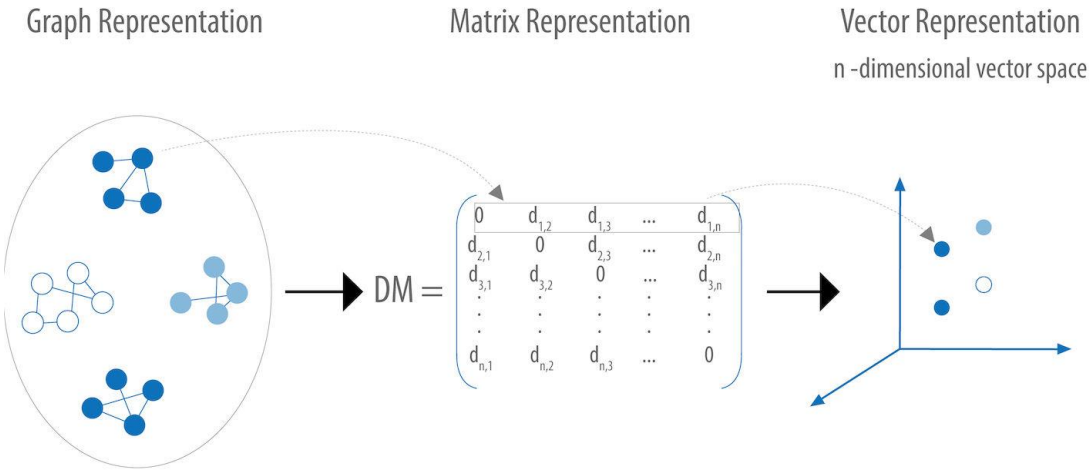
Η μηχανική μάθηση χρησιμοποιεί τους αλγορίθμους γράφων και τις ΒΔΓ για να εκπαιδεύσει το λογισμικό μέσω παραδειγμάτων, ώστε να μπορεί το πρόγραμμα να προβλέπει αποτελέσματα με όλο και μεγαλύτερη ακρίβεια βάσει συμπεριφορών, προγραμμάτων, αλγορίθμων κοκ. Όσο περισσότερα τα δεδομένα που θα έχει στη διάθεση του το πρόγραμμα εκπαίδευσης, τόσο πιο «έξυπνο» στις προβλέψεις του θα γίνει [27]. Μέσω, λοιπόν, των γράφων και κυρίως μέσω τις συνδεσιμότητας και των σχέσεων μεταξύ των οντοτήτων, οι αλγόριθμοι προσπέλασης γράφων προσφέρουν στην μηχανική μάθηση αυτό ακριβώς που χρειάζεται. Νέα

σχετικά δεδομένα προς εκπαίδευση. Όπου αξιολογώντας τις οντότητες και τις σχέσεις τους, θα μπορεί ο αλγόριθμος να εκπαιδεύεται όλο και καλύτερα. Ένα παράδειγμα είναι το e-shop το οποίο χρησιμοποιεί τόσο τεχνολογία γράφων για τους αλγορίθμους στα συστήματα προτάσεων, και μηχανική μάθηση για να προβλέψει το τι θέλουν να δουν οι νέοι χρήστες που θα μπουν στην σελίδα (βάσει των cookies, αναζητήσεων σε παρόμοια sites κοκ.). Μάλιστα η ψηφιακή βοηθός Alexa της Amazon χρησιμοποιεί επίπεδα ανάλυσης συμπεριφοράς τα οποία αναλύουν διάφοροι αλγόριθμοι μηχανικής μάθησης ώστε να βελτιώσουν τις προτάσεις και το περιεχόμενο που βλέπει ο χρήστης, ή του προτείνει η ψηφιακή βοηθός. Σημειώνεται δε, πως για να είναι πραγματικά αποδοτικός ο αλγόριθμος μηχανικής μάθησης θα πρέπει τα δεδομένα των γράφων να του δοθούν με τον κατάλληλο τρόπο. Σε αυτό βοηθάει η μηχανική των χαρακτηριστικών των γράφων, όπου μας βοηθάει να παίρνουμε τα πρωτόλεια δεδομένα που μας δίνει για παράδειγμα ο αλγόριθμος προσπέλασης γράφων, και να τα μετατρέπουμε έτσι ώστε ο αλγόριθμος μηχανικής μάθησης να τα αξιοποιήσει, να τα κατηγοριοποιήσει κλπ.

Η θεωρία γράφων, εμπνευσμένη από την ίδια την αλληλεπίδραση μεταξύ των ανθρώπων, αποδεικνύει πως οι σχέσεις μεταξύ των οντοτήτων καθορίζουν σε μεγάλο βαθμό τη συμπεριφορά (σχετικά χαρακτηριστικά κλπ.) της εκάστοτε οντότητας. Αυτό αξιοποιείται στην μηχανική μάθηση πλήρως. Το 2012 στην έρευνα «A 61-million-person experiment in social influence and political mobilization» (Εικόνα 13) αποδείχθηκε πως εάν ένας άνθρωπος θέλει να ψηφίσει στις εκλογές, η πιθανότητα να ψηφίσουν οι φίλοι του, η οικογένεια του κλπ. αυξάνεται.



Εικόνα 13: Έρευνα «A 61-million-person experiment in social influence and political mobilization».



Εικόνα 14: Από τους αλγόριθμους γράφων στους αλγόριθμους επιλογής και μετατροπής των αποτελεσμάτων και τέλος σε δεδομένα υπό τη μορφή διανυσμάτων.

2.3 Βάσεις Δεδομένων Ψηφιακής Ασφάλειας σε Γράφο

Η διαρκής προσαρμογή, η πρόληψη και η έγκαιρη αντίδραση στις προκλήσεις ενός διαρκούς μεταβαλλόμενου περιβάλλοντος, αποτελούν το ισχυρότερο θεμέλιο για την αποτελεσματική

διαμόρφωση μιας στρατηγικής κυβερνοασφάλειας, σε δημόσιους και ιδιωτικούς οργανισμούς. Όπως είναι λογικό όσο περισσότερο στηρίζεται η λειτουργία ενός οργανισμού στον ψηφιακό κόσμο, τόσο πιο επιτακτική είναι η ανάγκη για την προστασία μιας υποδομής (φυσικής, ψηφιακής, διαδικασιών κλπ.) από ενδεχόμενες ψηφιακές απειλές. Οι δημόσιοι και ιδιωτικοί οργανισμοί και οι υποδομές τους αποτελούν στόχοι πληθώρας κυβερνοεπιθέσεων. Από απειλές που προέρχονται από μεμονωμένους εγκληματίες, μέχρι επιθέσεις φερόμενες ως απόρροια ενεργειών τρίτων κρατών, το περιβάλλον κυβερνοαπειλών είναι διαρκώς μεταβαλλόμενο, οδηγώντας σε μια εγγενή αδυναμία άμεσης προστασίας του. Οι υπάρχουσες συνθήκες που οξύνθηκαν μέσα στην πανδημία του κορονοϊού, επιβεβαιώνουν την ανάγκη για ένα μοντέλο κυβερνοασφάλειας, το οποίο θα προσαρμόζεται και θα ανανεώνεται. Σημειώνεται δε, πως μόνο το 2020 σύμφωνα με έρευνα της εταιρίας κυβερνοασφάλειας F5 Networks που δραστηριοποιείται σε Ελλάδα και Κύπρο, οι επιθέσεις phishing αυξήθηκαν κατά 15% σε σχέση με το 2019, ενώ τα περιστατικά που καταγγέλθηκαν αυξήθηκαν κατά 220% συγκριτικά με τον ετήσιο μέσο όρο κατά τη διάρκεια της πανδημίας. Επίσης, οι μεγάλες αυξήσεις phishing συνέπεσαν χρονικά με τα κατά τόπους lockdown, κάτι που δείχνει την άμεση σχέση της αύξησης των επιθέσεων με την εξ' αποστάσεως εργασία. Ταυτόχρονα στην Ευρώπη, ουκ ολίγες φορές μέσα στην πανδημία, πραγματοποιήθηκαν κυβερνοεπιθέσεις σε νοσοκομεία (Phishing, Ransomware κλπ.) και άλλες μονάδες υγειονομικού ενδιαφέροντος.

Οι επιθέσεις στον κυβερνοχώρο καθώς και η κυβερνοασφάλεια, πραγματοποιούνται σε περίπλοκα περιβάλλοντα οργανισμών με πολλούς παράγοντες να επηρεάζουν την επιτυχία μιας επίθεσης ή μιας απόκρουσης της. Ενδεικτικά είναι τα παραδείγματα, της τοπολογίας του δικτύου, των ρυθμίσεων που κάνει ο διαχειριστής, των ευπαθειών της υποδομής, των συστημάτων ανίχνευσης εισβολών κοκ. Εν έτη 2021 αυτό που λείπει από τους αναλυτές κυβερνοασφάλειας, δεν είναι τα σύγχρονα ψηφιακά μέσα άμυνας όπως firewall, ή καταγραφών για αλλαγές και ύποπτες κινήσεις. Λείπει πρώτον, η εκπαίδευση των εργαζόμενους και του κοινού ευρέως στην ψηφιακή ασφάλεια (και η συμμόρφωση του κατ' επέκταση σε πολιτικές και διαδικασίες ασφάλειας) και δεύτερον τα εργαλεία ανάλυσης των εκατοντάδων χιλιάδων -ή πολύ περισσότερων- καταγραφών, ώστε ο υπεύθυνος κυβερνοασφάλειας να μπορέσει να αναλύσει με αποδοτικό τρόπο άμεσα τυχόν νέες απειλές, ευπάθειες ή και περιστατικά επιθέσεων. Φυσικά, μέσα από την βελτίωση της ανάλυσης των απειλών, ευπαθειών και επιθέσεων, οι επιστήμονες της κυβερνοασφάλειας θα βελτιώσουν και τις υπάρχουσες πολιτικές και διαδικασίες των οργανισμών, συμβάλλοντας και μέσω αυτού στη μείωση του χάσματος μεταξύ εργαζόμενων (και λοιπών χρηστών), με τα θέματα ψηφιακής ασφάλειας [28].

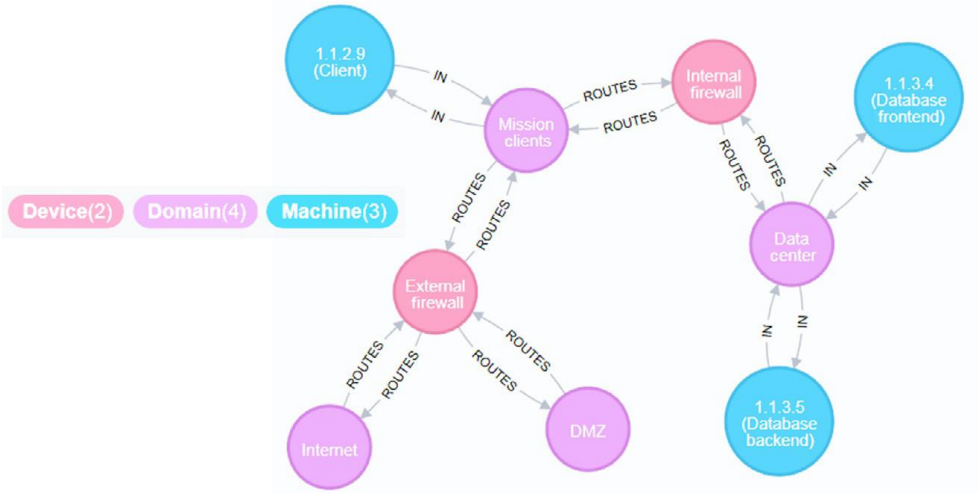
Σε αυτό το πεδίο έρχονται οι ΒΔΓ, και η τεχνολογία γράφων, να προσφέρουν στην βελτίωση τόσο σε επίπεδο αποδοτικότητας όσο και σε επίπεδο ανάλυσης των θεμάτων ψηφιακής ασφάλειας. Στην ουσία, η χρήση τους θα προσφέρει στους διαχειριστές, αναλυτές και συνολικά στους οργανισμούς έναν σύνδεσμο μεταξύ του πολύπλοκου περιβάλλοντος, των χιλιάδων πληροφοριών με την ανωτέρου επιπέδου εξαγωγή γνώσεων. Για παράδειγμα, ένα λογισμικό SIEM (Security Information & Event Management) θα βοηθήσει (με την προϋπόθεση ότι ρυθμίστηκε ορθά) στην καταγραφή δεδομένων με συγκεκριμένο τρόπο. Ωστόσο, τα δεδομένα αυτά δεν παύουν να είναι αποκομμένες πτυχές μιας συνολικότερης πληροφορίας που πρέπει ο διαχειριστής να γνωρίζει. Οι ΒΔΓ θα συμβάλουν στη σύνδεση των επιμέρους πληροφοριών.

Πέρα από τα συστήματα ελέγχου πρόσβασης που άπτονται των διαδικασιών και πολιτικών ασφάλειας ενός οργανισμού, οι ΒΔΓ μπορούν να χρησιμοποιηθούν και για την δικτυακή προστασία της υποδομής, την συμμόρφωση των πολιτικών και διαδικασιών στα διάφορα πρότυπα και φυσικά στην ανάλυση και διαχείριση επικινδυνότητας.

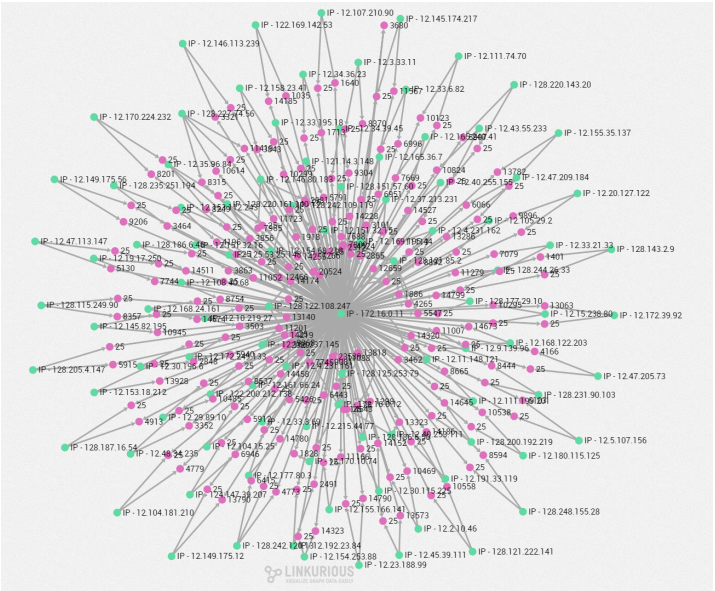
2.3.1 Περίπτωση Χρήσης – Εισβολή στο Δίκτυο ενός Οργανισμού

Ένας οργανισμός δέχεται εισβολή στο δίκτυο του. Η εισβολή ξεκινά μέσω του δικτύου συνεργαζόμενης εταιρίας που έχει πρόσβαση σε δεδομένα του οργανισμού. Η εισβολή ξεκινά από Trojan λογισμικό. Η εισβολή στην συνεργαζόμενη εταιρία είναι επιτυχής και έτσι το δίκτυο της γίνεται κόμβος εκκίνησης για την επίθεση στον οργανισμό. Έστω ότι οι επιτιθέμενοι καταφέρνουν να παραβιάσουν το δίκτυο του οργανισμού (μεσολαβούν προφανώς και άλλα

βήματα) και να πάρουν μη εξουσιοδοτημένη πρόσβαση σε κρίσιμα για αυτόν δεδομένα. Η ουσία είναι πως υπήρξαν συγκεκριμένα, αριθμήσιμα βήματα για την επίτευξη της επίθεσης, με αντίστοιχες δυνατότητες του οργανισμού να την αναγνωρίσει και να την αποτρέψει. Τα επίπεδα για μια τέτοια ενέργεια από πλευράς οργανισμού είναι δύο. Η σωστή ρύθμιση για την καταγραφή δεδομένων που σχετίζονται με την ακεραιότητα, εμπιστευτικότητα και διαθεσιμότητα ενός πληροφοριακού συστήματος, που θέλουμε να προστατέψουμε. Και η αναγνώριση – ανάλυση των δεδομένων που καταγράφονται από πλευράς του διαχειριστή. Σίγουρα τα συστήματα SIEM με την σωστή ρύθμισή τους όπως είπαμε και παραπάνω θα συμβάλουν σε αυτήν την διαδικασία. Ωστόσο, η χρήση των ΒΔΓ, και μάλιστα με συγκεκριμένο τρόπο (ώστε να παίρνουν αυτοματοποιημένα τα μορφοποιημένα δεδομένα των SIEM, να τα φτιάχνουν με τη μορφή γράφων, να αναπαριστούν συνδέσεις μεταξύ των τύπων γεγονότων, των αγαθών του πληροφοριακού συστήματος κοκ.) θα περιορίσει την οποιαδήποτε ολιγωρία από πλευράς διαχειριστή ή προσωπικού ασφάλειας. Μάλιστα θα συμβάλει στην βέλτιστη ανάλυση μιας προσπάθειας επίθεσης και πιθανώς στην εύρεση ψηφιακών πειστηρίων σε περίπτωση που ο οργανισμός θελήσει να κινηθεί νομικά [29].



Εικόνα 15: Αναπαράσταση μέσω Neo4j για Firewall Blocking Responses σε ένα δίκτυο.



Εικόνα 16: UDP Storm Attack – Αναπαράσταση των δεδομένων από Firewall σε γράφο [30].

Αξίζει να σημειωθεί πως βασισμένο σε Neo4j είναι το λογισμικό CyGraph [31], υλοποιημένο από επιστήμονες της κυβερνοασφάλειας, με στόχο να φτιάχνουν μοντέλα γράφων για την εκτέλεση επιθέσεων. Το λογισμικό, παίρνει ένα ευρύ φάσμα πληροφοριών που ξεκινούν από τα αμιγώς τεχνικά όπως τοπολογία δικτύου, αισθητήρες, SIEM Data κλπ. και φτάνει ως τους στόχους που μπορεί να έχει μια επίθεση, τις επιπτώσεις στον οργανισμό κλπ. Έτσι παράγει γράφους με κατάλληλες συνδέσεις για μελέτη από τους διαχειριστές και υπεύθυνους ασφάλειας των οργανισμών. Πιο συγκεκριμένα τα τέσσερα επίπεδα της Εικόνας 17 είναι:

1. Υποδομή Δικτύου: Πως δομείται το δίκτυο, τοπολογία (φυσική και μη), αισθητήρες, δεδομένα καταγραφής κλπ.
2. Κατάσταση στην Ψηφιακή Υποδομή: Πως έχει διαμορφώσει ο διαχειριστής το δίκτυο και το πληροφοριακό σύστημα, τι ρυθμίσεις έχει κάνει, τι κανόνες έχει βάλει και ποιες είναι οι ευπάθειες του Π.Σ. Δηλαδή ποια είναι τα στοιχεία εκείνα εντός του δικτύου που θα επηρεάσουν την έκβαση μιας επίθεσης.
3. Ψηφιακές Απειλές: Ο οργανισμός θα θέσει τις πιθανές απειλές και τους φορείς τους. Για παράδειγμα, είναι απειλή μια συνεργαζόμενη εταιρία που έχει πρόσβαση σε μέρος του δικτύου του πληροφοριακού συστήματος;
4. Στόχοι και Εξαρτήσεις: Μια επίθεση έχει συγκεκριμένους στόχους. Παράλληλα, ένα πληροφοριακό σύστημα λειτουργεί για να φέρει εις πέρας συγκεκριμένες εργασίες του οργανισμού. Ποια η σύνδεση αυτών; Ποια τα ψηφιακά στοιχεία που θα επηρεάσουν την ομαλή λειτουργία του οργανισμού; Αυτές τις πληροφορίες θα εντάξει ο διαχειριστής στο CyGraph στο τελικό στάδιο.

Έπειτα το λογισμικό θα εξάγει λαμβάνοντας υπόψη όλα αυτά τα δεδομένα, κατάλληλο γράφο προς μελέτη και φυσικά καθώς είναι υλοποιημένο σε Neo4j η όλη αυτή εργασία θα είναι σε σημαντικά μειωμένο χρόνο σε σχέση με τη χρήση άλλης ΒΔ ή και άλλης ΒΔΓ.

Εικόνα 17: Το εύρος πληροφοριών που λαμβάνει υπόψη για την δημιουργία ενός γράφου πιθανής επίθεσης το CyGraph [32].

2.3.2 Compliance – Συμμόρφωση σε κανονισμούς ασφάλειας και προσωπικά δεδομένα

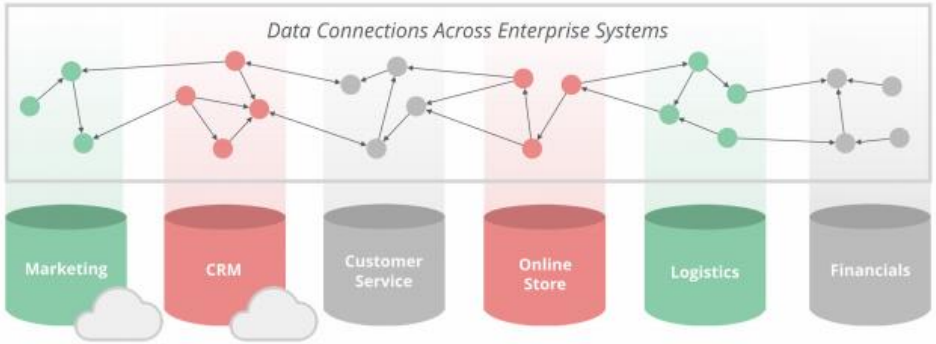
Οι οργανισμοί παγκοσμίως προσπαθούν να ακολουθήσουν τους σύγχρονους κανονισμούς, όπως για παράδειγμα για την προστασία προσωπικών δεδομένων (GDPR), για τις πιστοποιήσεις ISO κλπ. Οι αυστηροί κανόνες κάθε κανονισμού, δημιουργούν τη δυσκολία στον οργανισμό να τους ακολουθεί πλήρως και συστηματικά, ιδίως αν δεν έχει ορίσει σαφείς πολιτικές και

διαδικασίες, συμμορφωμένες στο αντίστοιχο πρότυπο. Ωστόσο, η σημασία για την πλήρη συμμόρφωση στους κανονισμούς ασφάλειας είναι μεγάλη για έναν σύγχρονο οργανισμό, πολλές φορές αναγκαία για την ύπαρξή του [33] [34]. Η εταιρία PwC αναφέρει πως 92% των πολυεθνικών επιχειρήσεων θέτει την συμμόρφωση στον GDPR, σαν ύψιστη προτεραιότητα του οργανισμού. Πως όμως ένας οργανισμός που χρησιμοποιεί για παράδειγμα προσωπικά δεδομένα πελατών, υπαλλήλων και άλλων ομάδων, θα μπορέσει να ακολουθήσει τον κανονισμό, ακόμη και αυστηρά ορισμένες πολιτικές και διαδικασίες προς αυτήν την κατεύθυνση, όταν η χρήση των δεδομένων διαπερνάται από πλήθος εφαρμογών; Σε αυτό το ερώτημα οι ΒΔΓ απαντούν πως με την ανάλυση των «διαδρομών» που θα κάνουν τα δεδομένα προς προστασία, στην ψηφιακή υποδομή ενός οργανισμού, ο οργανισμός έχοντας ορίσει παράλληλα διαδικασίες και πολιτικές, θα μπορεί να επιτηρεί την ορθή λειτουργία αυτών, και να αποδεικνύει τη συμμόρφωση του εν προκειμένω στον GDPR αλλά και γενικότερα. Πιο συγκεκριμένα, κάθε λογισμικό του οργανισμού που χρησιμοποιεί προσωπικά δεδομένα θα πρέπει να εγγράφει τις λεπτομέρειες στην ΒΔΓ και φυσικά από που τα παρέλαβε. Η ΒΔΓ θα μπορεί να οπτικοποιεί τους γράφους και τα δεδομένα δίνοντας σαφείς απαντήσεις ανάλογα με τους χρήστες που θα έχουν πρόσβαση, δηλαδή ελεγκτές του κανονισμού που θα πρέπει να δουν απόδειξη της συμμόρφωσης στον κανονισμό, υπαλλήλους του οργανισμού που θα επιτηρούν την συμμόρφωση στον κανονισμό εύκολα με αυτόν τον τρόπο και φυσικά, ατομικά, χρήστες που θα επιθυμούν να δουν τι δεδομένα έχει για αυτούς ο οργανισμός [35] [36].

Το Neo4j έχει υλοποιήσει λογισμικό «Neo4j Privacy Shield» όπου στόχο έχει τη σύνδεση των επιμέρους προσωπικών δεδομένων μέσω των εφαρμογών του Πληροφοριακού Συστήματος όπου γίνεται χαρτογράφηση: 1) της περιοχής των δεδομένων (φυσικά και ψηφιακά) 2) των συστημάτων και εφαρμογών που χρησιμοποιούν τα δεδομένα 3) του τρόπου και χρόνου που χρησιμοποιούνται τα δεδομένα 4) των χρηστών που έχουν πρόσβαση σε αυτά 5) των δικαιωμάτων που έχει ο οργανισμός ή ομάδες χρηστών εντός του πάνω στα δεδομένα και πως αυτά τα δικαιώματα αποκτήθηκαν 6) των μεταφορών των δεδομένων (βλ. από εφαρμογή σε εφαρμογή). Παράλληλα, το λογισμικό αυτό προσφέρει μια σειρά από εργαλεία και επιλογές εξαγωγής γράφων όπως φαίνεται και στην Εικόνα 20:

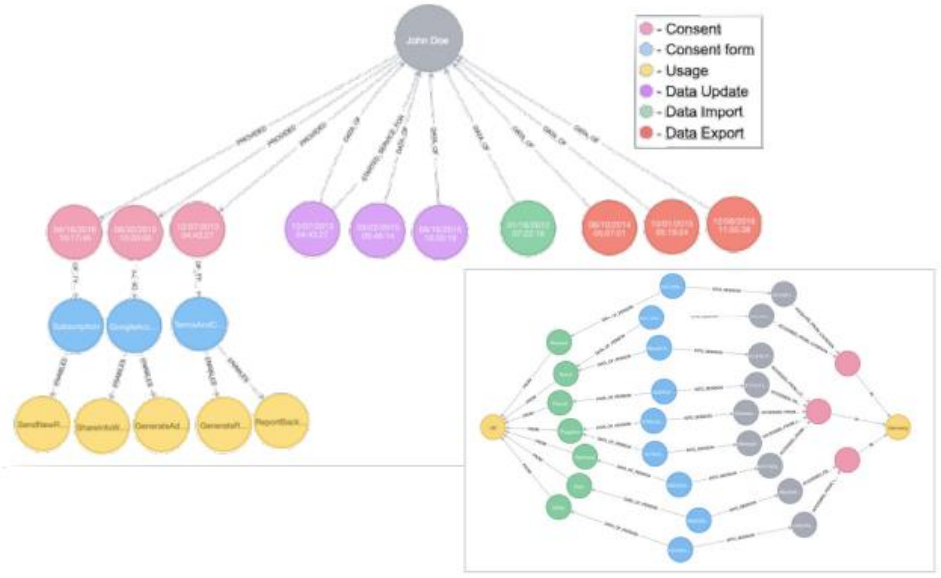
- Role-based Dashboards: Ανάλογα με τα δικαιώματα που έχει ο κάθε χρήστης της πλατφόρμας, λαμβάνει διαφορετικά δεδομένα τόσο υπό μορφή γράφων όσο και αναλυτικά δεδομένα.
- Connected Data Visualizations: Για κάθε κλειδί που ψάχνουμε για παράδειγμα ένα φυσικό πρόσωπο, παίρνουμε όλα τα δεδομένα, συνδεδεμένα όπως ακριβώς λειτουργούν οι ΒΔΓ με τις οντότητες και τις σχέσεις. Επιπρόσθετα εξάγονται και διαγράμματα, και χάρτες προσωπικών δεδομένων του χρήστη.
- Data Movement Patterns: Με την χρήση τεχνητής νοημοσύνης η πλατφόρμα αναγνωρίζει και παρουσιάζει πως τα προσωπικά δεδομένα μεταφέρονται στην υποδομή ενός οργανισμού.
- Threat and Pattern Detection: Με αυτοματοποιημένο τρόπο παρουσιάζονται σε κάθε βήμα πιθανές ευπάθειες και απειλές ως προς τα προσωπικά δεδομένα και τη χρήση του με σκοπό την άμεση ανταπόκριση από πλευράς οργανισμού.
- Consent Management: Εργαλείο διαχείρισης της συγκατάθεσης των χρηστών για τους τρόπους που θα χρησιμοποιούνται τα προσωπικά τους δεδομένα. Συμβάλει στην εύκολη εποπτεία των συναινέσεων των χρηστών, την εξαγωγή αναλυτικών δεδομένων κλπ.
- Data Usage Reports: Ποιοι, Πως, Πότε και Γιατί είχαν πρόσβαση στα προσωπικά δεδομένα των χρηστών.
- Data Archival Reports: Ενημέρωση για το πότε πρέπει προσωπικά δεδομένα να αρχειοθετηθούν ή να διαγραφούν.
- Proactive Alerts: Ενημέρωση των διαχειριστών όταν παρατηρούνται ασυνήθιστες συνδέσεις μεταξύ των δεδομένων ή τα δεδομένα χρησιμοποιούνται με τρόπο εκτός των πολιτικών και διαδικασιών του κανονισμού.
- What-if Analysis: Η πλατφόρμα δοκιμάζει πειραματικά, σενάρια επίθεσης, αλλαγές στον τρόπο πρόσβασης τα δεδομένα και άλλες περιπτώσεις που θα επηρεάσουν την συμμόρφωση στους κανονισμούς για τη χρήση προσωπικών δεδομένων.

Όπως βλέπουμε και στην παρακάτω εικόνα (18) Η χρήση των προσωπικών δεδομένων σε έναν οργανισμό, εμπεριέχεται σε παραπάνω της μιας, εφαρμογές,. Έτσι με τη χρήση των ΒΔΓ και βοηθητικών λογισμικών, και με τις ανάλογες εγγραφές σε αυτές, υπάρχει σαφής διαφάνεια στην χρήση των προσωπικών δεδομένων των χρηστών. Πιο συγκεκριμένα, πρέπει αρχικά να έχουμε ορίσει σαφώς τα αγαθά του Πληροφοριακού Συστήματος που αλληλοεπιδρούν με προσωπικά δεδομένα και έπειτα να χτίσουμε το λογικό μοντέλο των συνδέσεων. Στη συνέχεια θα χρησιμοποιήσουμε εφαρμογή για την εγγραφή των δεδομένων βάσει του μοντέλου στη ΒΔΓ. Τέλος θα μπορούμε βάσει των δικαιωμάτων πρόσβασής μας να εξάγουμε αποτέλεσμα σαν αυτό της εικόνας 19.

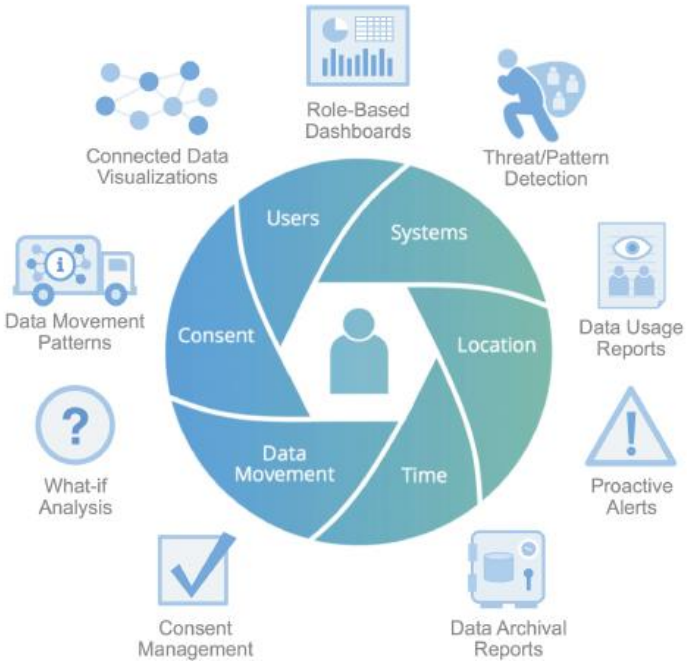


Εικόνα 18: Χρήση προσωπικών δεδομένων σε λειτουργίες και εφαρμογές ενός οργανισμού.

Αν για παράδειγμα ο χρήστης επιθυμεί να μάθει που, πως, πότε χρησιμοποιήθηκαν τα προσωπικά του δεδομένα από τις εφαρμογές του οργανισμού, τότε οι ΒΔΓ θα εξάγουν έναν γράφο αντίστοιχο με αυτόν της εικόνας 19.



Εικόνα 19: Μοντελοποίηση χρήσης προσωπικών δεδομένων σε ΒΔΓ.



Εικόνα 20: Το Neo4j Privacy Shield προσφέρει πλήθος εργαλείων και επιλογών για την βέλτιστη εξαγωγή γράφων, σύμφωνα με τις ανάγκες των χρηστών.

Εν τέλει, η χρήση των ΒΔΓ στην προσπάθεια των οργανισμών να ακολουθούν τους κανονισμούς για τη χρήση των προσωπικών δεδομένων των χρηστών, είναι πλήρως αποδοτική και διευκολύνει τη διαχείριση και την εποπτεία σε όλα τα επίπεδα. Η συμμόρφωση ενός οργανισμού στους εγχώριους και διεθνείς κανονισμούς, στις πολιτικές και διαδικασίες ασφάλειας της χρήσης προσωπικών δεδομένων, και κατ' επέκταση και δημόσια απόδειξη των παραπάνω, συμβάλει στο κέρδος της εμπιστοσύνης των χρηστών (εσωτερικών και εξωτερικών), στην αποδοτικότητα του οργανισμού που με γρήγορο και εύκολο τρόπο θα μπορεί να διαχειρίζεται και να εποπτεύει την ορθή χρήση των δεδομένων και στη μείωση του ρίσκου για οποιουδήποτε είδους πρόστιμο και ποινή για μη συμμόρφωση σε κανονισμούς.

2.3.3 Ανάλυση και Διαχείριση Επικινδυνότητας Πληροφοριακών Συστημάτων

Η χρήση των ΒΔΓ σε αυτό το πεδίο της ψηφιακής ασφάλειας, διευκολύνει και αυτοματοποιεί πολλά από τα βήματα για την συμμόρφωση των οργανισμών σε πρότυπα και κανονισμούς όπως αυτά του ISO 27000. Σε έναν σύγχρονο οργανισμό, τα αγαθά ενός Π.Σ αλληλοεπιδρούν και συνδέονται μεταξύ τους με ποικίλους τρόπους. Δοθέντος ενός Π.Σ και των αγαθών του, μπορούμε να αναπαριστήσουμε τις συνδέσεις μεταξύ των αγαθών στις ΒΔΓ ώστε να βελτιώσουμε την ανάλυση απειλών και ευπαθειών και φυσικά των τρόπων μέσω των οποίων ένας επιτιθέμενος μπορεί να παραβιάσει ένα αγαθό, με οποιοδήποτε σημείο εκκίνησης. Σημειώνεται δε, πως η παραβίαση ενός αγαθού δηλαδή, η διατάραξη, της εμπιστευτικότητας, ακεραιότητας, διαθεσιμότητάς του, δεν απαιτεί πάντα την δικτυακή παραβίαση, περίπτωση χρήσης που είδαμε παραπάνω. Η μελέτη με τη χρήση των ΒΔΓ μπορεί να διευκολυνθεί, να εξειδικευθεί (για παράδειγμα, η ανάλυση της επικινδυνότητας προϋποθέτει την ανάλυση και στα τρία επίπεδα, εμπιστευτικότητα, ακεραιότητα, διαθεσιμότητα του αγαθού) και εν τέλει να μας δώσει τις κατάλληλες απαντήσεις. Ταυτόχρονα, λόγω της ευελιξίας του σχήματος, μπορούμε πέρα από τα αγαθά και τις συνδέσεις τους, να ορίσουμε απειλές, ευπάθειες, χρήστες, μονοπάτια επίθεσης και πολλά άλλα, που ο ορισμός τους δεν ενδείκνυται σε παραδοσιακά λογισμικά ανάλυσης και διαχείρισης επικινδυνότητας είτε είναι αρκετά πολύπλοκος και χρονοβόρος. Ουσιαστικά οι ΒΔΓ

θα μας δώσουν τρία βασικά στοιχεία: 1) Εμβάθυνση στην ανάλυση 2) Αποδοτικότητα και Γρήγορη διαχείριση των δεδομένων 3) Φιλική προς το χρήστη ανάλυση των αποτελεσμάτων.

2.3.4 Έρευνα

Κλείνοντας την ενότητα των περιπτώσεων χρήσης στην κυβερνοασφάλεια, θέλουμε να αναλύσουμε το πεδίο έρευνα, ψηφιακή ασφάλεια και ΒΔΓ. Οι περιπτώσεις χρήσης φυσικά μπορούν να είναι τόσες, όσες και οι ιδέες ή τα ερευνητικά πεδία των επιστημόνων. Οι ερευνητές στην ψηφιακή ασφάλεια εστιάζουν σε τέσσερα σημεία: 1) Στην εύρεση ευπαθειών σε υποδομές και τεχνολογίες σε μεγάλη κλίμακα. 2) Στην εύρεση νέων τρόπων ψηφιακής ασφάλειας βάσει των παραπάνω. 3) Στην βελτίωση των ήδη υπαρχόντων ώστε να γίνουν πιο αποδοτικοί σε όλα τα επίπεδα 4) Στην συγκρότηση νέων κανονισμών, πολιτικών, διαδικασιών και προτύπων που θα συμβάλουν σε όλα τα παραπάνω. Επομένως ιδίως σε αυτό το πεδίο της ψηφιακής ασφάλειας, η χρήση των ΒΔΓ δεν θα γίνεται εξ' ολοκλήρου, αλλά στον βαθμό που θα εξυπηρετεί την συνέχιση των τεσσάρων αυτών σημείων. Στην ουσία η χρήση τους στην Έρευνα, εξειδικεύεται βάσει της χρήσης τους σε όλα τα υπόλοιπα πεδία ψηφιακής ασφάλειας.

Στη βιβλιογραφία υπάρχουν διάφορες ερευνητικές προσπάθειες συνδεδεμένες με τις οντολογίες ασφαλείας όπως και εφαρμογές βάσεων δεδομένων γράφων πάνω τους, παρόλα αυτά οι στόχοι και η σκοπιά τους διαφοροποιούνται. Ανάλογα με το πρόβλημα κάθε οντολογία μπορέι να ασχολείται με εξειδικευμένα αντικείμενα όπως συγκεκριμένες οντότητες του χώρου της ασφαλειας (πχ. Ευπάθειες και αδυναμίες λογισμικού).

Στη βιβλιογραφία, υπάρχουν αρκετές ερευνητικές προσπάθειες που σχετίζονται με τον ορισμό των οντολογιών ασφάλειας, αν και μπορεί να διαφέρουν ως προς τον στόχο και το πεδίο εφαρμογής τους. Ανάλογα με το πρόβλημα που διερευνάται, κάθε οντολογία μπορεί να επικεντρωθεί σε θέματα τόσο συγκεκριμένα όπως οι οντότητες ασφαλείας όπως οι αδυναμίες λογισμικού, οι ευπάθειες λογισμικού, οι κακόβουλοι χρήστες, τα συστήματα ανίχνευσης εισβολών, οι επιθέσεις και τα αντίμετρα ασφαλείας ή τόσο ευρύ όσο οι πολιτικές ασφάλειας, η ασφάλεια δικτύων και η διαχείριση δικτύων, τα συστήματα διαχείρισης ασφάλειας πληροφοριών κ.λπ.

Μία από τις πιο πρόσφατες οντολογίες ασφαλείας που παρουσιάζονται στο [37] δίνει έμφαση σε πληροφορίες που προέρχονται από υπάρχουσες πλατφόρμες συλλογής και διαχείρισης Open Source Cyber Threat Intelligence(OSCTI), οι οποίες εστιάζουν σε δείκτες συμβιβασμού χαμηλού επιπέδου (Indicators of Compromise or IOC). Για να γεφυρωθεί το υπάρχον χάσμα των IOC υψηλότερου επιπέδου, παρουσιάζεται στο [37] μια βάση δεδομένων γράφων που ονομάζεται SecurityKG, η οποία αποτελεί ένα σύστημα για αυτοματοποιημένη συλλογή και διαχείριση OSCTI. Το SecurityKG είναι σε θέση να εξάγει πληροφορίες από αναφορές ημιδομημένου κειμένου μέσω της χρήσης Τεχνητής Νοημοσύνης (AI) και Ανάλυσης Φυσικής Γλώσσας (NLP).

Στο [38] παρουσιάζεται μια οντολογία ασφαλείας που συνδέει γνωστές βάσεις δεδομένων με στοιχεία ασφαλείας όπως το NVD, το CVE, το CWE και το ATT&CK. Παρουσιάζεται ένα γράφημα συγκεντρωτικών δεδομένων που ονομάζεται BRON, το οποίο επιτρέπει την αμφίδρομη, σχεσιακή ανίχνευση διαδρομής εντός οντοτήτων. Στη συνέχεια, το BRON χρησιμοποιείται για τον εντοπισμό προτύπων επίθεσης, τακτικών και τεχνικών που εκμεταλλεύονται τα CVE. Επιπλέον, το BRON είναι σε θέση να υποστηρίξει μια υπόθεση που εκφράζεται σε απλό κείμενο, που περιλαμβάνει πληροφορίες ευρέσιμες μέσω του γραφήματος δεδομένων που περιλαμβάνει.

Μια άλλη πρόσφατη προσέγγιση που παρουσιάζεται στο [39] παρέχει ένα πλαίσιο που επιτρέπει την ενημέρωση της πολιτικής ελέγχου πρόσβασης σε υποδομές Cloud με τη χρήση Cyber Threat Intelligence. Επιπλέον, εξετάζει το ενδεχόμενο ενημέρωσης των πολιτικών ελέγχου πρόσβασης χρησιμοποιώντας συλλογική γνώση συσσωρευμένη απο πρόσφατες δραστηριότητες στον κυβερνοχώρο μιας υποδομής. Για να περιγραφεί η συσχέτιση μεταξύ των πολιτικών ασφαλείας και των αναφορών ασφαλείας, χρησιμοποιείται ένας συνδυασμός της οντολογίας DOLCE-spray [16] και της δομημένης γλώσσας STIX2.

Στο [40] αξιολογείται η σταθερή ανάπτυξη του διαδικτύου των πραγμάτων (Internet of Things IoT) ως αυξανόμενη απειλή για την ασφάλεια, καθώς η ασφάλεια στο IoT δεν είναι ακόμη ένα ώριμο πεδίο. Παρουσιάζουν το DS4IoT, μια οντολογία ασφάλειας δεδομένων που καλύπτει την

αναπαράσταση εννοιών ασφάλειας δεδομένων. Μια άλλη ερευνητική προσπάθεια που παρουσιάζεται στο [41] υπογραμμίζει τη σημασία της κατασκευής γραφημάτων γνώσης ως υποδομή δεδομένων σχετικά με την ασφάλεια στον κυβερνοχώρο. Η προσέγγισή τους συνεπάγεται μια βάση γνώσεων μαζί με ένα σύνολο απο λογικούς κανόνες. Σε αυτό το παράδειγμα παρατηρείται μια ισχυρή σχέση με το NVD, το MITER και το γνωστό μοντέλο Asset-Vulnerability-Threat.

Στο [42] διάφορες οντολογίες ασφάλειας δικτύου προσδιορίζονται και δομούνται σε οκτώ διακριτές κατηγορίες: Απειλές, IDS, Ειδοποιήσεις, Επιθέσεις, Ευπάθειες, Αντίμετρα, Πολιτικές Ασφάλειας και Διαχείριση Δικτύου.

Επίσης το Exploit Prediction Scoring System (EPSS) που παρουσιάζεται στο [11] είναι ένα ανοιχτό πλαίσιο βασισμένο σε δεδομένα που αξιοποιούνται στην αξιολόγηση τρωτών σημείων, στο πλαίσιο του υπολογισμού της πιθανότητας εκμετάλλευσης μιας ευπάθειας ενός συστήματος εντός των πρώτων δώδεκα μηνών μετά τη δημόσια αποκάλυψη της. Αυτό το σύστημα βαθμολόγησης έχει σχεδιαστεί για να είναι απλό και ευέλικτο, ενώ παρέχει ακριβείς εκτιμήσεις σχετικά με την εκμετάλλευση ευπαθειών. Επιπλέον, η υλοποίηση προορίζεται για επεκτασιμότητα, ώστε να μπορεί να ενημερώνεται όσο περισσότερα και καλύτερα δεδομένα γίνονται διαθέσιμα, με βάση αυτή τη λογική επιτρέπει στους χρήστες είτε να αναζητήσουν την πιθανότητα εκμετάλλευσης για καταγεγραμμένα CVE είτε να δημιουργήσουν μια προσαρμοσμένη ευπάθεια ορίζοντας τα αντίστοιχα χαρακτηριστικά χειροκίνητα. Μια λειτουργική έκδοση του EPSS calculator βρίσκεται στον ιστότοπο του οργανισμού kenna research.

Τέλος, στο [43] ορίζεται μια οντολογία ασφαλείας για τη μοντελοποίηση της αξιολόγησης κινδύνου σε επίπεδο επιχειρήσεων. Η οντολογία περιέχει οντότητες για τα βασικά στοιχεία αξιολόγησης κινδύνου, όπως Απειλές, Ευπάθειες, Μηχανισμούς Ασφαλείας και υπάρχοντα συστήματα μαζί με τις διασυνδέσεις τους. Ωστόσο, το μοντέλο είναι μόνο εννοιολογικό και δεν υποστηρίζεται από καμία υλοποίηση ή πηγές δεδομένων είτε δομημένης είτε μη δομημένης μορφής.

Η οντολογία ασφαλείας που παρουσιάζεται σε αυτό το έγγραφο ενσωματώνει και επεκτείνει την παραπάνω σχετική εργασία προτείνοντας μια ολιστική οντολογία ασφάλειας που συνδυάζει πληροφορίες που προέρχονται από: (α) γνωστές ταξινομίες ασφαλείας και βάσεις δεδομένων ευπάθειας, όπως στο [38]. (β) εργαλεία συλλογής πληροφοριών ασφάλειας δικτύου, τα οποία είναι σε θέση να συλλέγουν περιβαλλοντικές πληροφορίες σχετικά με την κατάσταση ασφάλειας δικτύου και λογισμικού και να τα συνδέουν με γνωστές ταξινομίες· και (γ) ημιδομημένο και αδόμητο κείμενο που προέρχεται από αναφορές OSCTI και σχετικές πηγές, όπως στο [37].

3 Τεχνολογίες, Πηγές Δεδομένων, Αρχιτεκτονική Λογισμικού και Οντολογία

3.1 Τεχνολογίες

3.1.1 Λήψη και Ανάγνωση Δεδομένων μέσω python

Requests

Η βιβλιοθήκη Requests αποτελεί το στανταρντ για την πραγματοποίηση αιτημάτων HTTP μέσω Python. Οι έτοιμες συναρτήσεις που προσφέρει η Requests Αφαιρεί την πολυπλοκότητα της υποβολής αιτημάτων με την χρήση ενός απλού API, ώστε οι προγραμματιστές να μπορούν απλά να εστιάζουν στην αλληλεπίδραση με τις διαδικτυακές υπηρεσίες και την κατανάλωση δεδομένων που απαιτούν οι εφαρμογές που αναπτύσσουν.

Beautiful Soup

Το Beautiful Soup είναι μια βιβλιοθήκη της Python που χρησιμοποιείται για την εξαγωγή δεδομένων από αρχεία HTML και XML. Μπορεί να λειτουργήσει σε συνεργασία με έτοιμα ή προσαρμοσμένα parser και παρέχει ιδιαίτερους τρόπους πλοήγησης, αναζήτησης και τροποποίησης δεδομένων σε αρχεία της παραπάνω μορφής. Προγραμματιστές μπορούν να εξοικονομήσουν ώρες ή ημέρες εργασίας χρησιμοποιώντας τις έτοιμες συναρτήσεις της παραπάνω βιβλιοθήκης. [44] [45]

3.1.2 Neo4j

Το Neo4j αποτελεί σύστημα διαχείρισης βάσεων δεδομένων γράφων αποτελώντας μια από τις πλέον αξιόπιστες λύσεις για τις ΒΔΓ που χρησιμοποιούν απευθείας την τεχνολογία γράφων στη βάση. Αυτός ο τύπος των συστημάτων διαχείρισης ΒΔΓ ονομάζεται και Native Graph Database. Η πλατφόρμα ξεκίνησε να αναπτύσσεται το 2007, ενώ το 2010 εξέδωσε την πρώτη της σταθερή έκδοση 1.0. Σημειώνεται πως αυτήν τη στιγμή έχει δημοσιευθεί και χρησιμοποιείται η έκδοση 4.2.7. Ανήκει στην κατηγορία freemium (συνδυασμός free & premium) δηλαδή προσφέρει τόσο δωρεάν -μάλιστα ανοιχτού κώδικα- όσο και επί πληρωμή εκδόσεις και πακέτα λογισμικού ενώ για την δωρεάν παροχή ο χρήστης πρέπει να εγγραφεί στην πλατφόρμα. Διαθέτει ενεργό repository στο GitHub, όπου είναι διαθέσιμες όλες οι δωρεάν παροχές της πλατφόρμας (τόσο το λογισμικό της ΒΔΓ όσο και επιπρόσθετα λογισμικά οπτικοποίησης δεδομένων, εξαγωγής αναλυτικών δεδομένων κοκ.). Το Neo4j έχει υλοποιηθεί μέσω της γλώσσας Java ενώ η αλληλεπίδραση με την ΒΔΓ γίνεται μέσω της γλώσσας Cypher -για την οποία θα μιλήσουμε στη συνέχεια- μέσω http endpoint & bolt protocol (Δυαδικό Πρωτόκολλο που ανέπτυξε το Neo4j για την επικοινωνία εφαρμογών – πελατών με τους διακομιστές της ΒΔΓ.) [46] [47].

Τεχνολογία-Καινοτομία

Στην προηγούμενη ενότητα μελετήσαμε την λογική και τεχνολογία πίσω από τις ΒΔΓ, καθώς και βασικά χαρακτηριστικά τους. Αξίζει, λοιπόν, να εμβαθύνουμε στο πως το Neo4j υλοποιεί την τεχνολογία των ΒΔΓ, και ποια εκείνα τα στοιχεία το κάνουν, ένα από τα βασικά εργαλεία χρήσης των ΒΔΓ και εξαγωγής αναλυτικών δεδομένων χάρη σε αυτές.

Μια σημαντική διαδικασία στην δομή των γράφων, είναι η αναζήτηση από κόμβο σε κόμβο (node traversal). Η διαδικασία αυτή ουσιαστικά δοθέντος ενός σημείου εκκίνησης προσπελαύνει τους κόμβους έναν προς έναν αξιοποιώντας τις σχέσεις μεταξύ τους, ώστε να φτάσει σε ένα σημείο τερματισμού ή να εξάγει το οποιοδήποτε αποτέλεσμα. Σε μαθηματικό επίπεδο υπάρχουν δεκάδες αλγόριθμοι που χρησιμοποιούν και βελτιώνουν αυτήν τη διαδικασία. Στις ΒΔΓ η διαδικασία αυτή της προσπέλασης των κόμβων είναι ο πλέον αποδοτικός τρόπος «περιπλάνησης» εντός των εγγραφών, μιας και δεν χρειάζεται όπως είπαμε και στην προηγούμενη ενότητα η επιμέρους ομαδοποίηση εγγραφών, οι εμφωλευμένες επαναλήψεις στην αναζήτηση σε τέτοιες ομάδες κοκ. Το Neo4j διαθέτει μια διεπαφή API εμπλουτισμένη με πλήθος τρόπων για την προσπέλαση των κόμβων, ώστε ο χρήστης να μπορεί να περιηγηθεί στο σχήμα

της ΒΔΓ. Αυτό σε συνδυασμό, με τον απλό τρόπο αναζήτησης δεδομένων μέσω της γλώσσας Cypher και του REST API, προσφέρει στον χρήστη ένα εύρος επιλογών και δυνατοτήτων, για την εξαγωγή των επιθυμητών αποτελεσμάτων και συμπερασμάτων από τους γράφους της βάσης.

Η αξιοποίηση μάλιστα αυτού του τρόπου προσπέλασης των γράφων έναντι των αναζητήσεων στα δεδομένα σε μια σχεσιακή ΒΔ ή σε ένα κατανεμημένο σύστημα διαχείρισης ΒΔ, έχει σαν κύριο χαρακτηριστικό το ότι ψάχνει χρησιμοποιώντας τον δοθέντα αρχικό κόμβο, σύμφωνα με το τι θα υποδείξει ο χρήστης. Με αυτόν τον τρόπο αποφεύγει να προσπελαύνει όλες τις εγγραφές, άρα εκ των πραγμάτων είναι ο πιο γρήγορος τρόπος αναζήτησης (σύμφωνα πάντα με την περίπτωση χρήσης, τα δεδομένα, το ερώτημα στη ΒΔ κοκ.). Ως native ΒΔΓ το Neo4j κερδίζει χάρη στην δομή αποθήκευσης των δεδομένων και τον τρόπο που τα επεξεργάζεται, με σημαντικό σημείο αυτής της μεθοδολογίας να είναι η αποδοτική προσπέλαση των γράφων.

Το Neo4j ενώ είναι μη σχεσιακό σύστημα διαχείρισης ΒΔ, και θα περίμενε κανείς, πως όπως και τα υπόλοιπα συστήματα διαχείρισης ΒΔ της κατηγορίας (βλ. MongoDB, Apache Spark, Google BigTable κλπ.) δεν θα είναι διαμορφωμένο για την εκτέλεση δοσοληψιών με τον βέλτιστο τρόπο, όπως θα μπορούσε να είναι ένα σχεσιακό σύστημα διαχείρισης ΒΔ, καταφέρνει να είναι πλήρως συμμορφωμένο με τις ιδιότητες ACID (Atomicity – Ατομικότητα, Consistency – Συνέπεια, Isolation – Απομόνωση, Durability – Μονιμότητα) και τα συστήματα OLTP. Ιδιαιτέρως σε εργασιακό περιβάλλον, η πλατφόρμα πρέπει να υποστηρίζει πλήρως τις ACID ιδιότητες. Γι' αυτό ακριβώς το Neo4j συνδυάζοντας την native ΒΔΓ, την απλή και πλούσια εξαγωγή αναλυτικών δεδομένων, με την πλήρη ανταπόκριση στις ανάγκες των συνεχών δοσοληψιών σε μια ΒΔ, κερδίζει συνεχώς έδαφος στις επιχειρήσεις έναντι άλλων ΒΔΓ και ευρύτερα NoSQL λύσεων.

Σε αυτό το σημείο θα εμβαθύνουμε στον τρόπο με τον οποίο πραγματοποιούνται οι δοσοληψίες στο Neo4j [48]. Μια δοσοληψία, δηλαδή μια σειρά γεγονότων που εφαρμόζονται στην ΒΔ, μπορεί ολοκληρωτικά να πετύχει ή να αποτύχει. Δηλαδή δεν μπορεί εντός μιας δοσοληψίας να επιτύχει ένα ερώτημα και να αποτύχει μια σύνδεση, ή να επιτύχει ένα ερώτημα και να αποτύχει το επόμενο. Τότε η δοσοληψία θα αποτύχει και θα γίνει το λεγόμενο roll back. Οι δοσοληψίες έχουν ορισμένο κύκλο ζωής: 1) Εκκινούν 2) Εκτελούν καμία ή παραπάνω λειτουργίες 3) Ολοκληρώνονται επιτυχώς ή αποτυγχάνουν (κάνουν rollback). Η σημασία της ορθής λειτουργίας των δοσοληψιών είναι τεράστια ιδίως όταν στην ΒΔ δουλεύουν άνω του ενός χρήστες. Αυτό συμβαίνει, διότι, κάθε χρήστης εν δυνάμει μπορεί να επηρεάσει δεδομένα της ΒΔ τα οποία ταυτόχρονα επηρεάζει και ο άλλος χρήστης (σε άλλη προφανώς ταυτόχρονη σύνδεση στη ΒΔ). Σε μια ρεαλιστική (μεγάλη) κλίμακα, τέτοιες εργασίες γίνονται από εκατοντάδες ή χιλιάδες χρήστες ταυτόχρονα. Ενώ στην Εικόνα 4 βλέπουμε έναν τρόπο εύρεσης των συνδέσεων των χρηστών και συσχέτισής τους με την εκάστοτε δοσοληψία, αυτό σε πραγματικό περιβάλλον είναι μη εφικτό. Λόγω του ότι οι συνδέσεις και οι δοσοληψίες αλλάζουν σε δευτερόλεπτα, σε κάθε εκτέλεση των δύο εντολών που παρουσιάζονται στην Εικόνα 4, θα παίρναμε τελείως διαφορετικά αποτελέσματα. Στο πιθανό αυτό ζήτημα το Neo4j δίνει την λύση της εφαρμογής ανοιχτού κώδικα «Prometheus» όπου ενεργοποιεί στην ουσία συνεχή παρακολούθηση του Neo4j Server που λειτουργεί η ΒΔΓ και μεταξύ άλλων έχουμε όλα τα απαραίτητα δεδομένα και αρχεία καταγραφής για τις δοσοληψίες, τις συνδέσεις χρηστών κοκ. Συνολικά το Neo4j υποστηρίζει πλήρως τις ACID ιδιότητες:

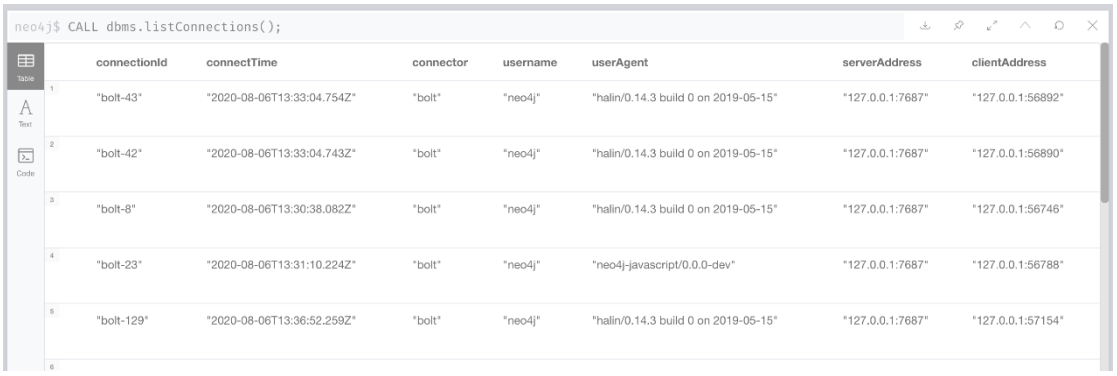
- Atomicity – Ατομικότητα: Σε περίπτωση που οποιοδήποτε κομμάτι της δοσοληψίας αποτύχει, αποτυγχάνει ολόκληρη η δοσοληψία και δεν επηρεάζει την κατάσταση της ΒΔ.
- Consistency – Συνέπεια: Καμία δοσοληψία δεν επηρεάζει την συνέπεια της ΒΔ, συνεπώς οποιαδήποτε δοσοληψία προσπαθεί να την αλλάξει, αποτυγχάνει. Για παράδειγμα δεν μπορεί ένας χρήστης σε ένα πεδίο που δέχεται ακέραιους αριθμούς, να γράψει αλφαριθμητικό χαρακτήρα σε ένα ερώτημα μιας δοσοληψίας. Αυτή θα αποτύχει, αλλά η ΒΔ δεν θα επηρεαστεί. Έπειτα από την επιτυχή δοσοληψία κάθε προσπέλαση στα δεδομένα που επεξεργάστηκαν θα λαμβάνει ως αποτέλεσμα τα ενημερωμένα πια, δεδομένα.
- Isolation – Απομόνωση: Όπως είπαμε και παραπάνω για τις παράλληλες διεργασίες στην βάση, καμιά δοσοληψία δεν μπορεί να επεξεργαστεί δεδομένα της ΒΔ, τα οποία επεξεργάζονται από άλλη την ίδια στιγμή. Επίσης, τα ίδια δεδομένα μπορούμε

ταυτόχρονα να τα επεξεργαστούμε σε μια δοσοληψία και σε μια άλλη να τα διαβάσουμε. Με την χρήση του Neo4j Java API οι κόμβοι και οι σχέσεις μπορούν να κλειδωθούν. Δηλαδή έστω ότι εντός μια δοσοληψίας ένας κόμβος θα υποστεί ενημέρωση. Ο κόμβος κλειδώνεται και έτσι καμιά άλλη δοσοληψία ταυτόχρονα δεν θα τον επεξεργαστεί, μέχρι η πρώτη να ολοκληρωθεί επιτυχώς ή να κάνει rollback (σε κάθε περίπτωση να τερματίσει). Οι κόμβοι και οι σχέσεις υφίστανται κλείδωμα εγγραφής σε πληθώρα επεξεργασιών όπως αλλαγές σε ιδιότητες τους, εγγραφές και διαγραφές σε κόμβους και σχέσεις μεταξύ τους κοκ.

- Durability – Μονιμότητα: Σε περίπτωση επιτυχούς δοσοληψίας, τα δεδομένα γράφονται στον δίσκο, δεν χάνονται, δεν καταστρέφονται κάτω από οποιεσδήποτε συνθήκες π.χ. μια ξαφνική δυσλειτουργία της ΒΔ ή επανεκκίνηση της ΒΔ.

Εμβαθύνοντας, αξίζει να δούμε πως υλοποιείται μια δοσοληψία στο Neo4j [49]. Κάθε δοσοληψία αναπαρίσταται ως αντικείμενο στη μνήμη του οποίου η κατάσταση δείχνει στις αντίστοιχες εγγραφές της ΒΔ. Το αντικείμενο ελέγχεται από τον διαχειριστή κλειδώματος όπως αναφέραμε παραπάνω, ο οποίος κλειδώνει τους κόμβους και τις σχέσεις, για εγγραφή, ενημέρωση ή διαγραφή. Σε περίπτωση που έχουμε rollback της δοσοληψίας, το αντικείμενο απορρίπτεται – διαγράφεται από τη μνήμη, οι σχετικοί κόμβοι και σχέσεις ξεκλειδώνονται για εγγραφή, ενημέρωση και διαγραφή, ενώ σε περίπτωση επιτυχούς δοσοληψίας γίνεται η εγγραφή στο δίσκο. Κατά τις εγγραφές στον δίσκο το Neo4j χρησιμοποιεί το Write Ahead Log, όπου οι επικείμενες αλλαγές εγγράφονται στο active transaction log -όπως αναφέραμε τόσο με την εφαρμογή Prometheus όσο και με την σωστή εκτέλεση της call dbms.listTransactions() έχουμε πλήρη πρόσβαση στο active transaction log-. Στη συνέχεια θα γίνει η εγγραφή στο δίσκο, όπου πλέον η οποιαδήποτε επεξεργασία θα είναι μόνιμη. Όταν η διαδικασία εγγραφής τελειώσει, τότε οι αλλαγές θα εφαρμοστούν στον γράφο και εφόσον και αυτό το βήμα ολοκληρωθεί τότε όλα τα σχετικά με τη δοσοληψία κλειδώματα θα φύγουν η δοσοληψία θα ολοκληρωθεί.

Με την εντολή call dbms.listConnections() (Εικόνα 21) το Neo4j επιστρέφει όλες τις ενεργές συνδέσεις χρηστών με το ID της σύνδεσης, την ώρα σύνδεσης, τον τρόπο σύνδεσης, το όνομα χρήστη κλπ. Στην περίπτωση που δοκιμάζαμε την εντολή call dbms.listTransactions() το Neo4j θα επέστρεφε σε αρκετές -αν όχι όλες- περιπτώσεις την δοσοληψία που είναι συνδεδεμένη με το ID της σύνδεσης (Connection_ID). Δηλαδή εύκολα γνωρίζουμε ποιος χρήστης, προχωράει στην κάθε δοσοληψία. Ενδεικτικά οι βασικές μέθοδοι για συνδέσεις, δοσοληψίες και ερωτήματα στο Neo4j είναι: dbms.listTransactions(), dbms.killTransaction(), dbms.listConnections(), dbms.killConnection(), dbms.listQueries(), dbms.killQueries(), dbms.killQuery().



Εικόνα 21: Εντολή call dbms.listConnections()

Αναφερθήκαμε παραπάνω σε κάποιες από τις βασικές λειτουργίες του Neo4j, με έμφαση σε συγκεκριμένες τεχνικές λεπτομέρειες, που μας δίνουν μια σαφή εικόνα της τεχνολογίας πίσω από την πλατφόρμα αυτή. Ωστόσο, αξίζει να αναρωτηθούμε γιατί το Neo4j θεωρείται μια από τις κορυφαίες πλατφόρμες στα συστήματα διαχείρισης ΒΔΓ; Ποιες είναι οι άλλες επιλογές που έχει ένας χρήστης των ΒΔΓ και ως προς τι τα συγκρίνουμε;

Σε αυτό το ερώτημα προσπάθησαν να απαντήσουν οι επιστήμονες Diogo Fernandes και Jorge Bernandino στο άρθρο τους με τίτλο «Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4j and OrientDB» [50]. Τα κυριότερα χαρακτηριστικά που πρέπει να έχει μια ΒΔΓ βάσει των αναλύσεων των Buerli (2012) και Cox (2017) είναι τα εξής:

- Ευέλικτο Σχήμα: Την στιγμή που οι σχεσιακές ΒΔ χρειάζονται νέους πίνακες, μεγάλες εναλλαγές σε υπάρχοντες, νέους τύπους δεδομένων κοκ. οι ΒΔΓ με τη χρήση κόμβων και ακμών μπορούν εύκολα να αναπαραστήσουν νέα δεδομένα.
- Γλώσσα Ερωτημάτων: Στις ΒΔΓ υπάρχει πληθώρα γλωσσών, ανάλογα την πλατφόρμα. Το μέτρο σύγκρισης έγκειται στο πως οι γλώσσες χρησιμοποιούνται για την εξαγωγή αποτελεσμάτων, πόσο απλές και αποτελεσματικές είναι.
- Κατανομή Δεδομένων: Χαρακτηριστικό των κατανεμημένων συστημάτων διαχείρισης ΒΔ, όταν ένα μεγάλο δείγμα δεδομένων θα πρέπει να τμηθεί σε κομμάτια για την βέλτιστη επεξεργασία και αποθήκευσή του.
- Αντίγραφα Ασφαλείας: Οι ΒΔΓ θα πρέπει να προσφέρουν λειτουργίες για προγραμματισμό, πραγμάτωση και επαναφορά αντιγράφου ασφαλείας της βάσης. Ένα πλήρες αντίγραφο πρέπει να περιέχει όλα τα αρχεία δεδομένων και όλες τις απαραίτητες πληροφορίες για την πλήρη λειτουργία της ΒΔ, την χρονική στιγμή που πάρθηκε το αντίγραφο [51].
- Multi-Model: Να μπορεί να υποστηρίξει περισσότερους τύπους οργάνωσης δεδομένων και φυσικά να οπτικοποιήσει τα δεδομένα και τις σχέσεις τους υπό τη μορφή γράφων, πινάκων, εγγράφων (βλ. json exporting).
- Multi-Architecture: Κατά την ανάπτυξη ενός συστήματος διαχείρισης ΒΔ, οι αρχιτεκτονικές ανάπτυξης που χρησιμοποιούνται ποικίλουν ανάλογα με το επιθυμητό αποτέλεσμα.
- Επεκτασιμότητα: Πως οι ΒΔΓ χρησιμοποιούν το υλισμικό και κατά πόσο είναι έτοιμες να λειτουργήσουν ομαλά σε πιθανές αλλαγές του ή σε αύξηση του όγκου δεδομένων προς επεξεργασία.
- Υποστήριξη Cloud Υπηρεσιών: Μια ΒΔΓ ως υπηρεσία στο cloud είναι σημαντικό πεδίο διαφοροποίησης μεταξύ των ΒΔΓ καθώς λύνει σημαντικά τα θέματα της επεκτασιμότητας αλλά και προσφέρει πληθώρα δυνατοτήτων στον τελικό χρήστη.

Κλίμακα από 0 έως 4 *0 → Καθόλου 4 → Πλήρως*	AllegroGraph	ArangoDB	InfiniteGraph	Neo4j	OrientDB
Ευέλικτο Σχήμα	1	3	3	4	3
Γλώσσα Ερωτημάτων	3	3	3	4	3
Κατανομή Δεδομένων	3	3	0	0	3
Αντίγραφα Ασφαλείας	3	2	3	4	3
Multi-Model	4	4	2	2	4
Multi-Architecture	3	4	3	4	3
Επεκτασιμότητα	3	4	3	4	3
Υποστήριξη Cloud Υπηρεσιών	3	3	4	4	3
Σύνολο	23	26	21	26	25

Τα αποτελέσματα δείχνουν πως σχεδόν όλες οι ΒΔΓ κινούνται στα ίδια περίπου επίπεδα με μικρές επιμέρους διαφορές κυρίως στην κατανομή των δεδομένων όπου έχουμε αυτές που ακολουθούν το κλασσικό NoSQL μοντέλο, στην υποστήριξη cloud υπηρεσιών και το πόσο ευέλικτο σε αλλαγές είναι το σχήμα που μπορεί να υποστηρίξει η ΒΔΓ. Συνοψίζοντας, το Neo4j έχει ένα απολύτως ευέλικτο σχήμα, είναι πλήρως επεκτάσιμο σε αλλαγές στο υλισμικό ή τον όγκο των δεδομένων, έχει απλή και εύχρηστη γλώσσα, την Cypher για την οποία θα μιλήσουμε παρακάτω, υποστηρίζει μέσω ειδικών Drivers γλώσσες προγραμματισμού όπως η Java, Spring, Scala, Javascript, Python, υποστηρίζει HTTP API για διαχείριση της ΒΔΓ, ευρετήριο με τη χρήση του λογισμικού Apache Lucence και της σχετικής βιβλιοθήκης, κρατάει online αντίγραφα ασφάλειας, πλήρως ελεύθερα προς επεξεργασία, προσφέρει λύσεις ΒΔΓ ως cloud υπηρεσία, έχει απλό και εύχρηστο περιβάλλον χρήσης με εξαιρετική οπτικοποίηση των δεδομένων, εξάγει δεδομένα σε πλήθος μορφών όπως .json, .xls, .png images κοκ., έχει ισχυρή κοινότητα υποστήριξης (λόγω του ότι βασίζεται και σε λογισμικό ανοικτού κώδικα) και φυσικά ως native graph database είναι η πλατφόρμα με την υψηλότερη απόδοση. Ωστόσο δεν υποστηρίζει την κατανομή των δεδομένων (sharding) κάτι που οι περισσότερες NoSQL λύσεις ακόμη και ΒΔΓ το προσφέρουν.

Κλείνοντας την ενότητα αυτή, θα αναφερθούμε σε μια από τις εμπορικές δυνατότητες του Neo4j, την πλατφόρμα Aura που αποτελεί Βάση Δεδομένων Γράφων ως υπηρεσία στο cloud. Είναι υπηρεσία με ειδικά πακέτα συνδρομής που ποικίλουν, και αποτελεί την μεγαλύτερη cloud ΒΔΓ παγκοσμίως. Ουσιαστικά φέρνει όλες τις καινοτομίες του Neo4j που προαναφέραμε, σε ένα εμπορικό προϊόν, που θα λειτουργεί εκτός εταιρικού περιβάλλοντος, προσφέροντας μια εξαιρετική απόδοση ακόμη και σε πολύ μεγάλο φορτίο. Σημαντικές προς αναφορά είναι και οι λειτουργίες ασφάλειας που προσφέρει με κρυπτογράφηση των δεδομένων, σύστημα ελέγχου πρόσβασης για τους χρήστες και απομόνωση vpc.

Γλώσσα Cypher

Η ανάπτυξη του προγράμματος της διπλωματικής εργασίας γίνεται με τη χρήση της γλώσσας Python και της γλώσσας Cypher που αποτελεί γλώσσα ερωτημάτων του Neo4j [52]. Η Cypher επιτρέπει στους χρήστες την αποθήκευση, επεξεργασία και προσπέλαση των δεδομένων από τη ΒΔΓ. Είναι μια γλώσσα ερωτημάτων, εύκολη στην εκμάθηση, στην κατανόηση και χρήση, ταυτόχρονα όμως με μεγάλες δυνατότητες και λειτουργίες. Η σύνταξη της γλώσσας βασίζεται σε έναν τρόπο οπτικού και λογικού μονοπατιού για την εργασία πάνω στους κόμβους και τις σχέσεις αυτών σε οποιοδήποτε γράφο. Είναι ουσιαστικά μια γλώσσα εμπνευσμένη από την SQL σε συνδυασμό με ASCII-Art στοιχεία. Ταυτόχρονα, επιτρέπει τη δημιουργία ερωτημάτων για σύνθετες λειτουργίες πάνω στη βάση, με τέτοιο τρόπο ώστε η εκτέλεσή τους να είναι η βέλτιστη σε επίπεδο απόδοσης και υλοποιεί όλες τις συνήθεις λειτουργίες των γλωσσών προγραμματισμού αλλά και ερωτημάτων. Η Cypher δεν είναι απλά η γλώσσα του Neo4j. Είναι κομμάτι ενός ευρύτερου εγχειρήματος ανοιχτού κώδικα του Open Cypher όπου η κοινότητα έχει αναπτύξει διάφορες μικρές παραλλαγές της γλώσσας όπου χρησιμοποιούνται σε πλήθος βάσεων δεδομένων γράφων αλλά και λογισμικών που εισάγουν την τεχνολογία γράφων σε άλλα υπάρχοντα με τη μορφή Addons.

Στην πραγματικότητα η Cypher δεν διαφέρει πολύ από την SQL. Είναι γλώσσα ερωτημάτων, απλά προσαρμοσμένη στην τεχνολογία γράφων. Στόχο έχει δηλαδή, να μπορεί να εκφράσει τις γραφικές αναπαραστάσεις που θα δείξουν στη βάση τον τρόπο με τον οποίο θα δουλέψει πάνω στα δεδομένα και στο σχήμα. Με την χρήση των ASCII-Art συμβόλων, η Cypher έγινε περισσότερο φιλική προς τον χρήστη, ώστε ο τελευταίος να μπορεί να εκφράσει σε προγραμματισμό εύκολα και αποδοτικά μια λειτουργία πάνω σε έναν γράφο. Φυσικά μέσω της γλώσσας υποστηρίζονται όλοι οι αλγόριθμοι που επιδρούν στους γράφους, όπως οι αλγόριθμοι εύρεσης συντομότερων μονοπατιών μεταξύ κόμβων κοκ.

Παραδείγματα

Στην επόμενη υπό-ενότητα θα αναλύσουμε την γλώσσα Cypher μέσω παραδειγμάτων, ώστε στην συνέχεια να γίνει απολύτως κατανοητή η παρουσίαση της χρήσης της εντός του προγράμματος που αναπτύξαμε.

Δημιουργία Κόμβων και Σχέσεων:

- Δημιουργία απλού κόμβου → create (n)
- Δημιουργία κόμβων εντός επανάληψης → foreach (i in range(0,49) | create (n))
- Δημιουργία κόμβου με ετικέτα → create (n: Label)
- Δημιουργία κόμβου με ετικέτα και ιδιότητες → create (n: Label {property1:value1, property2:value2})
- Δημιουργία διαφορετικών κόμβων με ιδιότητες → create (n: Label {property1:value1}), (p:Label {property1:value1, property2:value2})
- Δημιουργία σχέσης μεταξύ των κόμβων n και p → create (n)-[r:Rel_Name]->(p)
- Δημιουργία της παραπάνω σχέσης με ιδιότητες → create (n)-[r:Rel_Name {property1:value1, property2:value2}]->(p)

Με την χρήση του Python Driver του Neo4j μπορούμε εντός του προγράμματος σε python να εντάξουμε και queries σε Cypher. Μάλιστα μπορούμε να περάσουμε δυναμικά δεδομένα από το πρόγραμμα στο query. Έστω q το query, x το python dictionary. Με ενεργό session βάσει του python driver μπορούμε να περάσουμε μεταβλητές από το dictionary στο query.

```
q = “create (n: Person {Name: $Name})”
x = {“Name”: Variable_Name}
session.run(q,x)
```

Δημιουργία κλειδιού σε συγκεκριμένη ιδιότητα ενός κόμβου, ώστε να μην είναι δυνατές οι διπλές εγγραφές σε αυτήν την ιδιότητα → create constraint Constraint_Name if not exists on (p:Person) assert p.AMKA is unique

Δημιουργία ευρετηρίου σε ιδιότητα ενός κόμβου, ώστε η προσπέλαση με βάση αυτήν την ιδιότητα να είναι πιο γρήγορη → create index Index_Name if not exists for (p:Person) on (p.Age)

Διαγραφή κλειδιού → drop constraint Constraint_Name if exists

Διαγραφή ευρετηρίου → drop index Index_Name if exists

Διαγραφή Κόμβων και Σχέσεων:

- Διαγραφή όλων των κόμβων και των σχέσεων του σχήματος → match (n) detach delete (n)
- Διαγραφή όλων των κόμβων του σχήματος (προϋποθέτει να μην είναι συνδεδεμένοι με άλλους δηλαδή να μην έχουν σχέσεις, δηλαδή να έχει προηγηθεί η εντολή detach για τους κόμβους που θα διαγράψουμε) → match (n) delete (n)
- Διαγραφή κόμβων και των σχέσεων τους με συγκεκριμένη ετικέτα → match (n: Label) detach delete (n)
- Διαγραφή κόμβων και των σχέσεων τους με ετικέτα και χαρακτηριστικά → match (n: Label) where n.Name = Value detach delete (n)
- Διαγραφή κόμβων που έχουν συγκεκριμένη σχέση → match (n:Person)-[:Friend]->(p:Person) detach delete (n), (p)

Όπως θα παρατηρήσατε, για να διαγράψουμε έναν κόμβο ή μια σχέση, θα πρέπει πρώτα να θέσουμε τι ακριβώς θέλουμε να διαγράψουμε, δηλαδή να ψάξουμε για το συγκεκριμένο στοιχείο. Όπως αναφέραμε σε προηγούμενες ενότητες, στις ΒΔΓ όταν θέλουμε να προσπελάσουμε στοιχεία του γράφου πρέπει να θέσουμε σημείο εκκίνησης (συνήθως ομάδα σημείων). Το πόσο ακριβής ή γενική θα είναι αυτή η ομάδα των σημείων εκκίνησης εξαρτάται από το ερώτημα που θέλουμε να εκτελέσουμε και φυσικά από την εμπειρία του χρήστη που γράφει τον κώδικα.

Προσπέλαση Κόμβων:

- Εκτύπωση όλων των κόμβων του σχήματος → match (n) return (n)

- Εκτύπωση όλων των κόμβων της συγκεκριμένης κατηγορίας → match (n:Person) return (n)
- Εκτύπωση όλων των κόμβων της συγκεκριμένης κατηγορίας που έχουν ένα κοινό χαρακτηριστικό → match (n:Person) where n.Name = Value return (n)
- Διαφορετικός τρόπος εκτέλεσης του παραπάνω ερωτήματος (ελαφρώς διαφοροποιημένη απόδοση, διαφορές παρατηρηθούν σε μεγάλες ομάδες κόμβων, μεγάλα datasets κοκ.) → match (n: Person {Name: Value})
- Εκτύπωση όλων των κόμβων που έχουν συγκεκριμένη σχέση → match (n)-[o:Rel_Name]->(p) return n,o,p
- Εκτύπωση συγκεκριμένων ιδιοτήτων των κόμβων → match (n:Person) return n.Name, n.Age
- Στην περίπτωση που ψάχνουμε κόμβο μη γνωρίζοντας την κατεύθυνση των σχέσεων του, αλλά γνωρίζοντας ότι έχει σχέση → match (:Person {Name:Value})—(:Person {Name:Value})
- Υπάρχουν περιπτώσεις που επιθυμούμε την ακριβή εκτύπωση του αποτελέσματος ενός σχεδίου αναζήτησης που θέτουμε στο ερώτημα. Δηλαδή θέλουμε στην περίπτωση που δεν υπάρξει αποτέλεσμα να πάρουμε επιστροφή null. Τότε χρησιμοποιούμε το optional match → optional match (n: Person {Name: Value1, Age: Value2}) return (n)
- Σύνθετη αναζήτηση κόμβων με διπλή σχέση. Σε ένα σχήμα με ταινίες, ηθοποιούς και σκηνοθέτες, θέλουμε να βρούμε τις ταινίες που σκηνοθετήθηκαν από συγκεκριμένο σκηνοθέτη καθώς και τα ονόματα των ηθοποιών που έπαιξαν σε αυτές → optional match (p:Actor)-[:acted_in]->(n)<-[:directed]-(d:Director {Name:Value}) return p.Name, n.Title

Στην Cypher μπορούμε να προσπελάσουμε τους κόμβους του γράφου, με ποικίλους τρόπους και σημεία εκκίνησης. Ακόμη και να μην γνωρίζουμε τις σχέσεις μεταξύ των κόμβων μπορούμε να ψάξουμε βάση της ύπαρξης ή μη σχέσης ((a)—(b)), την κατεύθυνση της ((a)→(b), (a)→(b)←(c) κλπ.), με ετικέτα στον κόμβο ή χωρίς ((a:Person)—(b)), θέτοντας την σχέση χωρίς να γνωρίζουμε τις δύο πλευρές της ((a)-[:Friend_Of]-(b)) ακόμη και το βάθος της αναζήτησης που θέλουμε να κάνουμε, κάτι ιδιαίτερα χρήσιμο στην υλοποίηση αλγορίθμων εύρεσης συντομότερων μονοπατιών: (a)-[*..5]->(b) δηλαδή ο κόμβος a να έχει το πολύ 5 σχέσεις που κατευθύνονται σε άλλους κόμβους.

Χρησιμοποιούνται κανονικά όλοι οι μαθηματικοί τελεστές, οι λογικοί τελεστές, οι λειτουργίες σε αλφαριθμητικά και οι regex τελεστές, οι κατηγοριοποιήσεις βάσει ιδιοτήτων κατά την εκτύπωση, οι ταξινομήσεις, οι περιορισμοί στην ποσότητα των κόμβων που θα εκτυπωθούν κλπ. Υπάρχει επίσης η λειτουργία case όπου θέτουμε συνθήκη και ανάλογα τους ελέγχους και τα αποτελέσματα προχωράμε σε συγκεκριμένες εντολές.

Τέλος θέλουμε να δείξουμε τον τρόπο με τον οποίο στη Cypher ελέγχουμε για την ύπαρξη ενός κόμβου ή μιας σχέσης ή μιας πιο σύνθετης δομής (κόμβος με σχέση σε άλλο συγκεκριμένο κόμβο κλπ.) και στην περίπτωση που δεν υπάρχει, τον φτιάχνουμε. Αυτό γίνεται με τη χρήση της εντολής merge, όπου αναζητεί το μονοπάτι ή τον κόμβο που θα ορίσουμε και στην περίπτωση που δεν υπάρχει, το δημιουργεί. Η εντολή αυτή συνδυάζεται με την ύπαρξη constraints & indexes διότι σε μεγάλους όγκους δεδομένων, η διαδικασία αναζήτησης θα γίνει πολύ γρήγορα και φυσικά η διαδικασία εγγραφής θα γίνει αποδοτικά και σωστά με την χρήση των constraints. Δεδομένου επίσης, ότι η merge ψάχνει σύμφωνα με το συγκεκριμένο όρισμα, όταν θέλουμε να δημιουργήσουμε έναν κόμβο ή μια σχέση με παραπάνω ιδιότητες, ψάχνουμε βάσει του κλειδιού, και σε επόμενο σημείο ορίζουμε τις ιδιότητες (βλ. παρακάτω)

Η χρήση της Merge:

- Αναζήτηση κόμβου και δημιουργία νέου → merge (p:Person)
- Ομοίως με ιδιότητα όπου ο νέος κόμβος δεν θα φτιαχτεί αν υπάρχει ήδη ένας με p.Name=Value → merge (p:Person {Name:Value})

- Στην περίπτωση που ο κόμβος δεν υπάρχει και δημιουργηθεί, πρέπει να ορισθούν συγκεκριμένες ιδιότητες → merge (p:Person {Name:Value}) on create set p.Age=Value2, p.SSID=Value3
- Στην περίπτωση που ο κόμβος υπάρχει (ή δεν υπάρχει), πρέπει να ορισθούν συγκεκριμένες ιδιότητες → merge (p:Person {Name:Value}) set p.Age=Value2, p.SSID=Value3
- Στην περίπτωση που ο κόμβος υπάρχει, πρέπει να ορισθούν συγκεκριμένες ιδιότητες → merge (p:Person {Name:Value}) on match set p.Age=Value2, p.SSID=Value3
- Χρήση της Merge σε σχέσεις → merge (:Person {Name:Value}]-[:Friend_Of]->(:Person {Name:Value2})
- Και σε μη κατευθυνόμενη σχέση → merge (:Person {Name:Value}]-[:Friend_Of]-(:Person {Name:Value2})

Τα παραπάνω παραδείγματα αποτελούν βασικά για τη χρήση της Cypher και του Neo4j, την οποία μάλιστα θα δούμε αναλυτικά στην περιγραφή του προγράμματος που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας.

Neo4j Browser

Το Neo4j Browser ή αλλιώς ο τρόπος με τον οποίο θα περιηγηθούμε στις δυνατότητες της πλατφόρμας στις διάφορες ΒΔ, αποτελεί ένα εργαλείο απευθυνόμενο κυρίως σε χρήστες με προγραμματιστικές γνώσεις, που επιτρέπει την εκτέλεση Cypher ερωτημάτων και οπτικοποιεί τα αποτελέσματα υπό τη μορφή γράφων και πινάκων. Είναι ουσιαστικά η προεπιλεγμένη διεπαφή του χρήστη με τις βάσεις και τα δεδομένα εντός τους, τόσο για την εμπορική όσο και για την ελεύθερη έκδοση.

Ο Cypher Editor προσφέρει εργαλεία βοήθειας στον χρήστη για να γράψει και να εκτελέσει απλά και σωστά το ερώτημα. Προσφέρει αυτόματη συμπλήρωση του κώδικα, ειδοποιήσεις για λάθη στη σύνταξη και φυσικά ειδικό χρωματισμό που βοηθάει οπτικά στην κατανόηση του κώδικα. Κάθε ερώτημα επιστρέφει πέραν των αποτελεσμάτων και τον χρόνο εκτέλεσης και σε περίπτωση επεξεργασίας δεδομένων, τον αριθμό των δεδομένων που υπέστησαν αλλαγές.

Το Neo4j Browser επιτρέπει επίσης, απλές αλλαγές σε χρώματα και μεγέθη των κόμβων και σχέσεων του γράφου. Τέλος, ο χρήστης έχει τη δυνατότητα να εκτελέσει REST requests [53]. Για παράδειγμα για την εκτέλεση ενός ερωτήματος cypher, αρκεί ένα post στο transaction cypher endpoint:

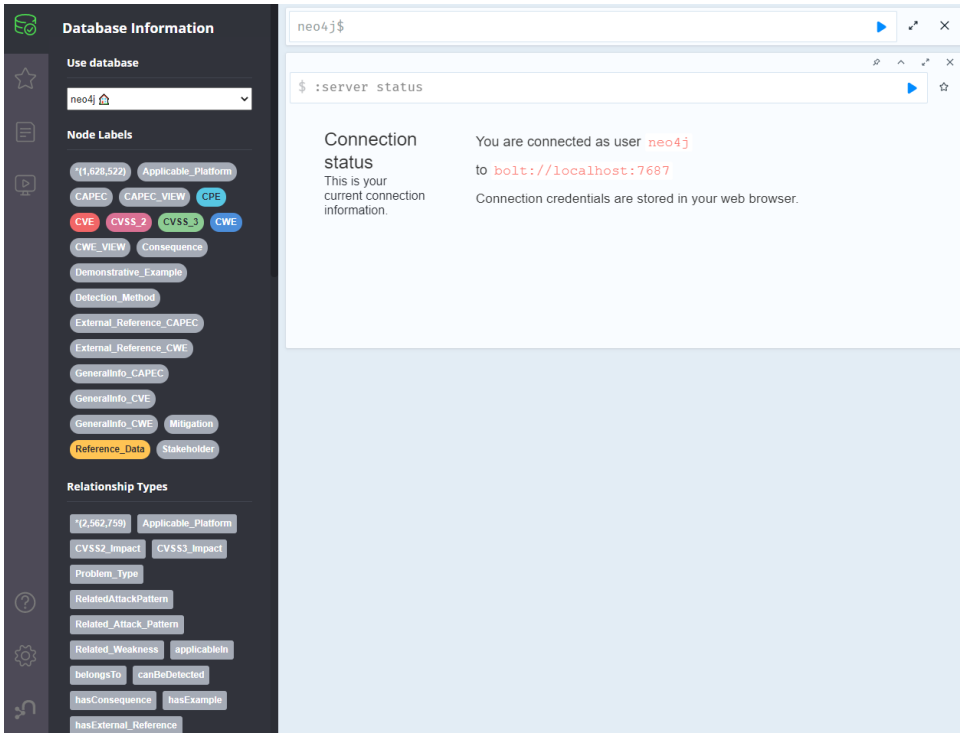
:POST /db/data/transaction/commit {"statements":[{"statement":"MATCH (m:Movie) WHERE m.title={title} RETURN m.title, m.released, labels(m)", "parameters":{"title":"Cloud Atlas"}}]}

Φυσικά οι δυνατότητες για αποστολή και προσπέλαση δεδομένων με τη χρήση του REST είναι απεριόριστες. Μπορούν για παράδειγμα να φτιαχτούν εφαρμογές για το Neo4j που θα χρησιμοποιούν REST Endpoints για την αλληλοεπίδραση του χρήστη με τη ΒΔΓ. Το Neo4j Browser τρέχει μέσω http τοπικά στην θύρα 7474 και μέσω bolt στο neo4j@bolt://localhost:7687/neo4j.

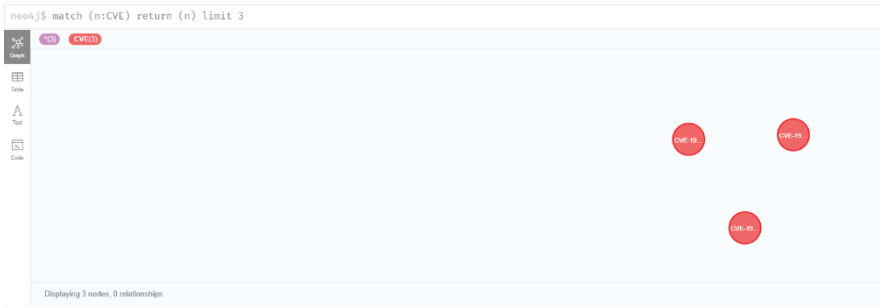
Για τους χρήστες που δεν είναι εξοικειωμένοι με τον προγραμματισμό, η πλατφόρμα τους δίνει τη δυνατότητα να αλληλοεπιδράσουν με τη ΒΔΓ με τη χρήση του Neo4j Bloom, το οποίο αποτελεί μια αυτοτελή εφαρμογή, παρόμοια με το Neo4j Browser που ωστόσο, έχει πολύ μεγαλύτερες δυνατότητες για αλλαγές στην οπτική αναπαράσταση των γράφων, και δεν απαιτεί από τον χρήστη την συγγραφή κώδικα cypher, αλλά προσφέρει μια περιήγηση σχεδόν σε φυσική γλώσσα, για την προσπέλαση και επεξεργασία των δεδομένων.

Παρακάτω βλέπουμε σε screenshots κάποια βασικά στοιχεία του Neo4j Browser και του Neo4j Bloom. Στην Εικόνα 22, έχουμε την αρχική σελίδα του Neo4j Browser. Στο πεδίο Database Information έχουμε κάποια βασικά στοιχεία της βάσης που χρησιμοποιούμε όπως τα Node Labels & Relationship Labels και την επιλογή αλλαγής σε κάποια άλλη βάση. Παρακάτω αριστερά έχουμε

το πεδίο των αγαπημένων cypher εντολών που έχουμε αποθηκεύσει. Παρακάτω έχουμε αρχεία με εντολές που μπορούμε να κάνουμε import και στο τέλος της πρώτης ομάδας αριστερά έχουμε παραδείγματα εντολών και datasets για να κατανοήσουμε καλύτερα την Cypher.Στην επόμενη ομάδα εικονιδίων αριστερά, έχουμε δύο ενημερωτικά πεδία και ένα πεδίο ρυθμίσεων.

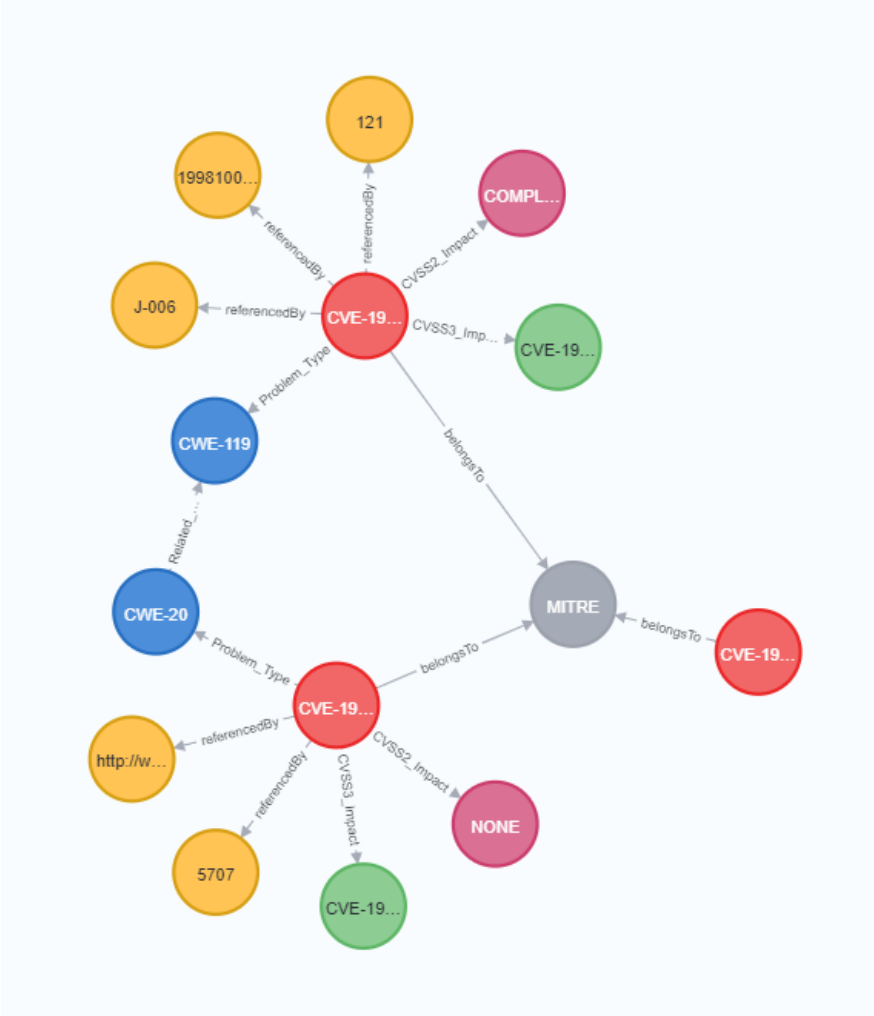


Εικόνα 22: Η αρχική σελίδα του Neo4j Browser.

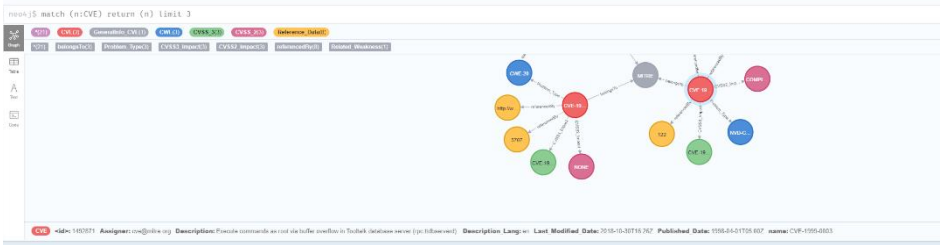


Εικόνα 23: Το αποτέλεσμα από μια Cypher εντολή. Μπορεί να αναπαρασταθεί με γράφο, με πίνακα, με plain text και με ανάλυση της εκτέλεσης του κώδικα.

Με διπλό κλικ στους κόμβους εκτυπώνονται στο ίδιο πεδίο οι σχέσεις καθώς και οι συσχετιζόμενοι κόμβοι (Εικόνα 24). Στην εικόνα αυτή βλέπουμε τα CVEs με κόκκινο, τα σχετικά CWEs με μπλε, τα References που έχουν γίνει για αυτά με κίτρινο, τα σκορ σε CVSS3 & CVSS2 με πράσινο και ροζ αντίστοιχα και τα γενικά δεδομένα του dataset στο οποίο υπάγονται με γκρι.



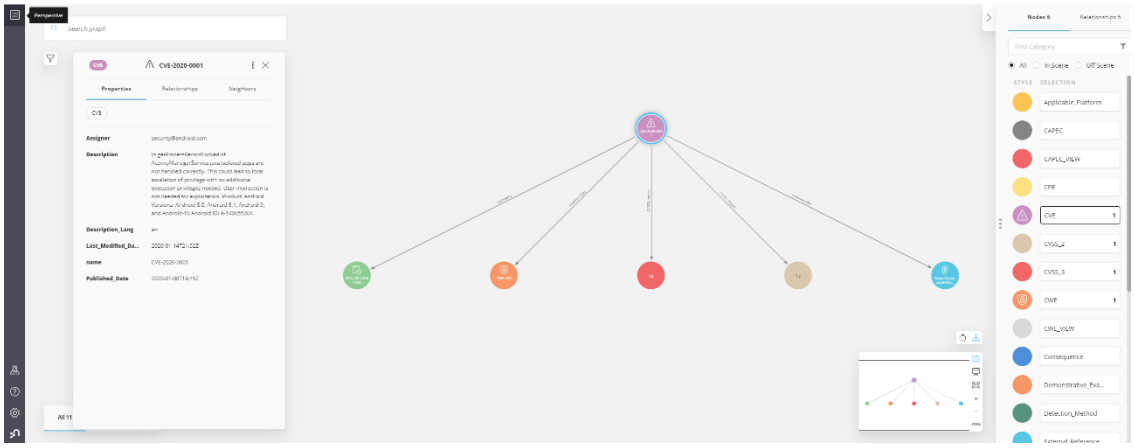
Εικόνα 24: Με διπλό κλικ στους κόμβους εκτυπώνονται στο ίδιο πεδίο οι σχέσεις καθώς και οι συσχετιζόμενοι κόμβοι.



Εικόνα 25: Σε οποιαδήποτε επιλογή κόμβου αναγράφονται παρακάτω όλα τα χαρακτηριστικά του με τις τιμές τους.

Στην εικόνα 26 βλέπουμε το περιβάλλον του Neo4j Bloom. Της εφαρμογής όπου είναι πιο φιλική προς τον χρήστη που δεν είναι προσαρμοσμένος σε πιο προγραμματιστικό περιβάλλον. Όπως φαίνεται, μπορούμε να επιλέξουμε εικονίδιο (και να εισάγουμε και δικά μας) για το κάθε Label των κόμβων, χρώμα, μέγεθος κοκ. ενώ η αναζήτηση γίνεται με full text search πάνω αριστερά.

Επίσης, η παρουσίαση των γράφων γίνεται με πολλούς διαφορετικούς τρόπους, εν προκειμένω έχουμε επιλέξει το ιεραρχικό μοντέλο υπό μορφή δέντρου.



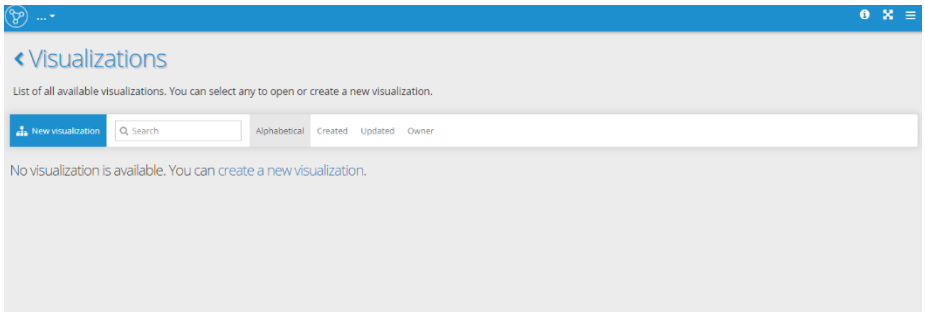
Εικόνα 26: Το περιβάλλον του Neo4j Bloom [54].

Graphlytic App

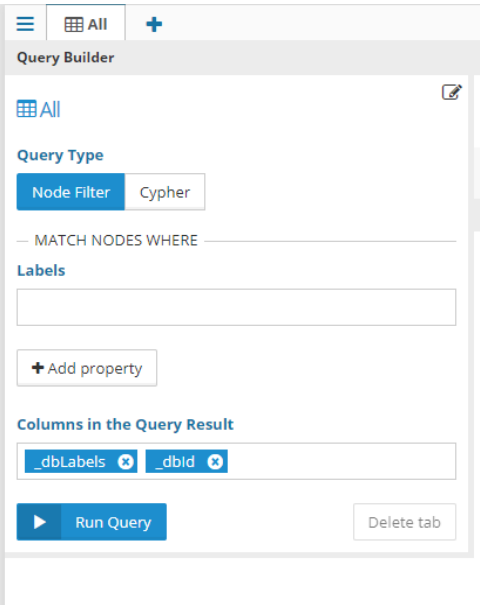
Η εφαρμογή Graphlytic είναι η δεύτερη εφαρμογή που χρησιμοποιούμε εντός του Neo4j (πέραν του bloom) με στόχο την βέλτιστη αναπαράσταση των γράφων και στην προκειμένη περίπτωση την εξαγωγή αναλυτικών δεδομένων [55]. Η εφαρμογή είναι δωρεάν για ιδιωτική χρήση ενώ είναι επί πληρωμή για εταιρική ή cloud χρήση. Τοπικά τρέχει μέσω browser στην θύρα 8110.

Η ουσιαστική διαφορά από το Neo4j Bloom, είναι αυτή της εξαγωγής στατιστικών στοιχείων και αναλυτικών δεδομένων από την βάση. Για κάθε ιδιότητα, έχουμε μια συνοπτική κατανομή των τιμών της, σύμφωνα με τους κόμβους και τα δεδομένα που έχουμε στον πίνακα. Ταυτόχρονα, τα στατιστικά ανανεώνονται αυτόματα, όταν επιλέγουμε παραπάνω κόμβους, όταν προσθέτουμε δεδομένα στην υπάρχουσα προεπισκόπηση κοκ. Οι κυριότερες μορφές αναπαράστασης των στατιστικών είναι με ιστόγραμμα και με λογαριθμική αναπαράσταση.

Πρώτο βήμα είναι η δημιουργία νέου visualization. Μπορούμε να δημιουργούμε και να αποθηκεύουμε visualizations, δηλαδή αναπαραστάσεις των δεδομένων της βάσης, ώστε να έχουμε σε καθένα τους γράφους που θέλουμε και με τον τρόπο τον οποίο θέλουμε (βλ. χρωματικές αλλαγές, εικονίδια κοκ.).



Εικόνα 27: Πρώτο βήμα είναι η δημιουργία νέου visualization.



Εικόνα 28: Μέσω του Advanced Search μπορούμε να αναζητήσουμε δεδομένα τόσο μέσα από full text search όσο και με Cypher εντολές.

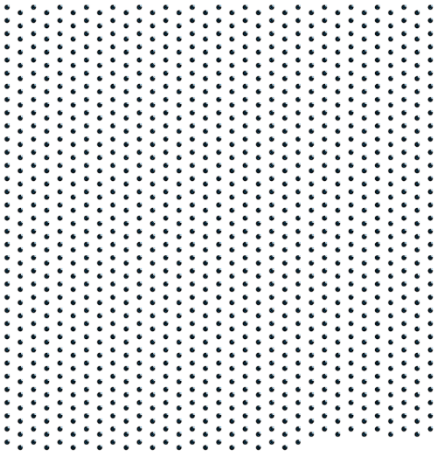
Έστω ότι ψάχνουμε τα CVEs του 2021 με CVSS3 Base Score = 10:

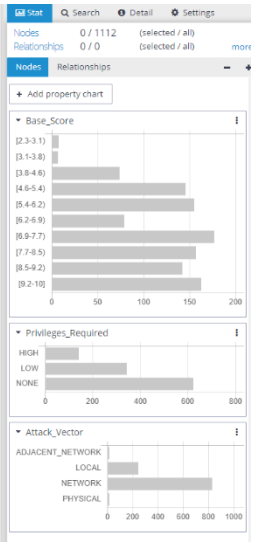
match(cve:CVE)-[:CVSS3_Impact]->(cvss:CVSS_3) where cvss.Base_Score = 10 and cve.name starts with ("CVE-2021") return (cve)

Εικόνα 29: 18 CVEs του 2021 είχαν CVSS3 Base Score ίσο με 10.

Για να μελετήσουμε τη δυνατότητα του Graphlytic να παράγει στατιστικά στοιχεία, ας πάρουμε ένα παράδειγμα. Θέλουμε να μελετήσουμε στοιχεία των CVSS3 μετρικών, για τα CVEs του Μαρτίου του 2020, που ήταν και ο πρώτος μήνας που παγκοσμίως είχαμε κορονοϊό.

match(cve:CVE)-[:CVSS3_Impact]->(cvss:CVSS_3) where cve.name starts with ("CVE-2020") and cve.Published_Date starts with ("2020-03") return (cvss)





Εικόνα 30: Επιλέξαμε να δούμε την κατανομή των Base Score, των Privileges Required και του Attack Vector για 1112 CVEs. Η εκτέλεση του query, η οπτικοποίηση και η εξαγωγή των στατιστικών ήταν άμεση (query + visualization < 1ms).

Τέλος, μπορούμε να αποθηκεύσουμε το περιβάλλον εργασίας που δημιουργήσαμε ή να εξάγουμε τον γράφο ως εικόνα και τα αποτελέσματά του ως αρχεία csv.

3.2 Πηγές Δεδομένων

Κατά την εκτέλεση του προγράμματος γίνεται λήψη των πιο πρόσφατων συλλογών δεδομένων που αφορούν τα CPEs, CVEs, CWEs και CAPECs, από τις επίσημες ιστοσελίδες των NIST & MITRE. Σε αυτήν την ενότητα θα μελετήσουμε συγκεκριμένα στοιχεία για κάθε συλλογή ξεχωριστά.

CPE: Common Platform Enumeration (Βλ. Ενότητα «Χρήσιμοι Ορισμοί» για σαφή ορισμό). Χρησιμοποιούμε τη συλλογή CPE-Match (συλλογή σε .json μορφή) από την επίσημη ιστοσελίδα της National Vulnerability Database του οργανισμού NIST. Μέσα σε αυτήν βρίσκονται πληροφορίες για όλες τις πλατφόρμες (λειτουργικά συστήματα, εφαρμογές, συσκευές, υλισμικό κοκ.) οι οποίες εμπλέκονται – αναφέρονται – εμπίπτουν σε γνωστές ευπάθειες. Κυριότερο στοιχείο του κάθε CPE εντός της λίστας είναι το uri, το οποίο αποτελεί ουσιαστικά τον τρόπο ονοματοποιίας. Από αυτήν τη συλλογή χρησιμοποιούμε μόνο το cpe23Uri δηλαδή το δοθέν uri για την πλατφόρμα της έκδοσης CPE 2.3. Πιο συγκεκριμένα το uri έχει την μορφή:

cpe:<cpe_version>:<part>:<vendor>:<product>:<version>:<update>:<edition>:<language>:<sw_edition>:<target_sw>:<target_hw>:<other>

- **cpe_version:** Η έκδοση της συλλογής. Στην προκειμένη περίπτωση 2.3.
- **part:** Οι πιθανές τιμές είναι 1) a – Applications, 2) h – Hardware, 3) o – Operating System. (Εννοιολογικά το part αναφέρεται και ως «type».)
- **vendor:** Το όνομα της εταιρίας που ανήκει η καταγεγραμμένη πλατφόρμα.
- **product:** Το όνομα της πλατφόρμας.
- **version:** Η έκδοση της πλατφόρμας.
- **update:** Η φάση στην οποία βρίσκεται η έκδοση (beta, update1, update2, SP κοκ.).
- **edition:** Ακριβής καθορισμός της έκδοσης της πλατφόρμας.
- **language:** Η γλώσσα της πλατφόρμας.

Παραδείγματα cpe – uri:

```
1.  cpe:2.3:a:qlik:qlikview_server:11.20:service_release_1:*:*:*:*:*:*
2.  cpe:2.3:o:verizon:fios_actiontec_mi424wr-gen31_router_firmware:40.19.36:*:*:*:*:*:*:*
3.  cpe:2.3:a:verizon:serialize-javascript:*:*:*:*:*:node.js:*:*
```

Μέσω της επίσημης ιστοσελίδας της συλλογής CPE, είναι διαθέσιμα και σχετικά μεταδεδομένα όπως το checksum με sha256, το συμπιεσμένο και αποσυμπιεσμένο μέγεθος του αρχείου και το timestamp της τελευταίας τροποποίησης της συλλογής.

Επίσημο Schema της συλλογής δεδομένων: https://csrc.nist.gov/schema/cpematch/feed/1.0/nvd_cpematch_feed_json_1.0.schema

CVE: Common Vulnerabilities and Exposures (Βλ. Ενότητα «Χρήσιμοι Ορισμοί» για σαφή ορισμό). Χρησιμοποιούμε τη συλλογή των καταλόγων CVEs που παρέχει η επίσημη ιστοσελίδα της National Vulnerability Database του οργανισμού NIST. Μέσα από το πρόγραμμα γίνεται λήψη και αποσυμπίεση όλων των CVE καταλόγων – αρχείων σε μορφή .json από το 2002 έως και την φετινή χρονιά καθώς και τα αρχεία CVE-Recent & CVE-Modified, τα οποία περιέχουν τα νεότερα καταγεγραμμένα CVEs και τα CVEs για τα οποία έγιναν αλλαγές και επεξεργασίες από τους αντίστοιχους αρμόδιους φορείς. Κάθε συλλογή δεδομένων CVE περιέχει τα πεδία CVE_data_type, CVE_data_format, CVE_data_version και τον πίνακα CVE_Items ο οποίος περιέχει τις εγγραφές των ευπαθειών. Κάθε εγγραφή για ευπάθειας, δηλαδή κάθε CVE πρέπει να περιέχει τουλάχιστον, το αναγνωριστικό ID, μια σύντομη περιγραφή και μια δημόσια αναφορά για την ευπάθεια. Από κει και πέρα στόχος είναι για κάθε ευπάθεια να υπάρχει μέτρηση του σκορ CVSS (βλ. Ενότητα «Χρήσιμοι Ορισμοί» για σαφή ορισμό) καθώς και άλλα δεδομένα όπως η κατηγορία ευπαθειών στην οποία ανήκει (CWE), η πλατφόρμα την οποία αφορά (CPE) και γενικώς όσο το δυνατόν περισσότερες πληροφορίες σύμφωνα με πεδία τα οποία θα μελετήσουμε σε επόμενη ενότητα.

Παράδειγμα της εγγραφής CVE με αναγνωριστικό «CVE-2021-24388»:


```
    "cve" : {
      "data_type" : "CVE",
      "data_format" : "MITRE",
      "data_version" : "4.0",
      "CVE_data_meta" : {
        "ID" : "CVE-2021-24388",
        "ASSIGNER" : contact@wpscan.com
      },
      "problemtype" : {
        "problemtype_data" : [ {
          "description" : [ {
            "lang" : "en",
            "value" : "CWE-79"
          } ]
        } ]
      },
      "references" : {
        "reference_data" : [ {
          "url" : "https://wpscan.com/vulnerability/e3f6576f-08cb-4278-8c79-3ef4d0b85cd9",
          "name" : "https://wpscan.com/vulnerability/e3f6576f-08cb-4278-8c79-3ef4d0b85cd9",
          "refsource" : "CONFIRM",
          "tags" : [ "Exploit", "Third Party Advisory" ]
        } ]
      },
      "description" : {
        "description_data" : [ {
          "lang" : "en",
          "value" : "In the VikRentCar Car Rental Management System WordPress plugin before 1.1.7, there is a
custom filed option by which we can manage all the fields that the users will have to fill in before saving
the order. However, the field name is not sanitised or escaped before being output back in the page, leading
to a stored Cross-Site Scripting issue. There is also no CSRF check done before saving the setting, allowing
attackers to make a logged in admin set arbitrary Custom Fields, including one with XSS payload in it."
        } ]
      }
    },
    "configurations" : {
      "CVE_data_version" : "4.0",
      "nodes" : [ {
        "operator" : "OR",
        "children" : [ ],
```

```
        "cpe_match" : [ {
          "vulnerable" : true,
          "cpe23Uri" : "cpe:2.3:a:e4j:vikrentcar_car_rental_management_system:*:*:*:*:*:wordpress:*:*",
          "versionEndExcluding" : "1.1.7",
          "cpe_name" : [ ]
        } ]
      } ]
    },
    "impact" : {
      "baseMetricV3" : {
        "cvssV3" : {
          "version" : "3.1",
          "vectorString" : "CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N",
          "attackVector" : "NETWORK",
          "attackComplexity" : "LOW",
          "privilegesRequired" : "LOW",
          "userInteraction" : "REQUIRED",
          "scope" : "CHANGED",
          "confidentialityImpact" : "LOW",
          "integrityImpact" : "LOW",
          "availabilityImpact" : "NONE",
          "baseScore" : 5.4,
          "baseSeverity" : "MEDIUM"
        },
        "exploitabilityScore" : 2.3,
        "impactScore" : 2.7
      },
      "baseMetricV2" : {
        "cvssV2" : {
          "version" : "2.0",
          "vectorString" : "AV:N/AC:M/Au:S/C:N/I:P/A:N",
          "accessVector" : "NETWORK",
          "accessComplexity" : "MEDIUM",
          "authentication" : "SINGLE",
          "confidentialityImpact" : "NONE",
          "integrityImpact" : "PARTIAL",
          "availabilityImpact" : "NONE",
          "baseScore" : 3.5
        },
        "severity" : "LOW",
        "exploitabilityScore" : 6.8,
        "impactScore" : 2.9,
        "acInsufInfo" : false,
        "obtainAllPrivilege" : false,
        "obtainUserPrivilege" : false,
        "obtainOtherPrivilege" : false,
        "userInteractionRequired" : true
      }
    },
    "publishedDate" : "2021-07-06T11:15Z",
    "lastModifiedDate" : "2021-07-09T14:23Z"
  }
```


Επίσημο Schema της συλλογής δεδομένων:
https://csrc.nist.gov/schema/nvd/feed/1.1/nvd_cve_feed_json_1.1.schema

CWE: Common Weakness Enumeration (Βλ. Ενότητα «Χρήσιμοι Ορισμοί» για σαφή ορισμό). Χρησιμοποιούμε τη συλλογή των CWEs που παρέχει η επίσημη ιστοσελίδα της MITRE. Η συλλογή περιλαμβάνει κατηγορίες ευπαθειών λογισμικού και υλισμικού. Δημιουργήθηκε ως πρωτοβουλία κοινότητας ερευνητών κυβερνοασφάλειας με στόχο την σωστή κατηγοριοποίηση και τον ορισμό των ευπαθειών. Επίσης η λίστα είναι εμπλουτισμένη με την καταγραφή συμπεριφορών, μηχανισμών και παραδειγμάτων εκμετάλλευσης και λοιπών πληροφοριών για την κάθε κατηγορία ευπάθειας. Η συλλογή ανήκει στην MITRE και ενημερώνεται από αυτήν καθώς και την κοινότητα ερευνητών CWE. Μέσω της εφαρμογής γίνεται λήψη της συλλογής σε μορφή .xml και μετατρέπεται σε μορφή .json για την βέλτιστη επεξεργασία της. Περιλαμβάνει τις επιμέρους λίστες Weaknesses, Categories, Views και External References. Πιο συγκεκριμένα:

- Weaknesses: Η λίστα αυτή περιλαμβάνει τις ευπάθειες και περιλαμβάνει σχετικές πληροφορίες για αυτές.
- Categories: Η λίστα αυτή κατηγοριοποιεί σε ακόμη ανώτερο επίπεδο τις ευπάθειες βάσει κοινών χαρακτηριστικών ή γνωρισμάτων. Τέτοια κοινά χαρακτηριστικά και γνωρίσματα

μπορεί να ποικίλουν όπως το περιβάλλον εμφάνισης, το πεδίο που λειτουργούν, τι πόρους χρησιμοποιούν κοκ.

- Views: Η λίστα αυτή αναπαριστά την οπτική βάσει της οποίας μπορεί κανείς να μελετήσει τις κατηγορίες ευπαθειών στη συλλογή. Υπάρχουν τρεις κατηγορίες Views:
    1. Graphs: Ιεραρχική αναπαράσταση των ευπαθειών βάσει συγκεκριμένων ενεργειών του χρήστη. Η αναπαράσταση ξεκινά με την κατηγορία (ανώτερο επίπεδο κατηγοριοποίησης) και ολοκληρώνεται με τις ευπάθειες (κατώτερο επίπεδο).
    2. Explicit Slices: Απλή λίστα εγγραφών που δεν σχετίζονται με συγκεκριμένες σχέσεις μεταξύ τους, αλλά μέσω εξωτερικών παραγόντων.
    3. Implicit Slices: Λίστα εγγραφών των ευπαθειών που σχετίζονται με συγκεκριμένες σχέσεις.
- External References: Η λίστα αυτή περιλαμβάνει συλλογή στοιχείων που χρησιμοποιούνται ως δημόσιες αναφορές (βλ. ερευνητικές εργασίες, δημόσια έγγραφα, μελέτες, αναφορές εκμετάλλευσης ευπαθειών κλπ.)

Παράδειγμα της εγγραφής CWE με αναγνωριστικό «1024» και όνομα «Comparison of Incompatible Types»:


```
{
"ID": "1024",
"Name": "Comparison of Incompatible Types",
"Abstraction": "Base",
"Structure": "Simple",
"Status": "Incomplete",
"Description": "The software performs a comparison between two entities, but the entities are of different,
incompatible types that cannot be guaranteed to provide correct results when they are directly compared.",
"Extended_Description": "In languages that are strictly typed but support casting/conversion, such as C or
C++, the programmer might assume that casting one entity to the same type as another entity will ensure that
the comparison will be performed correctly, but this cannot be guaranteed.  In languages that are not strictly
typed, such as PHP or JavaScript, there may be implicit casting/conversion to a type that the programmer is
unaware of, causing unexpected results; for example, the string \"123\" might be converted to a number type.
See examples.",
"Related_Weaknesses": {
"Related_Weakness": {
"Nature": "ChildOf",
"CWE_ID": "697",
"View_ID": "1000",
"Ordinal": "Primary"
}
},
"Weakness_Ordinalities": {
"Weakness_Ordinality": {
"Ordinality": "Primary"
}
},
"Applicable_Platforms": {
"Language": [
{
"Name": "JavaScript",
"Prevalence": "Undetermined"
},
{
"Name": "PHP",
"Prevalence": "Undetermined"
},
{
"Class": "Language-Independent",
"Prevalence": "Undetermined"
}
]
},
"Modes_Of_Introduction": {
"Introduction": {
```

```
"Phase": "Implementation"
}
},
"Common_Consequences": {
"Consequence": {
"Scope": "Other",
"Impact": "Varies by Context"
}
},
"Potential_Mitigations": {
"Mitigation": {
"Phase": "Testing",
"Description": "Thoroughly test the comparison scheme before deploying code into production. Perform positive testing as well as negative testing."
}
},
"Content_History": {
"Submission": {
"Submission_Name": "CWE Content Team",
"Submission_Organization": "MITRE",
"Submission_Date": "2018-01-04"
},
"Modification": {
"Modification_Name": "CWE Content Team",
"Modification_Organization": "MITRE",
"Modification_Date": "2020-02-24",
"Modification_Comment": "updated Relationships"
}
}
}
```


Επίσημο Schema της συλλογής δεδομένων: https://cwe.mitre.org/data/xsd/cwe_schema_latest.xsd

CAPEC: Common Attack Pattern Enumeration and Classification (Βλ. Ενότητα «Χρήσιμοι Ορισμοί» για σαφή ορισμό). Χρησιμοποιούμε την συλλογή των CAPECs που παρέχει η επίσημη ιστοσελίδα της MITRE. Μέσω της εφαρμογής γίνεται λήψη της συλλογής σε μορφή .xml και μετατρέπεται σε μορφή .json για βέλτιστη επεξεργασία. Η συλλογή αυτή αποτελεί μια προσπάθεια δημόσιας κατηγοριοποίησης και καταγραφής των γνωστών τρόπων ψηφιακής επίθεσης σε λογισμικό και υλισμικό. Έρχεται να ολοκληρώσει τον κύκλο των ψηφιακών κινδύνων, καθώς τα πρότυπα – μορφές επίθεσης εκμεταλλεύονται υπάρχουσες αδυναμίες στα πληροφοριακά συστήματα. Με αντίστοιχο τρόπο με την συλλογή των CWEs περιέχονται και εδώ επιμέρους λίστες Attack Patterns, Categories, Views και External References. Αντίστοιχα, η λίστα Attack Patterns περιέχει τις πληροφορίες για κάθε AP ξεχωριστά, η λίστα Categories, κατηγοριοποιεί σε ανώτερο επίπεδο τα attack patterns, η λίστα Views κατηγοριοποιεί τα Attack Patterns βάσει διαφορετικής οπτικής μελέτης τους και η λίστα External References περιέχει το υλικό με τις δημόσιες αναφορές για τα σχετικά Attack Patterns.

Παράδειγμα της εγγραφής CAPEC με αναγνωριστικό «107» και όνομα «Cross Site Tracing»:


```
{
                    "ID": "107",
                    "Name": "Cross Site Tracing",
                    "Abstraction": "Detailed",
                    "Status": "Draft",
                    "Description": "Cross Site Tracing (XST) enables an adversary to steal the victim's session cookie and possibly other authentication credentials transmitted in the header of the HTTP request when the victim's browser communicates to a destination system's web server. The adversary uses an XSS attack to have victim's browser sent an HTTP TRACE request to a destination web server, which will proceed to return a response to the victim's web browser that contains the original HTTP request in its body. Since the HTTP header of the original HTTP TRACE request had the victim's session cookie in it, that session cookie can now be picked off the HTTP TRACE response and sent to the adversary's malicious site. XST becomes relevant when direct access to the session cookie via the \"document.cookie\" object is disabled with the use of httpOnly attribute which ensures that the cookie can be transmitted in HTTP requests but cannot be accessed in other ways. Using SSL does not protect against XST. If the system with which the victim is interacting is susceptible to XSS, an adversary can exploit that weakness directly to get their malicious script to issue an HTTP TRACE request to the destination system's web server.",
                    "Likelihood_Of_Attack": "Medium",
                    "Typical_Severity": "Very High",
                    "Related_Attack_Patterns": {
                        "Related_Attack_Pattern": {
                            "Nature": "ChildOf",
                            "CAPEC_ID": "593"
```

```
                        }
                    },
                    "Execution_Flow": {
                        "Attack_Step": [
                            {
                                "Step": "1",
                                "Phase": "Explore",
                                "Description": "[Determine if HTTP Trace is enabled] Determine if HTTP Trace
is enabled at the web server with which the victim has an active session",
                                "Technique": "An adversary may issue an HTTP Trace request to the target web
server and observe if the response arrives with the original request in the body of the response."
                            },
                            {
                                "Step": "2",
                                "Phase": "Experiment",
                                "Description": "[Identify mechanism to launch HTTP Trace request] The
adversary attempts to force the victim to issue an HTTP Trace request to the targeted application.",
                                "Technique": "The adversary probes for cross-site scripting vulnerabilities to
force the victim into issuing an HTTP Trace request."
                            },
                            {
                                "Step": "3",
                                "Phase": "Exploit",
                                "Description": "[Create a malicious script that pings the web server with HTTP
TRACE request] The adversary creates a malicious script that will induce the victim's browser to issue an HTTP
TRACE request to the destination system's web server. The script will further intercept the response from the
web server, pick up sensitive information out of it, and forward to the site controlled by the adversary.",
                                "Technique": "The adversary's malicious script circumvents the httpOnly cookie
attribute that prevents from hijacking the victim's session cookie directly using document.cookie and instead
leverages the HTTP TRACE to catch this information from the header of the HTTP request once it is echoed back
from the web server in the body of the HTTP TRACE response."
                            },
                            {
                                "Step": "4",
                                "Phase": "Exploit",
                                "Description": "[Execute malicious HTTP Trace launching script] The adversary
leverages an XSS vulnerability to force the victim to execute the malicious HTTP Trace launching script"
                            },
                            {
                                "Step": "5",
                                "Phase": "Exploit",
                                "Description": "[Intercept HTTP TRACE response] The adversary's script
intercepts the HTTP TRACE response from teh web server, glance sensitive information from it, and forward that
information to a server controlled by the adversary."
                            }
                        ]
                    },
                    "Prerequisites": {
                        "Prerequisite": [
                            "HTTP TRACE is enabled on the web server",
                            "The destination system is susceptible to XSS or an adversary can leverage some
other weakness to bypass the same origin policy",
                            "Scripting is enabled in the client's browser",
                            "HTTP is used as the communication protocol between the server and the client"
                        ]
                    },
                    "Skills_Required": {
                        "Skill": {
                            "Level": "Medium",
                            "text": "Understanding of the HTTP protocol and an ability to craft a malicious
script"
                        }
                    },
                    "Resources_Required": {
                        "Resource": "None: No specialized resources are required to execute this type of
attack."
                    },
                    "Consequences": {
                        "Consequence": [
                            {
                                "Scope": "Confidentiality",
                                "Impact": "Read Data"
                            },
                            {
                                "Scope": [
                                    "Confidentiality",
                                    "Access Control",
                                    "Authorization"
```

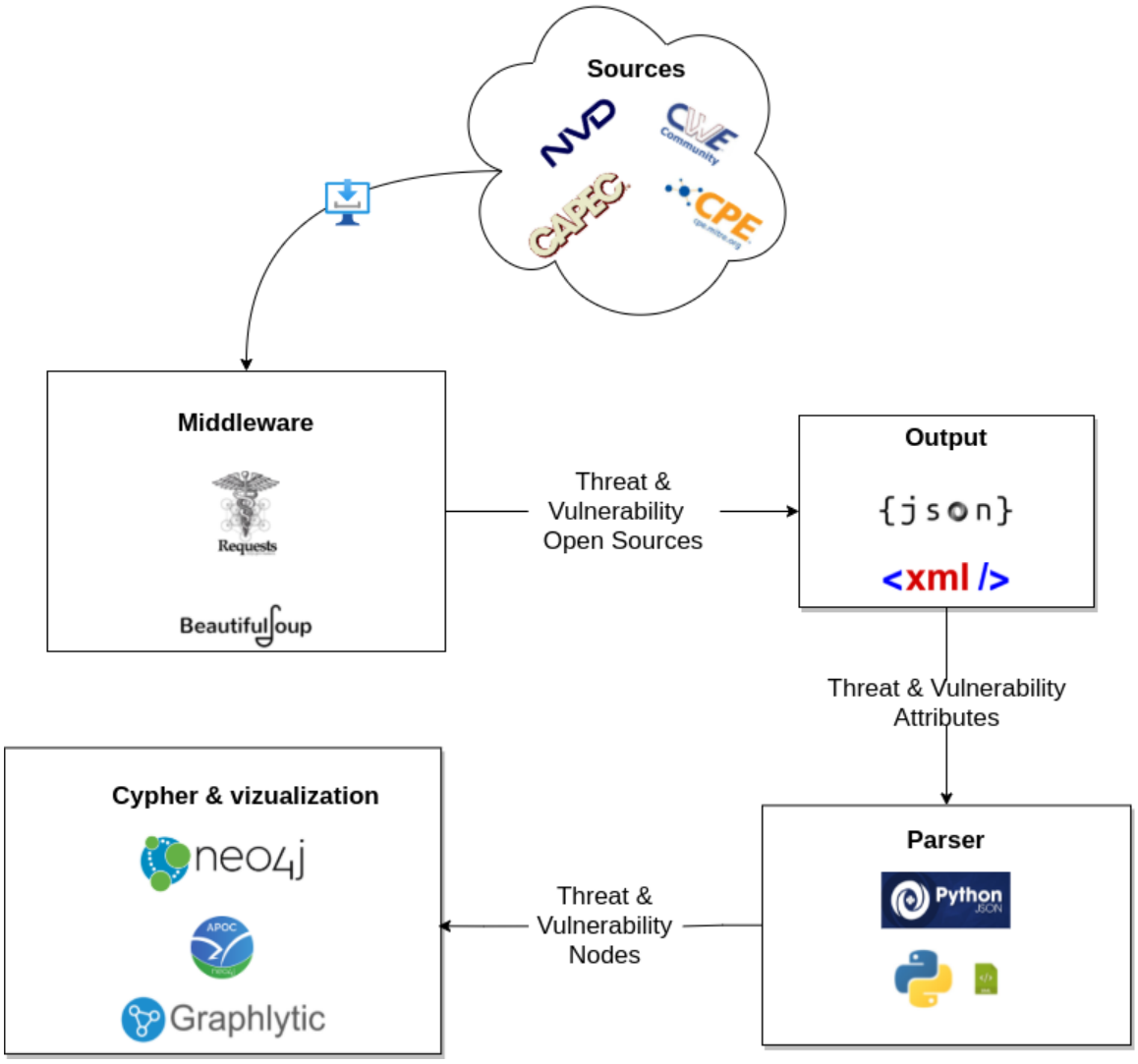
```
                                ],
                                "Impact": "Gain Privileges"
                            },
                            {
                                "Scope": "Integrity",
                                "Impact": "Modify Data"
                            }
                        ]
                    },
                    "Mitigations": {
                        "Mitigation": [
                            "Administrators should disable support for HTTP TRACE at the destination's web
server. Vendors should disable TRACE by default.",
                            "Patch web browser against known security origin policy bypass exploits."
                        ]
                    },
                    "Example_Instances": {
                        "Example": {
                            "xhtml:p": [
                                "An adversary determines that a particular system is vulnerable to reflected
cross-site scripting (XSS) and endeavors to leverage this weakness to steal the victim's authentication
cookie. An adversary realizes that since httpOnly attribute is set on the user's cookie, it is not possible to
steal it directly with their malicious script. Instead, the adversary has their script use XMLHTTP ActiveX
control in the victim's IE browser to issue an HTTP TRACE to the target system's server which has HTTP TRACE
enabled. The original HTTP TRACE request contains the session cookie and so does the echoed response. The
adversary picks the session cookie from the body of HTTP TRACE response and ships it to the adversary. The
adversary then uses the newly acquired victim's session cookie to impersonate the victim in the target
system.",
                                "In the absence of an XSS weakness on the site with which the victim is
interacting, an adversary can get the script to come from the site that they control and get it to execute in
the victim's browser (if they can trick the victim's into visiting their malicious website or clicking on the
link that they supplies). However, in that case, due to the same origin policy protection mechanism in the
browser, the adversary's malicious script cannot directly issue an HTTP TRACE request to the destination
system's web server because the malicious script did not originate at that domain. An adversary will then need
to find a way to exploit another weakness that would enable them to circumvent the same origin policy
protection."
                            ]
                        }
                    },
                    "Related_Weaknesses": {
                        "Related_Weakness": [
                            {
                                "CWE_ID": "693"
                            },
                            {
                                "CWE_ID": "648"
                            }
                        ]
                    },
                    "Taxonomy_Mappings": {
                        "Taxonomy_Mapping": {
                            "Taxonomy_Name": "OWASP Attacks",
                            "Entry_Name": "Cross Site Tracing"
                        }
                    },
                    "References": {
                        "Reference": {
                            "External_Reference_ID": "REF-3"
                        }
                    },
                    "Content_History": {
                        "Submission": {
                            "Submission_Name": "CAPEC Content Team",
                            "Submission_Organization": "The MITRE Corporation",
                            "Submission_Date": "2014-06-23"
                        },
                        "Modification": [
                            {
                                "Modification_Name": "CAPEC Content Team",
                                "Modification_Organization": "The MITRE Corporation",
                                "Modification_Date": "2017-05-01",
                                "Modification_Comment": "Updated Related_Attack_Patterns"
                            },
                            {
                                "Modification_Name": "CAPEC Content Team",
                                "Modification_Organization": "The MITRE Corporation",
                                "Modification_Date": "2017-08-04",
                                "Modification_Comment": "Updated Attack_Phases, Attack_Prerequisites,
```

```
Description Summary, Examples-Instances, Resources_Required"
                            },
                            {
                                "Modification_Name": "CAPEC Content Team",
                                "Modification_Organization": "The MITRE Corporation",
                                "Modification_Date": "2020-07-30",
                                "Modification_Comment": "Updated Description, Example_Instances"
                            },
                            {
                                "Modification_Name": "CAPEC Content Team",
                                "Modification_Organization": "The MITRE Corporation",
                                "Modification_Date": "2020-12-17",
                                "Modification_Comment": "Updated Description, Example_Instances,
Execution_Flow, Related_Attack_Patterns, Taxonomy_Mappings"
                            }
                        ]
                    }
                }
```


3.3 Αρχιτεκτονική Λογισμικού

Το λογισμικό που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής και υποστήριξε το ερευνητικό άρθρο «A Cybersecurity Ontology to Support Risk Information Gathering in Cyber-Physical Systems» [56] ακολουθεί μια συγκεκριμένη ροή δεδομένων που αξιοποιεί πέντε blocks:

- Τις γνωστές πηγές δεδομένων αδυναμιών OSCTI απο τις οποίες αντλεί τα απαραίτητα δεδομένα για την δημιουργία της τοπικής βάσης.
- Το middleware που τραβάει τα δεδομένα και τα μεταφράζει σε αρχεία που μπορούν να εισαχθούν στον κατάλληλο διερμηνέα.
- Τα αρχεία στην τελική μορφή πριν την εισαγωγή τους στον διερμηνέα.
- Ο python διερμηνέας που προετοιμάζει τα αρχεία και τα μεταφέρει στην βάση δεδομένων γράφων.
- Τέλος η βάση δεδομένων γράφων που απεικονίζει όλα τα δεδομένα μαζί με τις διασυνδέσεις που έχουν αναγνωριστεί μεταξύ τους.



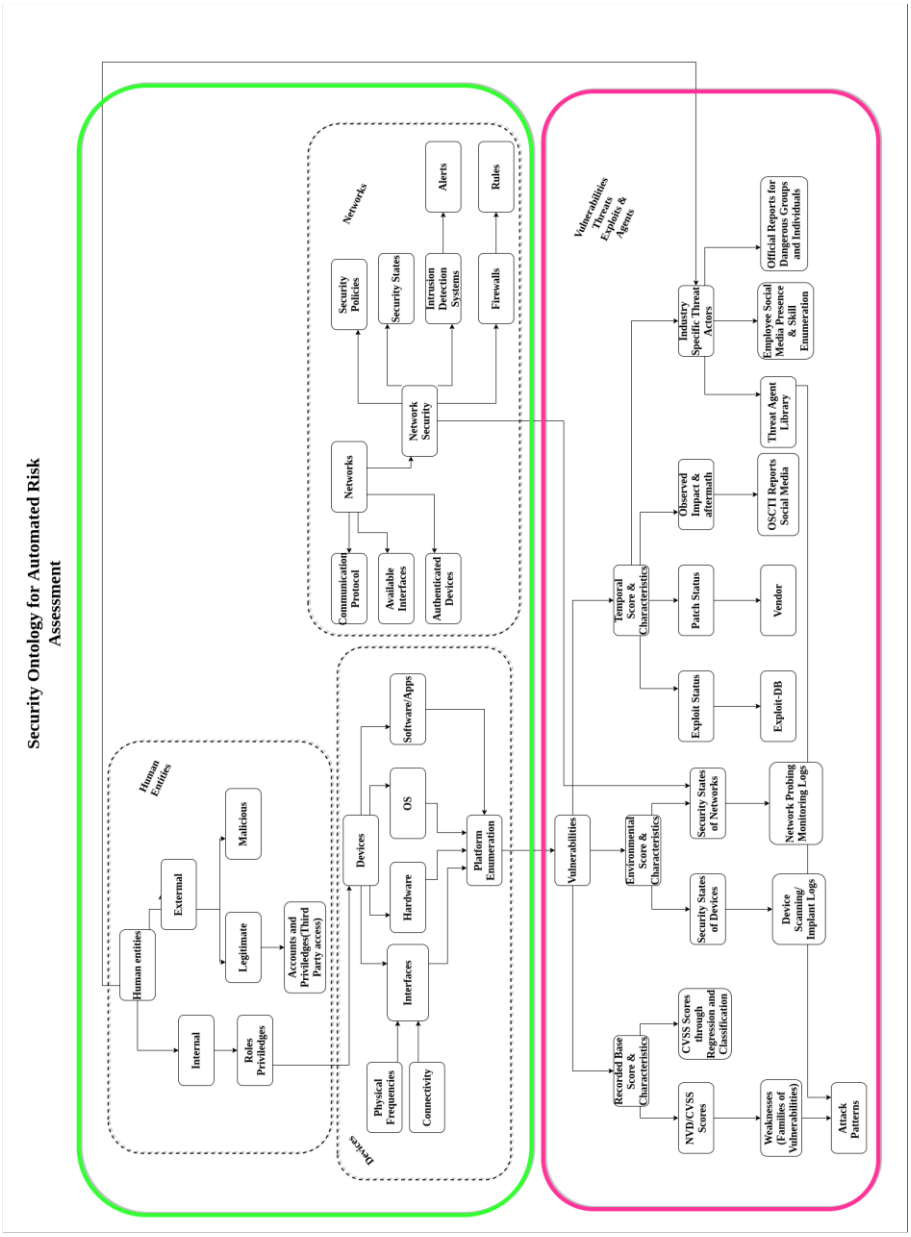
Εικόνα 31: Software Architecture High Design Level

3.4 Οντολογία

Στην οντολογία που χρησιμοποιεί ως υπόβαθρο η παραγόμενη βάση δεδομένων γράφου εκφράζουμε πληροφορίες που σχετίζονται με ευπάθειες, απειλές λογισμικού και κοινά μοτίβα επιθέσεων, που προέρχονται από δημόσιες πηγές δομημένων πληροφοριών. Για τις ευπάθειες που έχουμε αντλήσει από το NIST, υπάρχουν καταγεγραμμένες μεταβλητές που ακολουθούν τη δομή του μοντέλου CVSS που χαρακτηρίζει τις ευπάθειες χρησιμοποιώντας βασικά, περιβαλλοντικά και χρονικά διανύσματα χαρακτηριστικών. Οι μεταβλητές αυτές, αφορούν τα βασικά διανύσματα χαρακτηριστικών. Τα περιβαλλοντικά και χρονικά διανύσματα, αποτελούνται από δυναμικές μεταβλητές οι οποίες επηρεάζονται από το περιβάλλον στο οποίο παρουσιάζεται μια ευπάθεια και την χρονική στιγμή της παρουσίασης.

Τα παραπάνω δεδομένα μπορούν να συνδεθούν με πληροφορίες για ανθρώπινες οντότητες που καταγράφονται στο ανώτερο στρώμα, οι οποίες δρουν στο ίδιο περιβάλλον. Περισσότερες πληροφορίες σχετικά με τα χρονικά χαρακτηριστικά των ευπαθειών, όπως η

κατάσταση εκμετάλλευσης ή ενημέρωσης κώδικα, καθώς και η εμπιστοσύνη της αναφοράς μπορούν να αντληθούν από ανοιχτές πηγές. Για κάθε εγγραφή δημιουργείται ένα λεπτομερές υπό-γράφημα και πραγματοποιείται διεξοδική διερεύνηση για τις υπάρχουσες διασυνδέσεις που έχει με άλλες εγγραφές της ΒΔΓ.



Εικόνα 32: Security Ontology for Automated Risk Assessment [56].

Χρήση της οντολογίας για τη συσχέτιση κακόβουλων χρηστών με συγκεκριμένες ευπάθειες

Για τα περισσότερα μπλοκ της οντολογίας μας, οι διασυνδέσεις μπορούν να αντληθούν απευθείας από τις ανοιχτές πηγές που χρησιμοποιεί το εργαλείο και να εισαχθούν στο γράφημα γνώσης. Αυτό ισχύει για παράδειγμα για τα μπλοκ Vulnerability, Weakness και Attack Patterns. Ωστόσο, για άλλα μπλοκ, όπως τα Προφίλ Πράκτορα Απειλής, οι σχέσεις τους με τις υπόλοιπες

οντότητες οντολογίας δεν μπορούν να εξαχθούν άμεσα. Για να καταστεί δυνατή η υλοποίηση αυτών των σχέσεων εξετάζουμε τα χαρακτηριστικά τους και προσπαθούμε να ανακαλύψουμε ομοιότητες, με άλλες οντότητες που υπάρχουν στην ΒΔΓ. Επιπλέον, εξετάζουμε πολλαπλές προσεγγίσεις για την έκφραση προφίλ πρακτόρων απειλής, με σκοπό την επέκταση των αρχικών διανυσμάτων χαρακτηριστικών που περιγράφουν ευρέως τις εγγραφές. Ακολουθώντας αυτή την προσέγγιση, διευρύνονται οι δυνατότητες της εξαγωγής σχέσεων, καθιστώντας πιο πιθανό να αποφέρει ακριβή αποτελέσματα για συγκεκριμένες περιπτώσεις. Ένα πρακτικό σενάριο εξαγωγής σχέσεων μεταξύ παραγόντων απειλής, CWE, CVE και CPE, παράγεται χρησιμοποιώντας το CAPEC ως σκαλοπάτι.

Προκειμένου να χαρτογραφήσουμε τους παράγοντες απειλών στη μεθοδολογία μας, η αρχική μας προσέγγιση είναι να αντιγράψουμε τη μήτρα που παρουσιάζεται στο TAL της Intel [57]. Επιπλέον, επεκτείνουμε τα διανύσματα ιδιοτήτων των προφίλ TAL με διανύσματα ικανότητας CVSS. Επίσης, τα χαρακτηριστικά που παρέχει η Intel για τους πράκτορες απειλών περιλαμβάνουν πόρους, δεξιότητες και στόχους, ενώ παρόμοια χαρακτηριστικά παρατηρούνται στο σύνολο δεδομένων CAPEC, όπως φαίνεται στην παρακάτω εικόνα για να εκφραστούν οι απαιτήσεις για την εκτέλεση καταγεγραμμένων μοτίβων επίθεσης. Οι απαιτούμενες δεξιότητες για την ενεργοποίηση ενός μοτίβου επίθεσης μπορούν να συνδεθούν απευθείας με τις δεξιότητες που μπορεί να έχει ένας εισβολέας. Τα άλλα χαρακτηριστικά που παρουσιάζονται στον Πίνακα 5 απαιτούν περαιτέρω επεξεργασία προκειμένου να αντιστοιχιστούν, λόγω της περιγραφικής φύσης τους. Αυτό ισχύει ιδιαίτερα για χαρακτηριστικά όπως οι «απαιτούμενοι πόροι» και οι «συνέπειες», οι οποίες περιλαμβάνουν επίσης μια περιγραφή μαζί με το εύρος τους. Χρησιμοποιώντας αυτό το κοινό έδαφος, εξάγουμε περαιτέρω σχέσεις μεταξύ των «παραγόντων απειλής» που έχουμε κληρονομήσει στο γράφημα γνώσης από το TAL και τις μεμονωμένες εγγραφές CAPEC.

Στο [58] παρουσιάζονται παρόμοια χαρακτηριστικά για τα προφίλ παραγόντων απειλών ειδικά για το περιβάλλον υγειονομικής περίθαλψης, με μια μικρή ανατροπή. Αντί να χρησιμοποιείται μια κλίμακα [Χαμηλή, Υψηλή] για τις δυνατότητες ενός εισβολέα, σε αυτήν την περίπτωση οι δεξιότητες παρουσιάζονται ως διάνυσμα ικανότητας CVSS που μπορεί να συγκριθεί άμεσα με τα διανύσματα ευπάθειας CVSS των καταγεγραμμένων τρωτών σημείων. Αυτή η προσέγγιση χρησιμοποιείται για την επέκταση των προφίλ πρακτόρων απειλών του TAL, επιτρέποντας έτσι την αφαίρεση των σχέσεων μεταξύ των καταγεγραμμένων παραγόντων απειλών και των τρωτών σημείων στο γράφημα γνώσης μας.

TAL			CAPEC			
Skills	Resources	Objective	Skills Required	Resources Required	Typical Severity	Consequences Scope
None	Individual	Copy	None	Description	Very Low	Confidentiality
Minimal	Club	Deny	Low	-	Low	Integrity
Operational	Contest	Destroy	Medium	-	Medium	Availability
Adept	Team	Damage	High	-	High	-
-	Organization	Take	-	-	Very High	-
-	Government	All/None	-	-	-	-

Εικόνα 33: Similar attributes of Intel's TAL and CAPEC

4 Υλοποίηση και μελέτες περίπτωσης

4.1 Υλοποίηση

Η εφαρμογή «GraphKer» (συνδυασμός των λέξεων Graph και Hacker), η οποία υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας, αποτελεί αυτοτελής εφαρμογή που βασίζεται στην πλατφόρμα του Neo4j και λειτουργεί χωρίς γραφικό περιβάλλον διεπαφής μέσω της γραμμής εντολών. Στόχος της εφαρμογής είναι η χρήση της βάσης δεδομένων γράφων του Neo4j, για την αποθήκευση και γραφική αναπαράσταση ενός συνδυαστικού μοντέλου των γνωστών CPEs, CVEs, CWEs, CAPECs που είναι δημοσιευμένα στους οργανισμούς NIST & MITRE [59].

Όπως αναφέραμε και σε προηγούμενες ενότητες η χρήση των ΒΔΓ στην κυβερνοασφάλεια μπορεί να φέρει σημαντικά αποτελέσματα και πληροφορίες για την εξέλιξή της, και φυσικά την βελτίωση της ασφάλειας των οργανισμών και των γνώσεων των ειδικών του χώρου. Μέσω της εφαρμογής απευθυνόμαστε σε ερευνητές, ερασιτέχνες αλλά και επαγγελματίες που θέλουν να μελετήσουν ένα ολοκληρωμένο μοντέλο δεδομένων κυβερνοασφάλειας, των οποίων ο συνδυασμός θα αποφέρει γνώσεις και συμπεράσματα που δεν ήταν δυνατόν να εξαχθούν με την μερική μελέτη ενός ή και παραπάνω διαφορετικών συνόλων δεδομένων.

Η πρώτη έκδοση της εφαρμογής υλοποιήθηκε και παρουσιάστηκε στα πλαίσια της διπλωματικής εργασίας και είναι δημόσια διαθέσιμη μέσω GitHub. Όπως προαναφέρθηκε δεν έχει γραφικό περιβάλλον χρήσης, καθώς ο ρόλος της είναι η εισαγωγή και η μοντελοποίηση των δεδομένων σε κατάλληλους γράφους στην ΒΔΓ. Για την περιήγηση και μελέτη των δεδομένων συνίσταται η χρήση των Neo4j Browser, Neo4j Bloom και Graphlytic App. Σημειώνεται πως η μελέτη των δεδομένων και γράφων μπορεί να γίνει και με τη χρήση άλλων αυτοτελών εφαρμογών που υπάρχουν στην επίσημη λίστα του Neo4j όπως το GraphXR, το Neo4j Db Analyzer, το Charts κλπ.

Η εφαρμογή είναι υλοποιημένη με τη χρήση Python και Cypher και εκτελείται μέσω γραμμής εντολών στα λειτουργικά συστήματα (όλων των εκδόσεων) Windows & Linux (Debian Distributions) φυσικά με την εγκατάσταση συγκεκριμένων προαπαιτούμενων που αναγράφονται αναλυτικά στην ενότητα «Manual» [60]. Κατά την εκτέλεσή της εισάγονται και μοντελοποιούνται οι τελευταίες ενημερωμένες εκδόσεις των CPEs, CVEs, CWEs, CAPECs που είναι δημοσιευμένα στους οργανισμούς NIST & MITRE.

Στις παρακάτω ενότητες θα εμβαθύνουμε σε τρία επίπεδα:

1. Μελέτη των συνόλων δεδομένων των CPEs, CVEs, CWEs, CAPECs που είναι δημοσιευμένα στους οργανισμούς NIST & MITRE.
2. Μελέτη του τρόπου μοντελοποίησης για την εγγραφή στην ΒΔΓ και την αναπαράσταση σε γράφο.
3. Ανάλυση της προγραμματιστικής υλοποίησης της εφαρμογής.

Ενώ τέλος, θα παρουσιαστεί ο τρόπος εκτέλεσης της εφαρμογής και το σχετικό Manual που θα περιέχει τα προαπαιτούμενα της εφαρμογής, τον τρόπο εγκατάστασής τους, τις απαιτούμενες ρυθμίσεις κοκ.

4.1.1 Ανάλυση Προγραμματιστικής Υλοποίησης

Στην παρούσα ενότητα θα μελετήσουμε όλα τα στάδια της προγραμματιστικής υλοποίησης της εφαρμογής GraphKer, δηλαδή τον τρόπο με τον οποίο θα εισαχθούν τα δεδομένα των συλλογών με την κατάλληλη οντολογία που παρουσιάστηκε παραπάνω στην ΒΔΓ. Η εφαρμογή χωρίζεται σε τέσσερα στάδια:

1. Αλληλοεπίδραση με τον χρήστη, παρεχόμενες πληροφορίες λειτουργίας, τρόπος εκτέλεσης.

2. Λήψη και Αποσυμπίεση των συλλογών δεδομένων από τις επίσημες ιστοσελίδες.
3. Επεξεργασία και Καθορισμός των συλλογών δεδομένων και αρχείων εντολών που θα χρησιμοποιηθούν για την εισαγωγή στη βάση.
4. Αλληλοεπίδραση με την ΒΔΓ, εκτέλεση επερωτημάτων για την εισαγωγή των δεδομένων.

Παρακάτω θα αναλύσουμε κάθε στάδιο ξεχωριστά, όπου θα περιγράψουμε τα βήματα υλοποίησής του, καθώς και τα εξωτερικά πακέτα τα οποία χρησιμοποιούμε.

Αλληλοεπίδραση με τον χρήστη, παρεχόμενες πληροφορίες λειτουργίας, τρόπος εκτέλεσης

Στο στάδιο αυτό περιλαμβάνονται οι μέθοδοι **main** όπου γίνεται η επαφή με τον χρήστη μέσω παραμέτρων, επιλογών, ορισμάτων κοκ., **run** όπου βάσει των όσων όρισε ο χρήστης γίνεται η επιλογή των μεθόδων του κορμού που θα εκτελεστούν, **set_import_path** όπου δηλώνεται ο φάκελος που θα χρησιμοποιήσει το πρόγραμμα και το Neo4j για τις εισαγωγές αρχείων. Στις παρακάτω εικόνες φαίνεται η υλοποίηση των μεθόδων:

Χρησιμοποιώντας το πακέτο argparse της python θέτουμε παραμέτρους με την αντίστοιχη περιγραφή, για την αλληλοεπίδραση με τον χρήστη και την είσοδο ορισμάτων – πληροφοριών που θα χρειαστούν στην εκτέλεση του προγράμματος. Πιο συγκεκριμένα, υποχρεωτικά ο χρήστης εισάγει το bolt url της ΒΔΓ, το όνομα χρήστη, τον κωδικό του (για την βάση) και το directory path για τον φάκελο που χρησιμοποιεί για τα imports το Neo4j. Επίσης έχει την επιλογή μέσω του προγράμματος, να ανοίξει μετά το πέρας της εισαγωγής όλων των δεδομένων τον Neo4j Browser για να περιηγηθεί στη βάση, και την εφαρμογή Graphlytic, φυσικά σε περίπτωση που είναι εγκατεστημένη στο Neo4j.

```
def main():
    # Initialize the parser
    parser = argparse.ArgumentParser(
        description=" +-+-+-+-+-+-+-+-+ \n |G|r|a|p|h|K|e|r| \n +-+-+-+-+-+-+-+-+"
                    "\n \nWith GraphKer you can have the most recent update of cyber-security vulnerabilities, weaknesses, attack patterns and platforms "
                    "from MITRE and NIST, in an very useful and user friendly way provided by neo4j graph databases! \n \n"
                    "--Search, Export Data and Analytics, Enrich your Skills-- \n \n"
                    "**Created by Adamantios - Marios Berzovitis, Cyber-Security Expert MSc, BSc** \n"
                    "Diploma Research - MSc @ Distributed Systems, Security and Emerging Information Technologies | University Of Piraeus \n"
                    "LinkedIn:https://tinyurl.com/p57w4ntu \n"
                    "Github:https://github.com/amberzovitis \n \n"
                    "Enjoy! Provide Feedback!", formatter_class=argparse.RawTextHelpFormatter
    )

    # Add Parameters
    parser.add_argument('-u', '--urldb', required=True,
                        help="Insert bolt url of your neo4j graph database.")
    parser.add_argument('-n', '--username', required=True,
                        help="Insert username of your graph database.")
    parser.add_argument('-p', '--password', required=True,
                        help="Insert password of your graph database.")
    parser.add_argument('-d', '--directory', required=True,
                        help="Insert import path of your graph database.")
    parser.add_argument('-b', '--neo4jbrowser', choices=['y', 'Y'],
                        help="Press y or Y to open neo4jbrowser after the insertion of elements in your graph database.")
    parser.add_argument('-g', '--graphlytic', choices=['y', 'Y'],
                        help="Press y or Y to open Graphlytic app after the insertion of elements in your graph database.")

    args = parser.parse_args()
    if args.neo4jbrowser == "y" or args.neo4jbrowser == "Y":
        neo4jbrowser_open = True
    else:
        neo4jbrowser_open = False
    if args.graphlytic == "y" or args.neo4jbrowser == "Y":
        graphlytic_open = True
    else:
        graphlytic_open = False
    run(args.urldb, args.username, args.password,
        args.directory, neo4jbrowser_open, graphlytic_open)
    return
```

Εικόνα 34: Argparse

Στη μέθοδο run ορίζεται η σειρά εκτέλεσης του προγράμματος, καλώντας τις κατάλληλες μεθόδους. Περνώντας στο τελευταίο πεδίο της πρώτης φάσης, δηλαδή την κλήση της set_import_path, συνεχίζουμε με το δεύτερο και τρίτο στάδιο, δηλαδή την λήψη και αποσυμπίεση των συλλογών δεδομένων στον φάκελο (διαγράφουμε όλα τα αρχεία που πιθανώς να υπάρχουν

σε αυτόν από προηγούμενη εκτέλεση) και την κατάλληλη επεξεργασία των συλλογών δεδομένων και αρχείων Cypher Scripts ώστε να είναι όλα έτοιμα για την αλληλοεπίδραση με την ΒΔΓ.

```
# Define the functions that will be running
def run(url_db, username, password, directory, neo4jbrowser, graphlytic):
    set_import_path(directory)

    clear_directory()
    scrapper.download_datasets(import_path)
    xml_to_json()
    replace_unwanted_string_cwe()
    replace_unwanted_string_capec()
    copy_files_cypher_script()

    app = App(url_db, username, password)
    app.clear()
    app.close()

    app = App(url_db, username, password)
    app.schema_script()
    app.cve_insertion()
    app.cwe_insertion()
    app.capec_insertion()
    app.cpe_insertion()
    app.close()

    if neo4jbrowser:
        webbrowser.open("http://localhost:7474")
    if graphlytic:
        webbrowser.open("http://localhost:8110/")
    return
```

Εικόνα 35: run

Θέτουμε τη μεταβλητή του path σε global μορφή για να είναι καθολικά προσβάσιμη από όλες τις μεθόδους. Χρησιμοποιούμε επίσης το platform πακέτο της python για να έχουμε πρόσβαση σε δεδομένα του συστήματος που έχει το πρόγραμμα, εν προκειμένω με το platform.system στο λειτουργικό σύστημα, για να ορίσουμε σωστά τον import φάκελο του Neo4j.

```
# Set Import Directory
def set_import_path(directory):
    global import_path
    current_os = platform.system()
    if current_os == "Linux":
        import_path = directory
    elif current_os == "Windows":
        import_path = directory.replace("\\", "\\\\") + "\\\\"
```

Εικόνα 36: Import Path

Λήψη και Αποσυμπίεση των συλλογών δεδομένων από τις επίσημες ιστοσελίδες

Αφού έχει οριστεί το import path όπου θα γίνει η λήψη των συλλογών δεδομένων, προχωράμε στην διαγραφή όλων των αρχείων, στον πλήρη καθαρισμό του φακέλου της βάσης (όπου μπορεί να υπάρχουν αρχεία από προηγούμενες εκτελέσεις κοκ.) με τη μέθοδο **clear_directory**. Έπειτα καλούμε την μέθοδο **download_datasets** του **scraper.py**. Από εκεί διαδοχικά καλούνται οι μέθοδοι **download_files_cve_cpe, download_files_cwe** και **download_files_capec** όπου κατεβάζουν από τις επίσημες ιστοσελίδες, τις συλλογές δεδομένων και σε περίπτωση που είναι σε μορφή .zip τις αποσυμπιέζουν. Παρακάτω σε ενδεικτικές εικόνες θα παρουσιαστεί ο τρόπος λήψης και αποσυμπίεσης των συλλογών δεδομένων.

```python
def download_datasets(import_path):
    download_files_cve_cpe(import_path)
    download_files_cwe(import_path)
    download_files_capec(import_path)
```

Εικόνα 37: Μέσω αυτής της μεθόδου επικοινωνεί το main.py με το scraper.py για την λήψη και αποσυμπίεση των συλλογών δεδομένων.

Χρησιμοποιούμε τα πακέτα os, requests, zipfile, BeautifulSoup και platform. Συνοπτικά, βρίσκουμε από την ιστοσελίδα τα urls των αρχείων, γράφουμε το περιεχόμενο τους στον import φάκελο, και έπειτα τα αποσυμπιέζουμε διαγράφοντας, το αρχικό .zip αρχείο που πλέον δεν είναι χρήσιμο. Οι μέθοδοι για την λήψη των συλλογών cwe & capec ακολουθούν την ίδια λογική με τη διαφορά πως η συλλογή δεδομένων των capec δεν χρειάζεται αποσυμπίεση καθώς κάνουμε απευθείας λήψη αρχείου .xml.

```python
def download_files_cve_cpe(import_path):
    url = 'https://nvd.nist.gov/vuln/data-feeds'
    root = 'https://nvd.nist.gov/'
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'html.parser')
    all_hrfs = soup.find_all('a')
    all_links = [
        link.get('href') for link in all_hrfs
    ]
    zip_files = [
        dl for dl in all_links if dl and '.json.zip' in dl
    ]
    download_folder = import_path

    # Download and Unzip the files
    print('\nUpdating the Database with the latest CVE Files...')
    tries = 0
    for zip_file in zip_files:
        full_url = root + zip_file
        zip_filename = os.path.basename(zip_file)
        print(zip_filename)
        dl_path = os.path.join(download_folder, zip_filename)
        # 5 attempts to download and unzip the file correctly
        extract_dir = import_path
        while tries < 5:
            r = requests.get(full_url)
            dl_path = os.path.join(download_folder, zip_filename)
            with open(dl_path, 'wb') as z_file:
                z_file.write(r.content)
            # unzip
            try:
                z = zipfile.ZipFile(dl_path)
                z.extractall(os.path.join(download_folder, extract_dir))
                print(zip_filename + ' unzipped successfully')
                print('---------')
                z.close()
                current_os = platform.system()
                if current_os == "Linux":
                    file_to_delete = f'{extract_dir}' + f'/{zip_filename}'
                elif current_os == "Windows":
                    file_to_delete = f'{extract_dir}' + f'\\{zip_filename}'
                os.remove(file_to_delete)
                break
            except zipfile.BadZipfile:
                # Bad download, try again
                pass
            tries += 1
```

Εικόνα 38: Imports

Επεξεργασία και Καθορισμός των συλλογών δεδομένων και αρχείων εντολών που θα χρησιμοποιηθούν για την εισαγωγή στη βάση

Σε αυτό το στάδιο επεξεργαζόμαστε συλλογές δεδομένων και αρχεία εντολών, ώστε 1) να είναι επεξεργάσιμα με τον τρόπο και τη μορφή που θέλουμε, δηλαδή .json αρχεία και 2) να βρίσκονται στον import φάκελο. Πιο συγκεκριμένα χρησιμοποιούμε τη μέθοδο **xml_to_json** για να μετατρέψουμε σε json τις συλλογές δεδομένων των cwe & capec. Έπειτα με τις μεθόδους **replace_unwanted_string_cwe** & **replace_unwanted_string_capec** αφαιρούμε κάποιους ανεπιθύμητους χαρακτήρες από τις συλλογές, ώστε να είναι πιο εύκολα αξιοποιήσιμες στο επόμενο στάδιο της επικοινωνίας με την ΒΔΓ. Τέλος, μέσω της **copy_files_cypher_script** αντιγράφουμε τα αρχεία εντολών .cypher που αφορούν το σχήμα της βάσης από τον φάκελο Scripts (υποφάκελο σε αυτόν του προγράμματος) στον import φάκελο. Παρακάτω ενδεικτικά δείχνουμε τη χρήση κάποιων μεθόδων του βήματος (3).

Με την μέθοδο xml_to_json μετατρέπουμε τα δύο αρχεία .xml σε .json, συγκεκριμένα τα αρχεία για τα CWEs και τα CAPECs. Έτσι έχουμε να διαχειριστούμε μόνο .json αρχεία στη συνέχεια για

την εισαγωγή των δεδομένων στη βάση. Μιας και δεν χρειαζόμαστε πια τα .xml αρχεία, τα διαγράφουμε για την βέλτιστη χρήση του χώρου στον δίσκο.

```
def xml_to_json():
    # parse the import folder for xml files
    # open the input xml file and read
    # data in form of python dictionary
    # using xmltodict module
    for file in os.listdir(import_path):
        if file.endswith(".xml"):
            with open(import_path + f'{file}', encoding="utf8") as xml_file:
                data_dict = xmltodict.parse(xml_file.read())
                xml_file.close()
                # generate the object using json.dumps()
                # corresponding to json data
                json_data = json.dumps(data_dict)
                # Write the json data to output
                # json file
                jsonfile = import_path + f'{file}'
                jsonfile = jsonfile.replace(".xml", ".json")
                with open(jsonfile, "w") as json_file:
                    json_file.write(json_data)
                    json_file.close()
                os.remove(import_path + f'{file}')
```

Εικόνα 39: xml to json

Καθότι στα CWEs & CAPECs δεδομένα, έχουμε χαρακτήρες που θα δυσκολεύσουν την εισαγωγή τους στη βάση, προχωράμε σε μια σύντομη αλλαγή, πιο συγκεκριμένα διαγραφή ανεπιθύμητων χαρακτήρων, από το αρχείο .json τόσο για τα CWEs όσο και για τα CAPECs. Με την ίδια λογική κρατάμε το νέο αρχείο και διαγράφουμε το παλιό. Αντίστοιχη δουλειά πραγματοποιεί η μέθοδος replace_unwanted_string_capec.

```
# Flatten CWE Dataset File
def replace_unwanted_string_cwe():
    listOfFiles = os.listdir(import_path)
    pattern = "*.json"
    files = []
    for entry in listOfFiles:
        if fnmatch.fnmatch(entry, pattern):
            if entry.startswith("cwec"):
                files.append(entry)
                break
    file = import_path + files[0]
    fin = open(file, "rt")
    flattened_cwe = import_path + "cwe.json"
    fout = open(flattened_cwe, "wt")
    for line in fin:
        fout.write(line.replace('"@', '"'))
    fin.close()
    os.remove(file)
    fout.close()
```

Εικόνα 40: replace unwanted strings

Αλληλοεπίδραση με την ΒΔΓ, εκτέλεση επερωτημάτων για την εισαγωγή των δεδομένων

Στο τέταρτο και τελευταίο βήμα, περιέχονται όλες οι μέθοδοι που έμμεσα ή άμεσα συμβάλουν στην επικοινωνία με την ΒΔΓ και την εισαγωγή των δεδομένων.

Πιο συγκεκριμένα περιλαμβάνονται όλες οι μέθοδοι **της κλάσης App** καθώς και οι **files_to_insert_cpe, files_to_insert_capec, files_to_insert_cwe, files_to_insert_cve και replace_files_cypher_script.** Για κάθε μέθοδο files_to_insert καλούμε την replace_files_cypher_script ώστε δοθέντων των αρχείων για την αντίστοιχη συλλογή, να ενημερωθούν τα cypher αρχεία με τις αντίστοιχες αλλαγές και να περαστούν στον φάκελο import.

Στην κλάση App έχουμε όλες τις μεθόδους που λειτουργούν εντός της σύνδεσης που ανοίγεται μέσω του Neo4j Driver μεταξύ Python & Neo4j [61]. Συγκεκριμένα, **η _init_** είναι η σταθερή μέθοδος αρχικοποίησης της σύνδεσης, η **close** κλείνει τη σύνδεση με την ΒΔΓ, η **clear** και

clearSchema διαγράφουν δεδομένα και προϋπάρχοντα ευρετήρια στην ΒΔΓ, η **schema_script**, εισάγει στην βάση τα ευρετήρια, οι **query_cpe_script, query_cve_script, query_cwe_script, query_capec_script** εισάγουν τα αρχεία cyphers δηλαδή τα δεδομένα στην ΒΔΓ ενώ οι μέθοδοι **CVE,CAPEC,CWE,CPE_insertion** είναι οι μέθοδοι που στόχο έχουν να ορίσουν ποια αρχεία θα χρησιμοποιηθούν για την εισαγωγή των δεδομένων στη βάση και να καλέσουν τις μεθόδους **query_** που εκτελούν τα ερωτήματα.

Παρακάτω θα δούμε ενδεικτικά κάποιες μεθόδους του βήματος (4).

Μέρος της μεθόδου replace_files_cypher_script. Η μέθοδος αυτή όπως προαναφέρθηκε καλείται, κάθε φορά στο τέλος των files_to_insert, με όρισμα τα αρχεία για την κάθε συλλογή δεδομένων. Στα .cypher scripts που βρίσκονται στον φάκελο CypherScripts στο αρχείο του προγράμματος, η μεταβλητή filesToImport αλλάζει και ενημερώνεται σύμφωνα με τα αρχεία των συλλογών δεδομένων. Η μέθοδος φυσικά συνεχίζει με κάθε περίπτωση των δεδομένων, ακριβώς με τον ίδιο τρόπο.

```
# Copy Cypher Script files to Import Path
# Define Dataset Files in them
def replace_files_cypher_script(files):
    stringToInsert = "\""
    for file in files:
        stringToInsert += file + "\", \""
    stringToInsert = stringToInsert[:-3]

    current_path = os.getcwd()
    current_os = platform.system()
    if current_os == "Linux":
        current_path += "/CypherScripts/"
    elif current_os == "Windows":
        current_path += "\CypherScripts\\"

    if stringToInsert.startswith("\"nvdcpe"):
        toUpdate = current_path + "CPEs.cypher"
        fin = open(toUpdate, "rt")
        updatedFile = import_path + "CPEs.cypher"
        fout = open(updatedFile, "wt")
        for line in fin:
            fout.write(line.replace('filesToImport', stringToInsert))
        fin.close()
        fout.close()
    elif stringToInsert.startswith("\"nvdcve"):
        toUpdate = current_path + "CVEs.cypher"
        fin = open(toUpdate, "rt")
        updatedFile = import_path + "CVEs.cypher"
        fout = open(updatedFile, "wt")
        for line in fin:
            fout.write(line.replace('filesToImport', stringToInsert))
        fin.close()
        fout.close()
```

Εικόνα 41: replace_files_cypher_script

Η μέθοδος files_to_insert_cve έχει στόχο τον καθορισμό των αρχείων δεδομένων που θα χρησιμοποιήσουμε για την αντίστοιχη συλλογή, σύμφωνα με τα αρχεία που έχουμε στον φάκελο import. Τέλος καλεί την μέθοδο για την ενημέρωση των αρχείων .cypher. Ομοίως δουλεύουν και οι μέθοδοι για τις υπόλοιπες συλλογές.

```
# Define which Dataset and Cypher files will be imported on CVE Insertion
def files_to_insert_cve():
    listOfFiles = os.listdir(import_path)
    pattern = "*.json"
    files = []
    for entry in listOfFiles:
        if fnmatch.fnmatch(entry, pattern):
            if entry.startswith("nvdcpe") or entry.startswith("capec") or entry.startswith("cwe"):
                continue
            files.append(entry)
    replace_files_cypher_script(files)
    return files
```

Εικόνα 42: files_to_insert_cve

Στην μέθοδο run καλούμε μέσω του instance της App την μέθοδο cve_insertion (αντίστοιχα και για τις υπόλοιπες). Οι μέθοδοι _insertion καλούν τις μεθόδους για τον καθορισμό των αρχείων της συλλογής, και τέλος (με ενημερωμένα και τα .cypher αρχεία) καλούν τελικά την μέθοδο

εκτέλεσης του ερωτήματος, που όπως φαίνεται λειτουργεί απλά καλώντας με τη χρήση της βιβλιοθήκης APOC το αντίστοιχο .cypher αρχείο. Μετά την επιτυχή εκτέλεση του, ενημερώνεται αντίστοιχα και ο χρήστης στο terminal.

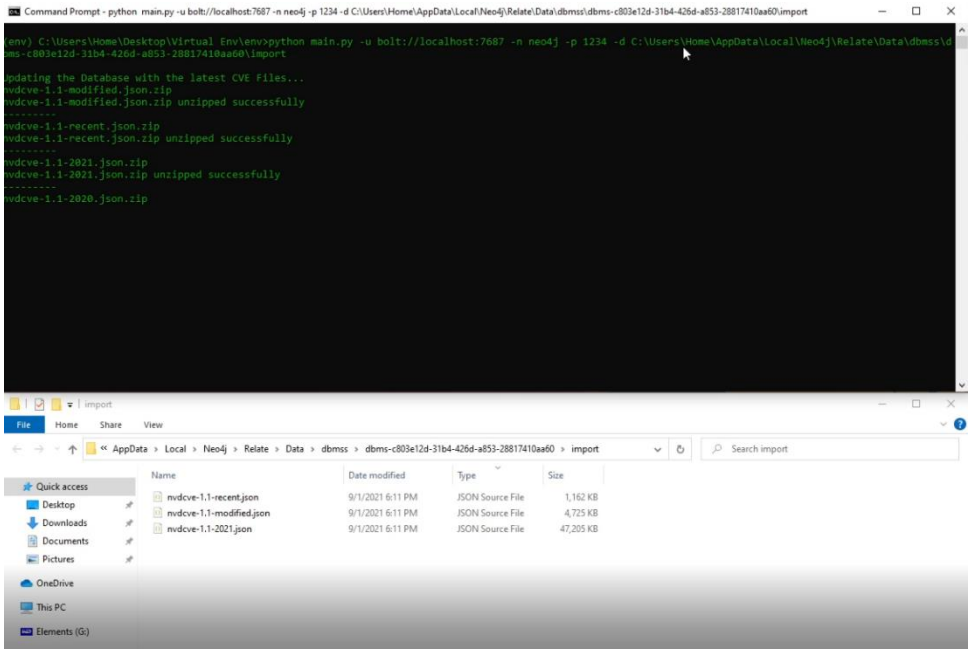
```
# Cypher Query to insert CVE Cypher Script
def query_cve_script(self, files):
    query = """CALL apoc.cypher.runFile("CVEs.cypher")"""
    session = self.driver.session()
    session.run(query)
    for file in files:
        print("\nCVE Files: " + file + " insertion completed. \n----------")

# Configure CVE Files and CVE Cypher Script for insertion
def cve_insertion(self):
    print("\nInserting CVE Files to Database...")
    files = files_to_insert_cve()
    for f in files:
        print('Inserting ' + f)
    self.query_cve_script(files)
```

Εικόνα 43: cve-insertion

Ο πηγαίος κώδικας του προγράμματος βρίσκεται δημόσια στο επίσημο GitHub Repository [amberzovitis/GraphKer: Open Source Tool - Cybersecurity Graph Database in Neo4j (github.com).]

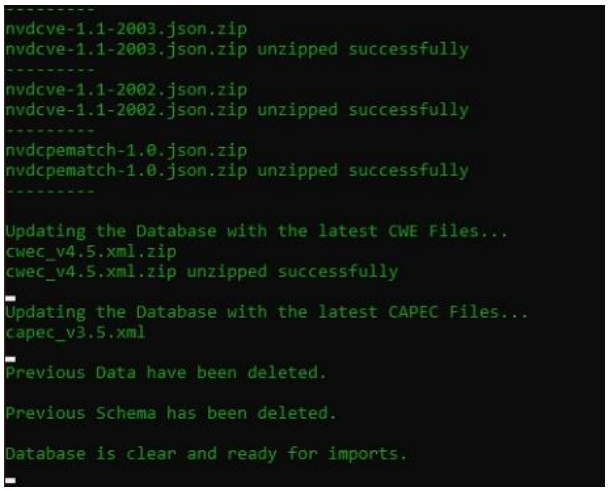
Δεδομένου ότι έχουμε προχωρήσει σε εγκατάσταση του Neo4j, και κατάλληλη παραμετροποίηση της ΒΔΓ στην οποία θα εισάγουμε τα δεδομένα, μπορούμε πια να εκτελέσουμε το πρόγραμμα. Στην προκειμένη περίπτωση των Windows 10, ο πλέον εύκολος τρόπος είναι η δημιουργία ενός python virtual environment όπου θα εγκαταστήσουμε τα απαραίτητα για την εκτέλεση πακέτα. Για την εκτέλεση θέτουμε υποχρεωτικά, το bolt url του Neo4j, το όνομα χρήστη και τον κωδικό στη ΒΔΓ, και το path του import φακέλου. Όπως φαίνεται παραπάνω το πρόγραμμα ξεκινά και κατεβάζει τα αρχεία.



Εικόνα 44: python virtual environment

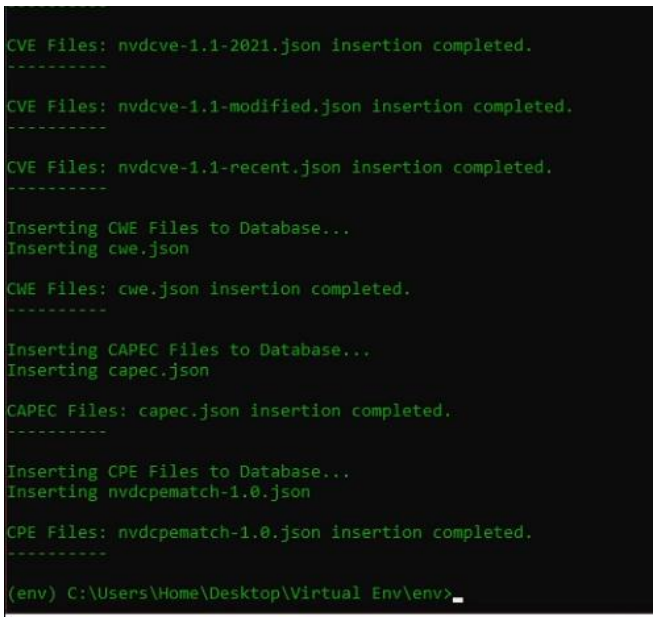
Μετά την λήψη όλων των αρχείων, το πρόγραμμα έχει ολοκληρώσει το βήμα 2 (βλ. Ενότητα Ανάλυσης Προγραμματιστικής Υλοποίησης) και το βήμα 3 και ξεκινά το βήμα 4 διαγράφοντας

οτιδήποτε προϋπήρχε στη βάση. Μην ξεχνάμε πως το πρόγραμμα εκτελείται παραπάνω από μια φορές στην ίδια βάση για να έχει τις πιο πρόσφατες ενημερώσεις των δεδομένων.



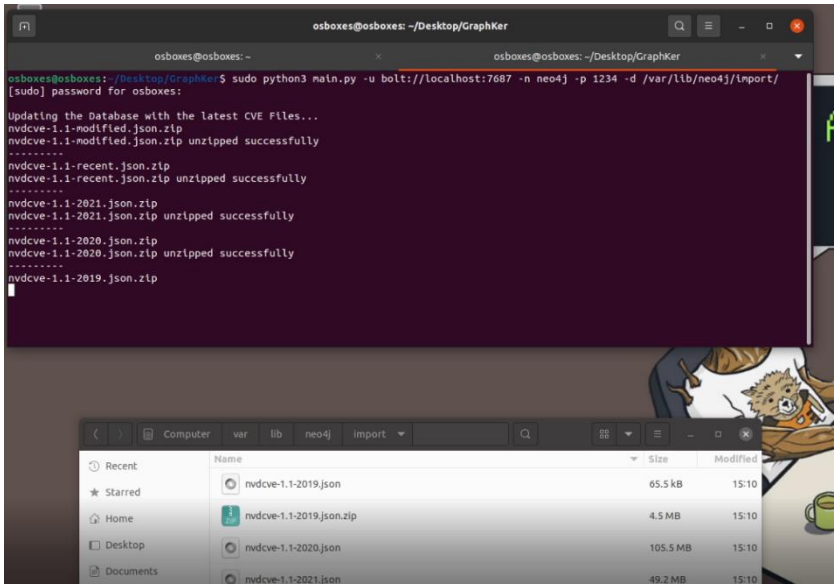
Εικόνα 45: Clear Database

Μετά την εισαγωγή όλων των δεδομένων το πρόγραμμα και δεδομένου ότι δεν έχουμε επιλέξει να ανοίξει μετά το τέλος της εισαγωγής το Neo4j Browser ή την εφαρμογή Graphlytic, το πρόγραμμα τερματίζει.



Εικόνα 46: Insertion Completed

Σε περιβάλλον Ubuntu 21.04 και γενικώς σε περιβάλλον Linux είναι πιο εύκολο για τον χρήστη αντί να φτιάξει virtual environment (κάτι το οποίο εξακολουθεί να έχει σαν επιλογή) να εγκαταστήσει απευθείας τα απαραίτητα για την εκτέλεση πακέτα. Προφανώς, έχοντας εγκαταστήσει το Neo4j και έχοντας παραμετροποιήσει κατάλληλα τη βάση, προχωράμε στην εκτέλεση του προγράμματος. Χρησιμοποιούμε sudo διότι επεμβαίνουμε με επεξεργασίες αρχείων σε φακέλους που μόνο διαχειριστής πρέπει να έχει τέτοια δικαιώματα (εκτός φυσικά αν έχουν εκχωρηθεί για τον αντίστοιχο χρήστη), και python3 καθώς κάποια από τα πακέτα λειτουργούν σωστά μόνο με την python3. Παρατηρούμε επίσης πως το import path φυσικά είναι διαφορετικό. Πλέον η εφαρμογή ξεκινάει και λειτουργεί ακριβώς με τον ίδιο τρόπο όπως και στα Windows.



Εικόνα 47: running on ubuntu

4.1.2 Μελέτες Περίπτωσης

Μελέτες Περίπτωσης

Για την ολοκληρωμένη ανάλυση και διαχείριση επικινδυνότητας σε ένα πληροφοριακό σύστημα, καθώς και για την συνεχή επαγρύπνηση ενός οργανισμού σε σχέση με την ψηφιακή του ασφάλεια, δεν αρκεί μια μελέτη που θα περιορίζεται στην εύρεση αδυναμιών, κινδύνων, και μέτρων αντιμετώπισης τους εντός των στενών ορίων του οργανισμού ή ακόμη και του πληροφοριακού συστήματος που εξετάζεται.

Για την συνεχή βελτιστοποίηση των διαδικασιών που ακολουθούνται για την ψηφιακή προστασία ενός οργανισμού, απαιτείται μια σύνθετη μελέτη που θα βλέπει πέραν όλων των άλλων, και τα φυσικά πρόσωπα που αποτελούν πιθανούς φορείς ψηφιακών επιθέσεων άμεσα ή έμμεσα. Μέσα από την μελέτη των πιθανών φορέων των ψηφιακών επιθέσεων, θα εξάγουμε όπως είναι λογικό, κοινά χαρακτηριστικά που μπορεί να έχουν, όπως για παράδειγμα το επίπεδο γνώσεων τους, το κίνητρο τους κοκ. Στην πραγματικότητα, θέλουμε να ορίσουμε άλλον έναν παράγοντα στη συνάρτηση της ψηφιακής προστασίας ενός οργανισμού, συνδυάζοντας τον με γνωστά υπάρχοντα πεδία, όπως οι τρόποι που μπορούν να πραγματοποιηθούν οι επιθέσεις, το επίπεδο ρίσκου και άλλα. Δηλαδή θέτουμε τον στόχο της αποδοτικής εξαγωγής γνώσεων και συμπερασμάτων, από μια ακόμη πιο σύνθετη δομή πληροφορίας.

Η διαδικασία αυτή δεν είναι διόλου άγνωστη για τους επιστήμονες της κυβερνοασφάλειας. Ερευνητικά κέντρα και οργανισμοί έχουν αναπτύξει μικρότερα ή μεγαλύτερα μοντέλα που συνδυάζουν τέτοιες πληροφορίες, καθώς και χαρτογραφήσεις διαφόρων φορέων επιθέσεων, ορίζοντας μάλιστα και συγκεκριμένα χαρακτηριστικά (όπως προαναφέραμε για παράδειγμα το επίπεδο γνώσεων).

Παρόλα αυτά, η χρήση τέτοιων μοντέλων δεν αποτελεί συνήθεια για τους οργανισμούς που αναπτύσσουν τα «τείχη προστασίας» τους. Για παράδειγμα, υπάρχει αδυναμία κατανόησης βασικών χαρακτηριστικών των φορέων επίθεσης, ενώ γενικεύσεις όπως «η επίθεση προήλθε από hackers» για την περιγραφή των πιθανών εισβολών σε πληροφοριακά συστήματα, δυσχεραίνουν την αντιμετώπιση τους, και φυσικά την πρόληψη αντίστοιχων μελλοντικών. Ακόμη και αν η ομάδα ψηφιακής ασφάλειας ενός οργανισμού συμφωνήσει πως υπάρχει η ανάγκη να οριστούν συγκεκριμένοι φορείς πιθανών επιθέσεων, ελλείψει συγκεκριμένων καθολικών προτύπων που θα ακολουθούνται, μια τέτοια διαδικασία θα είναι σταγόνα στον ωκεανό των όσων πρέπει να οριστούν. Ένα χαρακτηριστικό παράδειγμα της έλλειψης προτύπων είναι το πως, κακόβουλοι χρήστες και φορείς επιθέσεων που εισβάλουν σε πληροφοριακά συστήματα.

Μελέτη Περίπτωσης: Σάρωση ψηφιακού μηχανήματος για υπάρχοντα λογισμικά και λειτουργικά συστήματα

Ως περίπτωση χρήσης για την διερεύνηση της λειτουργικότητας του GraphKer διαλέξαμε την ανάλυση των αποτελεσμάτων του εργαλείου Nmap πάνω στο μηχάνημα Metasploitable 2. Μετά από ένα port scan για service & version enumeration ανιχνεύσαμε μια σειρά από υπάρχουσες υπηρεσίες και το λειτουργικό σύστημα του συστήματος υπό εξέταση. Μετά από μια αναζήτηση πάνω σε αυτή την πληροφορία αντιστοιχήσαμε ένα CPE uri σε κάθε ενεργή υπηρεσία. Για κάθε CPE uri που παράγουμε το χάρτη των συσχετιζόμενων:

- Ευπαθειών
- Αδυναμιών Λογισμικού
- Μοτίβων Επιθέσεων

Τα συστήματα που θα αναλυθούν είναι τα εξής:

- **cpe:/o:linux:linux_kernel:2.6 :**
- **cpe:/a:samba:samba::::::::**
- **cpe:/a:vsftpd_project:vsftpd:2.3.4:::::::**

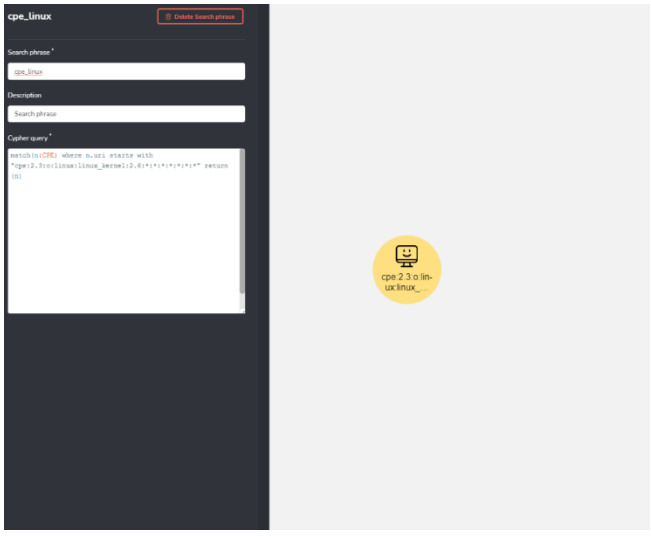
CPE 1 -OS

Η πρώτη αναζήτηση που τρέχουμε πάνω στο GraphKer επιβεβαιώνει πως το CPE uri του λειτουργικού συστήματος υπάρχει στη βάση και επιστρέφει τον κόμβο όπως φαίνεται στην Εικόνα 48.

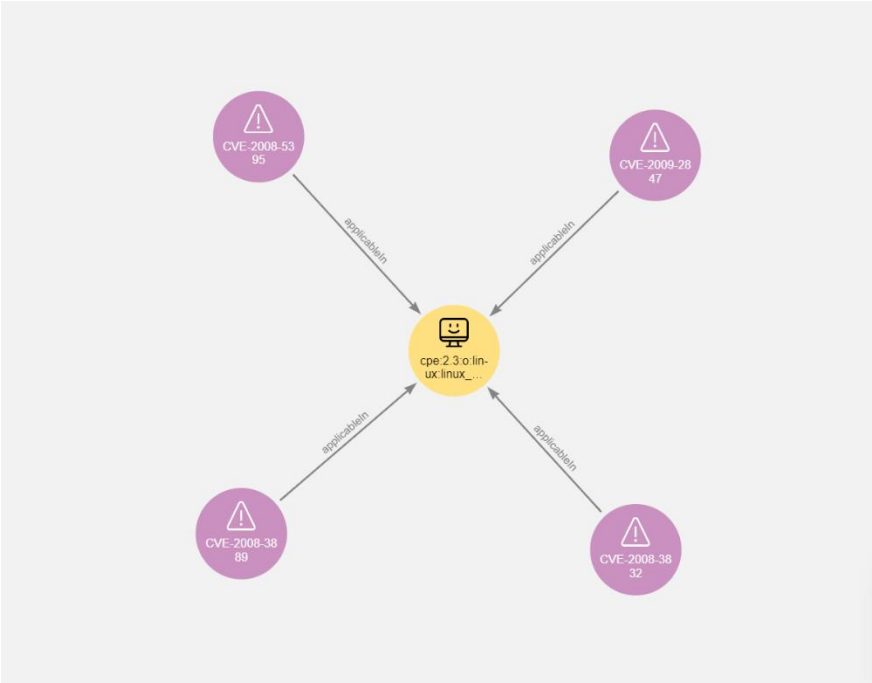
Στη συνέχεια για τον κόμβο που βρέθηκε αναζητούμε τις αριθμημένες ευπάθειες (CVE) όπως φαίνεται στην Εικόνα 49. Πάνω στις ευπάθειες που βρέθηκαν υπάρχει μια καταγεγραμμένη περιγραφή, τα χαρακτηριστικά που παράγουν το CVSS Score της συγκεκριμένης ευπάθειας καθώς και ορισμένες αναφορές σε καταγεγραμμένα exploit, patches και workarounds της ευπάθειας.

Τέλος, πραγματοποιείται μια αναζήτηση πάνω στις αδυναμίες λογισμικού και τα μοτίβα επιθέσεων που συνδέονται με τις ευπάθειες που βρέθηκαν στο προηγούμενο βήμα όπως φαίνεται στην Εικόνα 50.

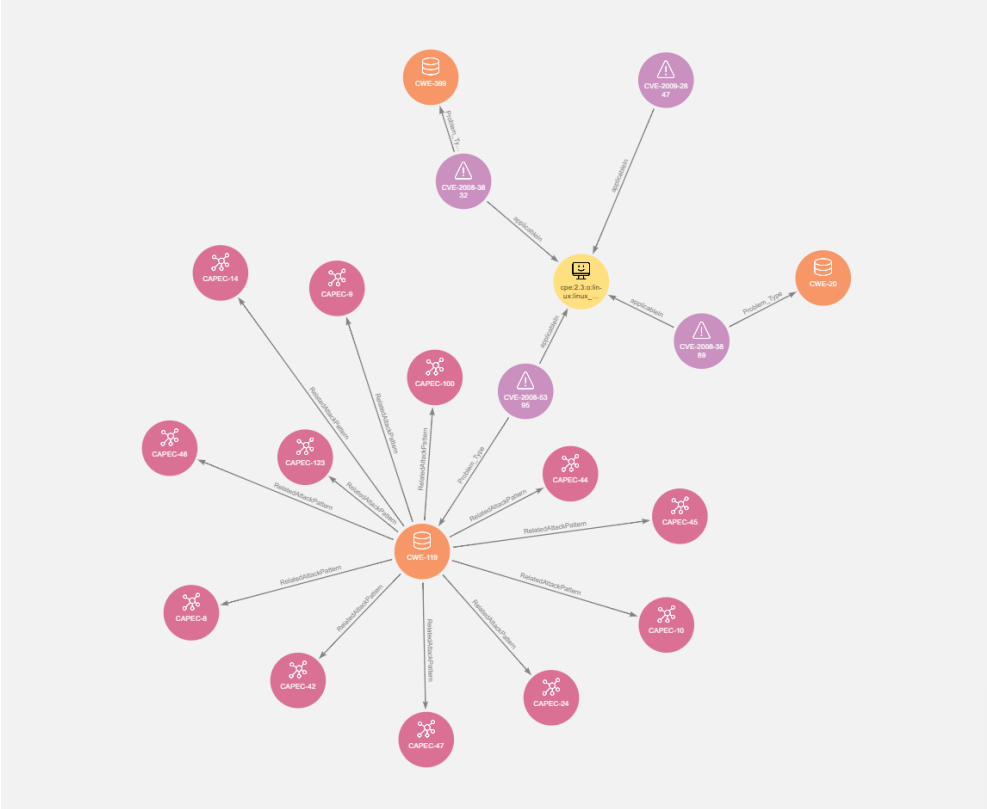
Οι αδυναμίες και τα μοτίβα επιθέσεων περιλαμβάνουν δικές τους περιγραφές καθώς και χαρακτηριστικά σχετικά με την επίδρασή τους στα συστήματα υπό μελέτη.



Εικόνα 48: OS CPE



Εικόνα 49: OS CPE Vulnerabilities



Εικόνα 50: OS CPE Vulnerabilities and Threats

Στον πίνακα παρουσιάζονται με λεπτομέρεια τα αποτελέσματα της αναζήτησης που πραγματοποιήθηκε. Το λειτουργικό σύστημα παρουσιάζει πολλαπλές ευπάθειες, αδυναμίες και μοτίβα επιθέσεων, δεδομένα που δύναται να χρησιμοποιηθούν στο πλαίσιο της ανάλυσης ρίσκου για τα διάφορα υποσυστήματα ενός οργανισμού. Αυτες οι πληροφορίες μπορούν να συνδυαστούν για να παράξουν σενάρια ως εξής:

Το linux kernel 2.6 παρουσιάζει την ευπάθεια CVE-2008-5395, μέσω της οποίας ένας χρήστης που έχει λογική πρόσβαση στο μηχάνημα μπορεί να προκαλέσει άρνηση υπηρεσίας εκμεταλευόμενος τη συνάρτηση του συστήματος parisc_show_stack function. Η ευπάθεια αυτή ανήκει στην κατηγορία αδυναμιών λογισμικού CWE-119 που αναφέρεται σε μη επαρκή περιορισμό λειτουργιών εντός των ορίων ενός buffer μνήμης. Ουσιαστικά η ευπάθεια αναφέρεται στην ύπαρξη της συγκεκριμένης αδυναμίας στο συγκεκριμένο λογισμικό, μπορεί να γίνει εκμεταλεύσιμη μέσω του μοτίβου επίθεσης CAPEC 10, δηλαδή μέσω υπερχείλισης buffer σε μεταβλητές περιβάλλοντος. Για το μοτίβο επίθεσης εξετάζουμε τα παρακάτω χαρακτηριστικά:

- Likelihood of Attack: High
- Skills Required: Low Level Skill- Ένας επιτιθέμενος μπορεί απλώς να υπερχειλίσει ένα buffer εισάγοντας μια μεγάλη συμβολοσειρά σε ένα διάνυσμα ένεσης που μπορεί να τροποποιηθεί από τον εισβολέα. Το αποτέλεσμα μπορεί να είναι η άρνηση υπηρεσίας.
- Consequences:
    - Μη εμπιστευσιμη εκτέλεση
    - Εκτέλεση μη εξουσιοδοτημένων εντολών
    - Ανάγνωση Δεδομένων
    - Μετατροπή Δεδομένων
    - Απόκτηση Δικαιωμάτων

Τα παραπάνω χαρακτηριστικά μπορούν στη συνέχεια να συγκριθούν με τις δυνατότητες και το κίνητρο του επιτηθέμενου, για τα προφιλ επιτιθέμενων που έχουν αναγνωριστεί ως σχετικά στο πλαίσιο της υπο μελέτη υποδομής.

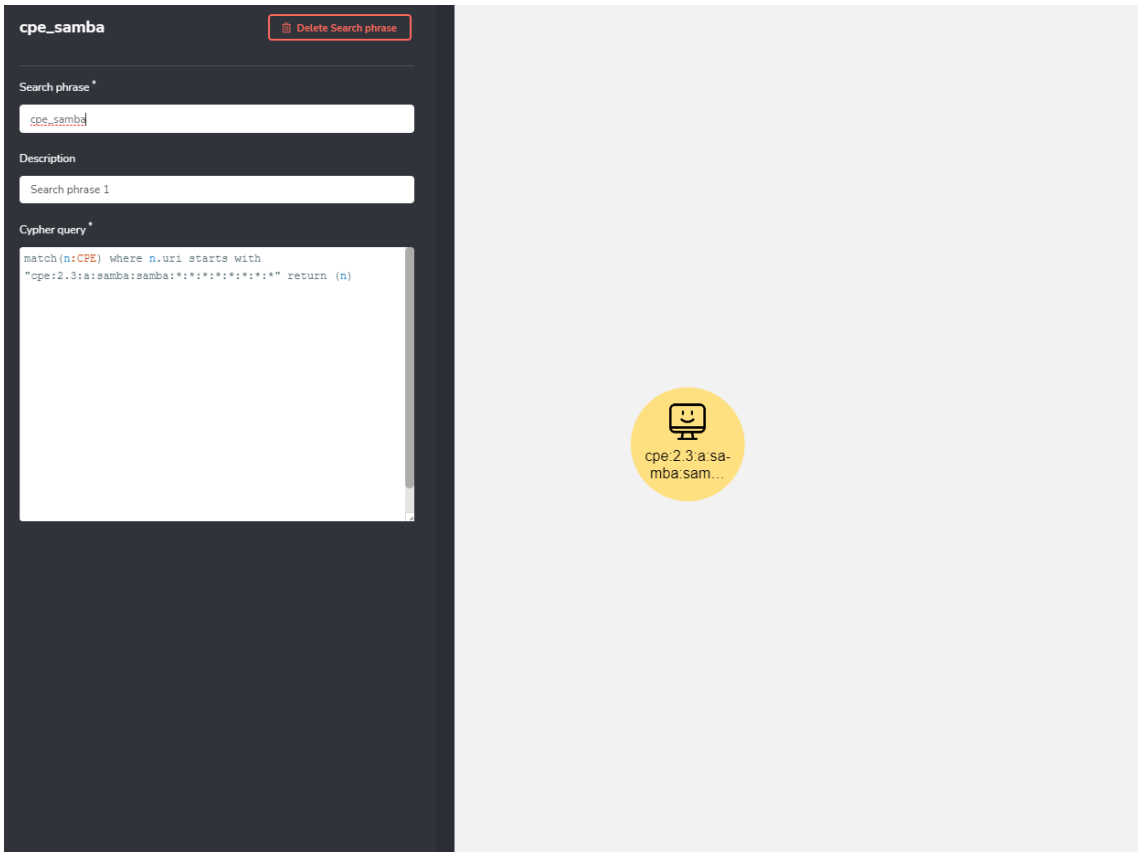
CPE	**cpe:2.3:o:linux:linux_kernel:2.6:*:*:*:*:*:*:***	
CVE	**CWE**	**CAPEC**
CVE-2009-2847	-	-
CVE-2008-3889	CWE-20	"CAPEC-22", "CAPEC-182", "CAPEC-588", "CAPEC-43", "CAPEC-250", "CAPEC-101", "CAPEC-209", "CAPEC-267", "CAPEC-42", "CAPEC-14", "CAPEC-64", "CAPEC-7", "CAPEC-46", "CAPEC-67", "CAPEC-120", "CAPEC-135", "CAPEC-72", "CAPEC-3", "CAPEC-13", "CAPEC-85", "CAPEC-261", "CAPEC-24", "CAPEC-45", "CAPEC-9", "CAPEC-81", "CAPEC-664", "CAPEC-88", "CAPEC-110", "CAPEC-47", "CAPEC-231", "CAPEC-104", "CAPEC-63", "CAPEC-78", "CAPEC-8", "CAPEC-230", "CAPEC-28", "CAPEC-79", "CAPEC-23", "CAPEC-473", "CAPEC-52", "CAPEC-108", "CAPEC-83", "CAPEC-10", "CAPEC-80", "CAPEC-136", "CAPEC-109", "CAPEC-53", "CAPEC-153", "CAPEC-71", "CAPEC-73", "CAPEC-31"
CVE-2008-5395	CWE-119	"CAPEC-46", "CAPEC-24", "CAPEC-100", "CAPEC-123", "CAPEC-8", "CAPEC-14", "CAPEC-9", "CAPEC-45", "CAPEC-44", "CAPEC-10", "CAPEC-42", "CAPEC-47"
CVE-2008-3832	CWE-399	-

CPE 2 -App

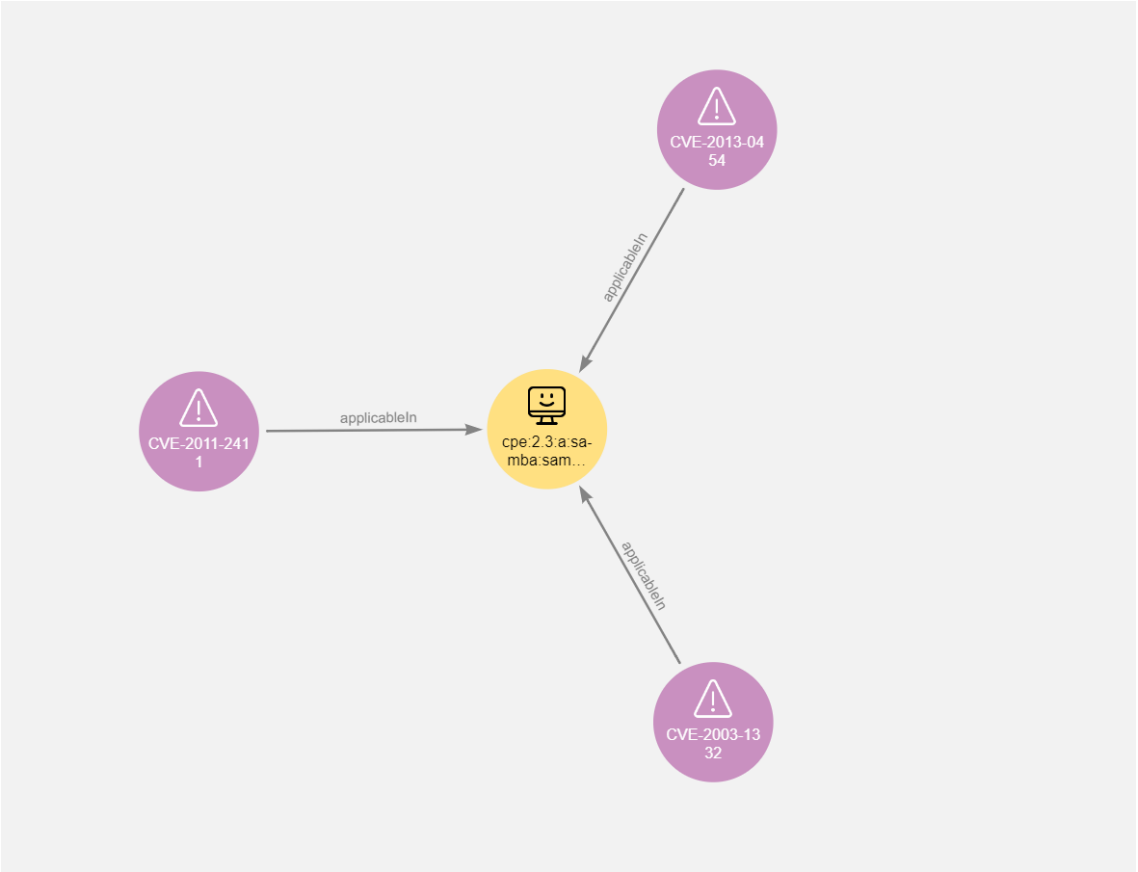
Η πρώτη αναζήτηση που τρέχουμε πάνω στο graphker επιβεβαιώνει πως το CPE uri της πρώτης εφαρμογής υπό εξέταση υπάρχει στη βάση και επιστρέφει τον κόμβο όπως φαίνεται στην Εικόνα 51.

Στη συνέχεια για τον κόμβο που βρέθηκε αναζητούμε τις αριθμημένες ευπάθειες (CVE) όπως φαίνεται στην Εικόνα 52. Πάνω στις ευπάθειες που βρέθηκαν υπάρχει μια καταγεγραμμένη περιγραφή, τα χαρακτηριστικά που παράγουν το CVSS Score της συγκεκριμένης ευπάθειας καθώς και ορισμένες αναφορές σε καταγεγραμμένα exploit, patches και workarounds της ευπάθειας.

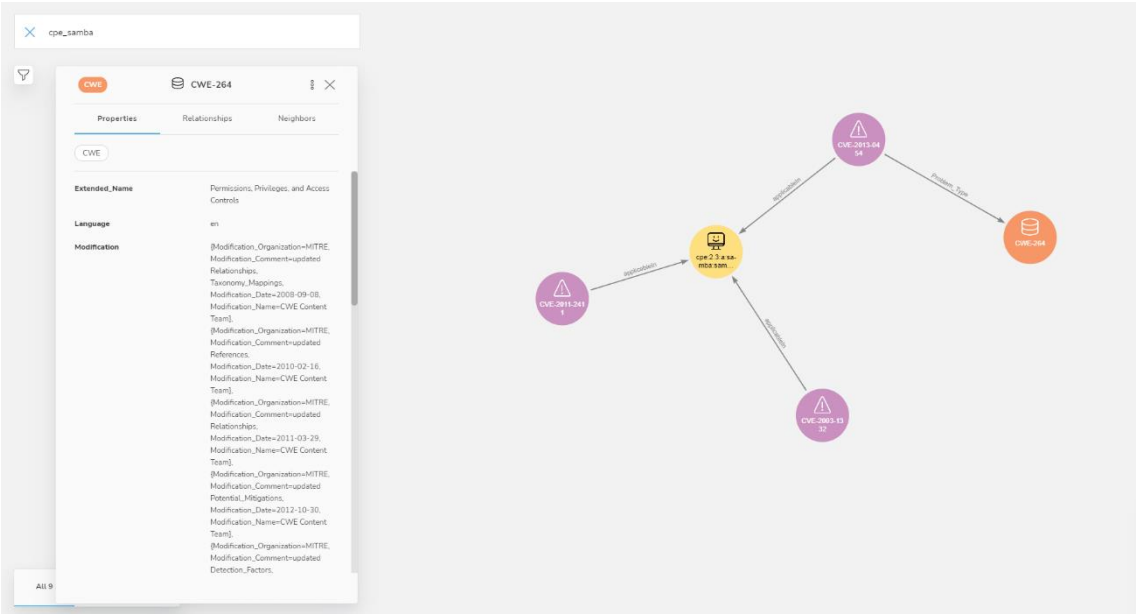
Τέλος πραγματοποιείται μια αναζήτηση πάνω στις αδυναμίες λογισμικού και τα μοτίβα επιθέσεων που συνδέονται με τις ευπάθειες που βρέθηκαν στο προηγούμενο βήμα όπως φαίνεται στην Εικόνα 53.



Εικόνα 51: Application 1 CPE



Εικόνα 52: Application 1 CPE Vulnerabilities



Εικόνα 53: Application 1 CPE Vulnerabilities and Threats

Στον πίνακα παρουσιάζονται με λεπτομέρεια τα αποτελέσματα της αναζήτησης που πραγματοποιήθηκε. Η εφαρμογή παρουσιάζει πολλαπλές ευπάθειες, ενώ παρατηρούμε λίγες συνδέσεις σε υπάρχουσες αδυναμίες και μοτίβα επιθέσεωνσε αντίθεση με το προηγούμενο παράδειγμα. Οι πληροφορίες που μας επέστρεψε η αναζήτηση μπορούν να συνδυαστούν για να παράξουν σενάρια ως εξής:

Το δίκτυο samba παρουσιάζει την ευπάθεια CVE-2013-0454, μέσω της οποίας ένας χρήστης που έχει πρόσβαση μέσω του διαδικτύου στο δίκτυο μπορεί να γράψει σε αρχεία που προορίζονταν μόνο για ανάγνωση. Η ευπάθεια αυτή ανήκει στην κατηγορία αδυναμιών λογισμικού CWE-264 που αναφέρεται αδυναμίες λογισμικού σχετικα με δικαιώματα, προνόμια και στοιχεία ελέγχου πρόσβασης.

Λόγω της έλλειψης πληροφορίας συνδεσιμότητας CVE-CWE-CAPEC σε αυτό το παράδειγμα είναι πιο δύσκολη η διασύνδεση του προφίλ επιτιθέμενου. Μια πιθανή λύση είναι η περαιτέρω επεξεργασία και εξαγωγή σχέσεων μεταξύ των καταγεγραμμένων CVE-CWE-CAPEC.

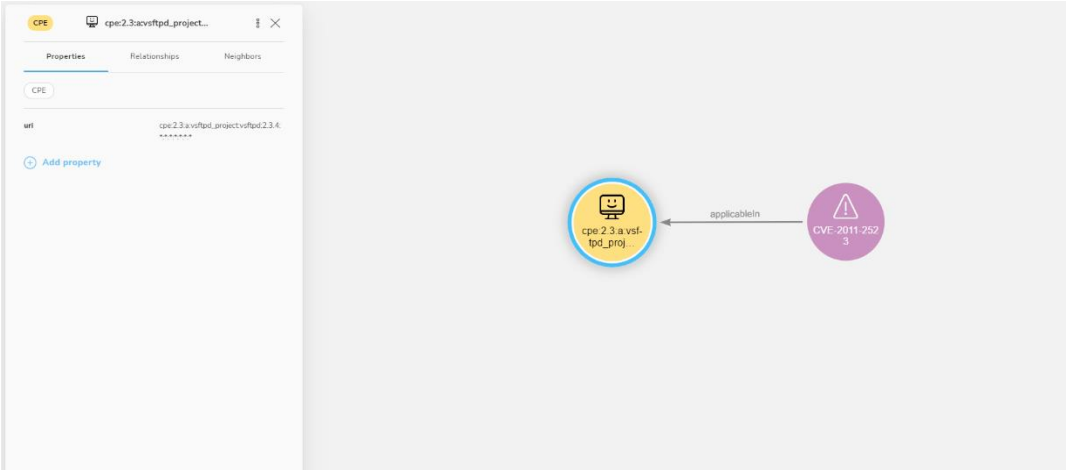
CPE	**cpe:2.3:a:samba:samba:*:*:*:*:*:*:*:***	
CVE	**CWE**	**CAPEC**
CVE-2011-2411	-	
CVE-2003-1332	-	
CVE-2013-0454	CWE-264	-

CPE 3 – Application 2

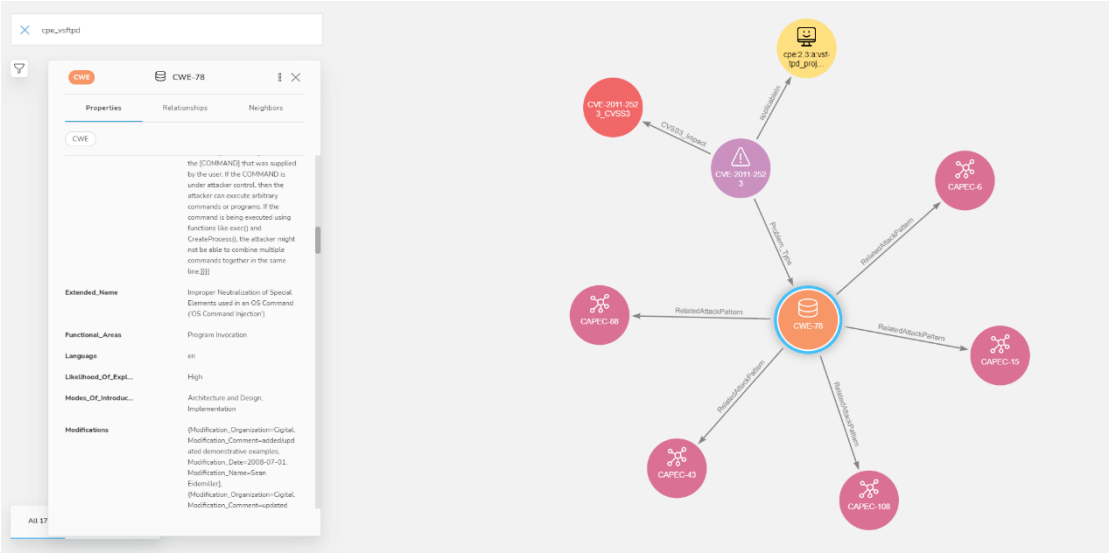
Η πρώτη αναζήτηση που τρέχουμε πάνω στο Graphker επιβεβαιώνει πως το CPE uri της δεύτερης εφαρμογής υπό εξέταση υπάρχει στη βάση και επιστρέφει τον κόμβο μαζί με τις αριθμημένες ευπάθειες (CVE) του όπως φαίνεται στην. Πάνω στις ευπάθειες που βρέθηκαν υπάρχει μια καταγεγραμμένη περιγραφή, τα χαρακτηριστικά που παράγουν το CVSS Score της συγκεκριμένης ευπάθειας καθώς και ορισμένες αναφορές σε καταγεγραμμένα exploit, patches και workarounds της ευπάθειας.

Στη συνέχεια πραγματοποιείται μια αναζήτηση πάνω στις αδυναμίες λογισμικού και τα μοτίβα επιθέσεων που συνδέονται με τις ευπάθειες που βρέθηκαν στο προηγούμενο βήμα όπως φαίνεται στην Εικόνα 55.

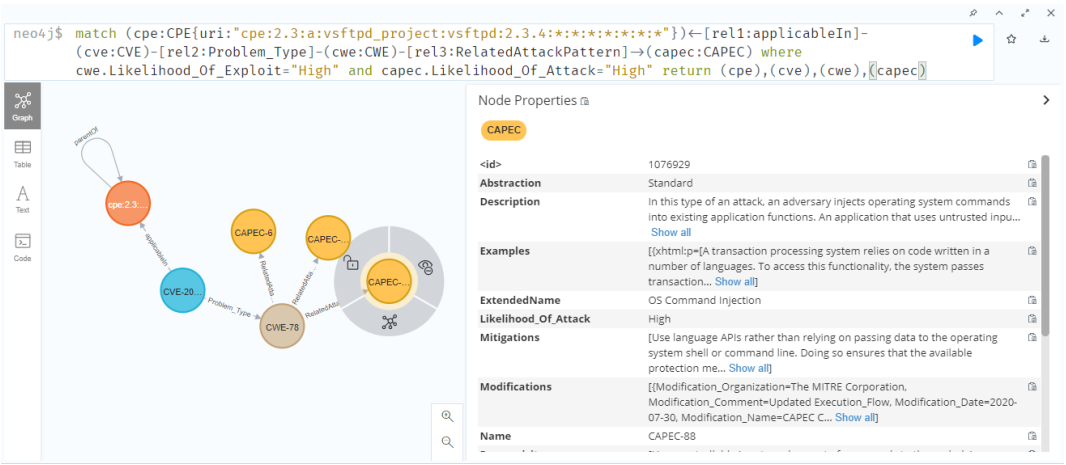
Τέλος κάνουμε περαιτέρω αναζήτηση πάνω στα μοτίβα CAPEC που βρέθηκαν.



Εικόνα 54: Application 2 CPE Vulnerabilities



Εικόνα 55: Application 2 CPE Vulnerabilities and Threats



Εικόνα 56: Application 2 CPE CAPEC information

CPE	**cpe:2.3:a:vsftpd_project:vsftpd:2.3.4:*:*:*:*:*:*:***	
CVE	**CWE**	**CAPEC**
CVE-2011-2523	CWE-78	CAPEC-88", "CAPEC-15", "CAPEC-43", "CAPEC-108", "CAPEC-6

5 Συμπεράσματα

Στο πλαίσιο της παρούσας διπλωματικής μελετήθηκαν πολλαπλές ανοιχτές πηγές που προσφέρουν πληροφορίες για ευπάθειες, αδυναμίες λογισμικού, μοτίβα επιθέσεων και υλοποιήθηκε ένας αποδοτικός τρόπος ανάκτησης τους και η μοντελοποίηση και εισαγωγή τους σε βάση δεδομένων γράφων με τη χρήση του Neo4j. Χρησιμοποιώντας τα ανοιχτά δεδομένα και προηγούμενες μεθοδολογίες διαχείρισης τους ως βάση, δημιουργήθηκε μια αρχιτεκτονική για την αυτόματη λήψη και την αποτελεσματική και οργανωμένη αποθήκευση των παραπάνω δεδομένων σε μια καινούργια βάση δεδομένων γράφων, το Graphker. Η έκταση και η ποικιλομορφία του Graphker προσφέρεται για την περαιτέρω εξαγωγή συνδέσεων και γνώσης από τα ήδη υπάρχοντα ανοιχτά δεδομένα και τα εκτεταμένα χαρακτηριστικά τους. Σε αυτό το πλαίσιο αναλύοντας τα χαρακτηριστικά των threat agent profiles που παρουσιάζονται στο πλαίσιο της ταξονομίας Intel TAL ανακαλύψαμε πολλές κοινές μεταβλητές με τους καταλόγους CWE και CAPEC που δύναται να γεφυρώσουν με έγκυρες διασυνδέσεις τις διαφορετικές οντότητες της οντολογίας μας. Τα κοινά αυτά χαρακτηριστικά μπορούν να θεωρηθούν ως δυνατότητες στην περίπτωση των threat agent και ως απαιτήσεις στην περίπτωση των ευπαθειών και των αδυναμιών, συνεπώς παράγεται ένα μέτρο σύγκρισης μεταξύ των ευπαθειών και των επιτιθέμενων που έχουν τις απαραίτητες δυνατότητες, προμήθειες και κίνητρο για να εκμεταλλευτούν τις υπάρχουσες αδυναμίες. Τέλος η συνδυαστική πληροφορία του γράφου μπορεί να χρησιμοποιηθεί για να παρέχει πληροφορία σε υπάρχουσες Risk Assessment μεθοδολογίες, πιο συγκεκριμένα παρέχοντας στο εργαλείο έναν χάρτη των φυσικών και ψηφιακών αγαθών μιας υποδομής μπορεί να παραχθεί αυτόματα η λίστα των συσχετιζόμενων ευπαθειών και αδυναμιών λογισμικού, μοτίβων επιθέσεων και τα προφίλ επιτιθέμενων που έχουν τα απαραίτητα χαρακτηριστικά για να τα υλοποιήσουν.

Η παρούσα μορφή της οντολογίας, παρότι εκτεταμένη παρουσιάζει διάφορες ελλείψεις και δυνατότητες βελτίωσης. Πιο συγκεκριμένα, παρατηρούμε ότι δεν υπάρχουν απευθείας σχέσεις μεταξύ CVE & CAPEC. Στο πλαίσιο της ΒΔΓ επιλύσαμε αυτό το πρόβλημα χρησιμοποιώντας το CWE ως συνδετικό κόμβο, μιας και παρουσιάζει διασυνδέσεις και με τους δύο καταλόγους. Η προσέγγιση αυτή μας απέφερε διασυνδέσεις για ένα 52% του CVE Dataset. Αυτό το πρόβλημα θα μπορούσε να επιλυθεί με διαφορετική προσέγγιση. Για παράδειγμα, η χρήση NLP για τη διασύνδεση CVE & CAPEC μέσω της εξαγωγής γνώσης μεταξύ των

Here, an approach to directly find common links between these dictionaries is proposed. Then, several patterns, which are combinations of similarity measures and popular algorithms such as term frequency–inverse document frequency, universal sentence encoder, and sentence BERT, are evaluated experimentally using the proposed approach.

Θα μπορούσαμε να κάνουμε περαιτέρω διερεύνηση σε αυτό το θέμα, χρησιμοποιώντας διαφορετικούς αλγορίθμους και ελέγχοντας τα αποτελέσματα στα ίδια ή και σε πιο εκτεταμένα δεδομένα.

Επίσης, έχουν υπάρξει προσεγγίσεις οι οποίες υποβοηθούν τον αυτόματο υπολογισμό των χρονικών μετρικών που αφορούν τις ευπάθειες λογισμικού. Αυτό το διάνυσμα μεταβλητών δεν υπάρχει καταγεγραμμένο στο CVE Database του NIST αλλά αρκετές πληροφορίες που είναι απαραίτητες για τη διαμόρφωσή του, μπορούν να βρεθούν στο πεδίο αναφορών των CVE καθώς και σε άλλα ανοιχτά μέσα όπως το Twitter, το Reddit και διάφορα Cyber Security Blogs. Αντίστοιχα και σε αυτήν την περίπτωση, η χρήση NLP αλγορίθμων θα μπορούσε να βοηθήσει στη διαμόρφωση μιας έγκυρης μεθοδολογίας για τον αυτόματο υπολογισμό αυτού του διανύσματος από ανοιχτά δεδομένα.

Οι παραπάνω είναι μόνο λίγες από τις πιθανές επεκτάσεις που θα μπορούσαν να εφαρμοστούν στο πλαίσιο της μεθοδολογίας μας με στόχο την αποδοτική και αυτοματοποιημένη ανάλυση και διαχείριση επικινδυνότητας ψηφιακών και φυσικών πληροφοριακών συστημάτων και υποδομών, μέσω της ΒΔΓ.

6 Βιβλιογραφικές Πηγές

[1] *National Vulnerability Database,* NIST - National Institute of Standards and Technology.

[2] *Common Platform Enumeration,* NIST - National Institute of Standards and Technology.

[3] *Common Weakness Enumeration,* MITRE Corporation.

[4] *Common Attack Pattern Enumeration and Classification,* MITRE Corporation.

[5] MITRE, "ATT&CK - MITRE," [Online]. Available: https://attack.mitre.org/.

[6] *Common Vulnerabilities and Exposures,* MITRE Corporation.

[7] I. Robinson, J. Webber and E. Eifrem, Graph databases: new opportunities for connected data, " O'Reilly Media, Inc.", 2015.

[8] A. Hodler and M. Needham, Graph data science for dummies, John Wiley & Sons, Inc, 2021.

[9] J. Webber and R. V. Bruggen, Graph databases for dummies, John Wiley & Sons, Inc., 2020.

[10] *Common Vulnerability Scoring System,* NIST - National Institute of Standards and Technology.

[11] J. a. R. S. a. E. B. a. R. M. a. A. I. Jacobs, "Exploit prediction scoring system (EPSS)," *arXiv preprint arXiv:1908.04856,* 2019.

[12] D. Ivan, "Graph Database vs Relational Database," 2021.

[13] S. Johan, "Guest View: Relational vs. graph databases: Which to use and when?," 2016.

[14] R. Ramakrishnan, J. Gehrke and J. Gehrke, Database management systems, McGraw-Hill New York, 2003.

[15] A. Vukotic, N. Watt, T. Abedrabbo, D. Fox and J. Partner, Neo4j in action, Manning Shelter Island, 2015.

[16] J. Webber, *Top 10 use cases: Knowledge graphs,* Neo4j Graph Data Platform, 2021.

[17] N. A. Christakis and J. H. Fowler, Connected: The surprising power of our social networks and how they shape our lives, Little, Brown Spark, 2009.

[18] W. Lyon, *The Story behind Russian Twitter Trolls: How They Got Away with Looking Human – and How to Catch Them in the Future,* Neo4j Graph Data Platform.

[19] *Personalized Product Recommendations with Neo4j,* Neo4j Graph Data Platform.

[20] D. Williams, *Mapping a Connected World: The Value of Geospatial Graph Visualization,* Neo4j Graph Data Platform.

[21] B. M. Sasaki, *Graph Databases for Beginners: The Basics of Data Modeling,* Neo4j Graph Data Platform.

[22] *Graph Databases for Entitlements and Access Control,* Neo4j Graph Data Platform.

[23] J. Zagalsky, *Graphs in Government - Fulfilling Your Mission with Neo4j,* Neo4j Graph Data Platform, 2021.

[24] G. Sadowski and P. Rathle, "Fraud detection: Discovering connections with graph databases," *White Paper-Neo Technology-Graphs are Everywhere,* 2014.

[25] J. Webber, *Top 10 Graph Database Use Cases: Fraud Detection,* Neo4j Graph Data Platform.

[26] J. Webber, *Top 10 Use Cases: Anti-Money Laundering,* Neo4j Graph Data Platform.

[27] A. Negro, Graph-powered machine learning, Simon and Schuster, 2021.

[28] B. D. Williams, "Graph databases find new applications in cybersecurity boom," 2017.

[29] L. Blog, "Cyber security : how to use graphs to do an attack analysis," 2021.

[30] L. Blog, "Graph data visualisation for cyber-security threats analysis," 2022.

[31] S. Noel, E. Harley, K. H. Tam, M. Limiero and M. Share, "CyGraph: graph-based analytics and visualization for cybersecurity," *Handbook of Statistics,* 2016.

[32] S. Noel, *Building a Big Data Architecture for Cyber Attack Graphs,* Neo4j Graph Data Platform.

[33] N. Mathur, *Overcoming CCPA compliance challenges - Neo4j,* Neo4j Graph Data Platform, 2021.

[34] *How to turn GDPR into a Strategic Advantage using Connected Data,* Neo4j Graph Data Platform.

[35] N. Mathur, *BCBS 239 compliance & Graph Technology - go.neo4j.com,* Neo4j Graph Data Platform, 2021.

[36] N. Mathur, *GDPR Compliance: The Challenges and Problems with Personal Data,* Neo4j Graph Data Platform.

[37] P. a. L. X. a. C. E. a. S. B. a. M. C. a. F. K. a. S. D. Gao, "A system for automated open-source threat intelligence gathering and management," in *Proceedings of the 2021 International Conference on Management of Data,* 2021, pp. 2716--2720.

[38] E. a. K. J. a. S.-R. M. a. R. B. a. X. K. a. R. N. a. O. U.-M. Hemberg, "Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting," *arXiv preprint arXiv:2010.00533,* 2020.

[39] A. S. F. M. a. T. D. Oleksii Osliak, "Towards Collaborative Cyber Threat Intelligence for Security Management," *ICISSP,* pp. 339-346, 2021.

[40] P. a. M. J. A. a. S. A. F. Gonzalez-Gil, "Lightweight data-security ontology for IoT," *Sensors,* p. 801, 2020.

[41] Y. a. Q. Y. a. S. H. a. J. R. a. L. A. Jia, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering - Elsevier,* pp. 53--60, 2018.

[42] D. a. R. G. Velasco, "Ontologies for Network Security and Future Challenges," *arXiv preprint arXiv:1704.02441,* 2017.

[43] A. a. S. S. a. o. Singhal, "Security ontologies for modeling enterprise level risk assessment," in *Proceedings of the 2012 Annual Computer Security Applications Conference, Orlando, FL, USA*, 2012, pp. 3--7.

[44] V. G. Nair, Getting started with beautiful soup, Packt Publishing Ltd, 2014.

[45] R. Mitchell, Web scraping with Python: Collecting more data from the modern web, " O'Reilly Media, Inc.", 2018.

[46] *Java Reference Neo4j v4.4,* Neo4j Graph Data Platform.

[47] *The Neo4j Operations Manual v4.4,* Neo4j Graph Data Platform.

[48] M. Needham and A. E. Hodler, Graph algorithms: practical examples in Apache Spark and Neo4j, O'Reilly Media, 2019.

[49] D. Allen, "How Queries Work in Neo4j," 2020.

[50] D. Fernandes and J. Bernardino, "Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB.," 2018.

[51] M. H. Boza and A. Muñoz, "(In) Security in Graph Databases-Analysis and Data Leaks.," 2017.

[52] *The Neo4j Cypher Manual v4.4,* Neo4j Graph Data Platform.

[53] *Status Codes Neo4j v4.4,* Neo4j Graph Data Platform.

[54] *Neo4j Bloom 2.0,* Neo4j Graph Data Platform.

[55] *Graphlytic - Graph Visualization and Analytics Software,* Demtec.

[56] C. Grigoriadis, A. M. Berzovitis, I. Stellios and P. Kotzanikolaou, "A Cybersecurity Ontology to Support Risk Information Gathering in Cyber-Physical Systems," 2022.

[57] T. Casey, "Threat agent library helps identify information security risks," *Intel White Paper,* 2007.

[58] I. a. K. P. a. G. C. Stellios, "Assessing IoT enabled cyber-physical attack paths against critical systems," *Computers \& Security - Elsevier,* p. 102316, 2021.

[59] A. M. Berzovitis, *GraphKer - Open Source Tool - Cybersecurity Graph Database in Neo4j,* Github.

[60] *Cypher Query Formatter,* Tristan Perry.

[61] *The Neo4j Python Driver Manual v4.4,* Neo4j Graph Data Platform.

7 Οδηγίες Χρήσεις GraphKer, Πίνακες και εξτρά υλικό

7.1 Manual

Στην ενότητα αυτή παρουσιάζεται το Manual του εργαλείου, ακριβώς όπως υπάρχει στο επίσημο GitHub Repository amberzovitis/GraphKer: Open Source Tool - Cybersecurity Graph Database in Neo4j (github.com).

GraphKer

Open-Source Tool - Cybersecurity Graph Database in Neo4j

|G|r|a|p|h|K|e|r|

{ open-source tool for a cybersecurity graph database in neo4j }

With GraphKer you can have the most recent update of cyber-security vulnerabilities, weaknesses, attack patterns and platforms from MITRE and NIST, in a very useful and user-friendly way provided by Neo4j graph databases!

Prerequisites

3 + 1 Steps to run GraphKer Tool

1) **Download and Install Neo4j Desktop**
   a. **Windows Users:**
   https://neo4j.com/download/ Create an account to get the license (totally free), download and install Neo4j Desktop. Useful Video: https://tinyurl.com/yjjbn8jx
   b. **Linux Users:**

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl
software-properties-common
curl -fsSL https://debian.neo4j.com/neotechnology.gpg.key | sudo
apt-key add -
sudo add-apt-repository "deb https://debian.neo4j.com stable
4.1"
sudo apt install neo4j
sudo systemctl enable neo4j.service
sudo systemctl status neo4j.service
```

You should have output that is like the following:

```
● neo4j.service - Neo4j Graph Database
Loaded: loaded (/lib/systemd/system/neo4j.service; enabled;
vendor preset: enabled)
Active: active (running) since Fri 2020-08-07 01:43:00 UTC; 6min
ago
Main PID: 21915 (java)
Tasks: 45 (limit: 1137)
Memory: 259.3M
```

```
CGroup: /system.slice/neo4j.service
. . .
```

Useful Video: https://tinyurl.com/vvpjf3dr

2) Create and Configure the Database

a. Create Database:

i. Windows Users:
You can create databases in whatever version you want (latest version preferable) through GUI or Neo4j Terminal.

- Create a new database in GUI: Just click the (+), set DB Name, Username and Password. Useful Tutorial: https://www.sqlshack.com/getting-started-with-the-neo4j-graph-database/
- Through Neo4j Shell: https://neo4j.com/docs/cypher-manual/current/databases/

ii. Linux Users:
When you start neo4j through systemctl, type `cypher-shell`, then `create database NAME;`. Now you must set this database, as default so when you start neo4j you start automatically this database. Go to `/etc/neo4j/neo4j.conf` and uncomment `dbms.default_database=neo4j` and change it with your new database name. Restart neo4j service and you are ready.

b. Configure Database:

i. Install APOC Plugin:

1. Windows Users:
In Neo4j Desktop Main Page --> Choose your Database --> Click Plugins --> APOC --> Install
2. Linux Users:
   a. Download APOC jar File: https://github.com/neo4j-contrib/neo4j-apoc-procedures/releases (--all.jar file)
   b. Place it in Plugins Folder --> check every folder path in Neo4j: https://neo4j.com/docs/operations-manual/current/configuration/file-locations/
   c. Modify the Database Configuration File to approve apoc procedures. Uncomment:
   `dbms.directories.plugins=plugins`
   Uncomment and Modify:
   ```
   dbms.security.procedures.unrestricted=apoc.*
   dbms.security.procedures.whitelist=apoc.*,apoc.coll.*,apoc.load.*
   #Loads unrestricted and white-listed procedures/plugins to the server
   ```
   Restart Neo4j: `systemctl restart neo4j`

ii. Configure Database Settings File:

1. Windows Users:

In Neo4j Desktop Main Page --> Choose your Database --> ... (Three Dots) --> Settings --> Go to last line and set the commands below --> Apply and Restart the Database

```
apoc.export.file.enabled=true
apoc.import.file.enabled=true
apoc.import.file.user_neo4j_config=false
cypher.lenient_create_relationship = true
```

2. Linux Users:
   Same as above, in the neo4j.conf file --> check every folder path in Neo4j: https://neo4j.com/docs/operations-manual/current/configuration/file-locations/

iii. Configure Memory Usage:
In Neo4j Configuration File (neo4j.conf): For 16GB RAM you can use 8G + 4G for heap. For 8GB RAM you can use 4G + 2G etc.

```
dbms.memory.heap.initial_size=4G
dbms.memory.heap.max_size=8G
dbms.memory.pagecache.size=4G
```

3) **Install requirements.txt**
GraphKer Uses: xmltodict, neo4j, requests, beautifulsoup4

```
pip install -r requirements.txt
```

4) **Install Applications Created for Neo4j**
There are several applications created especially for Neo4j that you can use for better experience and work.

Neo4j Bloom: Application for better graph presentations. Free and Easy to use.
Graphlytic: Third-Party App, better graph presentations, but most important auto-analytics and statistics. Free and Paid Editions. We can do the most locally with free edition. Learn More: https://graphlytic.biz/

Neo4j Database Analyzer: Third-Party App, Free, provides great analysis tools for our Data and our Schema. Learn More: https://community.neo4j.com/t/introducing-the-neo4j-database-analyzer/6197

Run GraphKer

```
// Default
python main.py -u BOLT_URL -n USERNAME -p PASSWORD -d IMPORT_PATH
// Run and Open Neo4j Browser
python main.py -u BOLT_URL -n USERNAME -p PASSWORD -d IMPORT_PATH -b y
// Run and Open Graphlytic App
python main.py -u BOLT_URL -n USERNAME -p PASSWORD -d IMPORT_PATH -g y
// Default Run Example in Ubuntu
```

```
sudo python3 main.py -u BOLT_URL -n USERNAME -p PASSWORD -d
/var/lib/neo4j/import/
```

Default Bolt URL for Neo4j: bolt://localhost:7687

Default Username in Neo4j Databases: neo4j

For Neo4j Import Folder check the link above with File Locations.

Estimated RunTime: **6-15 Minutes**. Depends on hardware.

At least 8GB in your hard drive.

7.2 Tables

Μοντελοποίηση – Οντολογία

Στην ενότητα αυτή θα παρουσιάσουμε τον τρόπο μοντελοποίησης των συλλογών στη βάση δεδομένων του Neo4j. Η μοντελοποίηση γίνεται με τη χρήση της γλώσσας Cypher ενώ στον παρακάτω πίνακα παρουσιάζονται όλες οι λεπτομέρειες για κάθε οντότητα που δημιουργείται. Στον πρώτο πίνακα καταγράφονται οι κόμβοι, με τα αναγνωριστικά, τα χαρακτηριστικά τους και συγκεκριμένες μετρικές γι' αυτά, ενώ στον δεύτερο πίνακα καταγράφονται οι σχέσεις μεταξύ των οντοτήτων (και οι σχετικές λεπτομέρειες για κάθε σχέση). Για κάθε χαρακτηριστικό (αναγνωριστικό ή μη) εμπεριέχεται σύντομη επεξήγηση, τύπος της μεταβλητής, εύρος τιμών που μπορεί να πάρει καθώς και αν είναι υποχρεωτικό για την ύπαρξη του κόμβου ή της σχέσης όπου υπόκεινται.

	Πίνακας Κόμβοι						
#	**Ετικέτα**	**Περιγραφή**	**Αναγνωριστικό**	**Χαρακτηριστικά**	**Τύπος**	**Πλήθος**	**Required**
1	CPE	(Βλ. «Χρήσιμοι Ορισμοί»)					
		(Βλ. «Συλλογές Δεδομένων – Datasets»)	uri		string	1	ΝΑΙ
2	CVE	(Βλ. «Χρήσιμοι Ορισμοί»)					
		Το ID του CVE	Name		string	1	ΝΑΙ
		Ποιος το εκδίδει		Assigner	string	[0,1]	ΟΧΙ
		Περιγραφή (μόνο σε eng)		Description	string array	[1,N]	ΝΑΙ
		Ημ/νία Έκδοσης		Published Date	string	[0,1]	ΟΧΙ
		Ημ/νία Τελευταίας Τροποποίησης		Last Modified Date	string	[0,1]	ΟΧΙ
3	General Info CVE	Γενικές Πληροφορίες του αρχείου του Dataset					
		Τύπος Δεδομένων	Data Type		string	1	ΝΑΙ

		Μορφή Καταγραφής	Data Format		string	1	NAI
		Έκδοση	Data Version		string	1	NAI
		Αριθμός Ευπαθειών στο Dataset	No CVEs		string	[0,1]	OXI
		Χρονική Στιγμή Έκδοσης του Dataset	Timestamp		string	[0,1]	OXI
4	CVSS_2	Μέτρηση Score CVSS (Βλ. «Χρήσιμοι Ορισμοί») Έκδοσης 2					
		CVSS_2 ID	Name		string	1	NAI
		Η έκδοση του CVSS		Version	string	1	NAI
		Σε μια γραμμή τα αποτελέσματα των βασικών μετρικών		Vector String	string	1	NAI
		Τρόπος εκμετάλλευσης της ευπάθειας		Access Vector	string	[0,1]	OXI
		Πολυπλοκότητα εκμετάλλευσης της ευπάθειας		Access Complexity	string	[0,1]	OXI
		Πόσες φορές πρέπει να αυθεντικοποιηθεί ο επιτιθέμενος για να εκμεταλλευτεί με επιτυχία την ευπάθεια		Authentication		[0,1]	OXI
		Επίπτωση στην Εμπιστευτικότητα		Confidentiality Impact	string	[0,1]	OXI
		Επίπτωση στην Ακεραιότητα		Integrity Impact	string	[0,1]	OXI
		Επίπτωση στη Διαθεσιμότητα		Availability Impact	string	[0,1]	OXI
		Αποτέλεσμα συνάρτησης βάσει των παραπάνω μετρικών		Base Score	number	1	NAI
		Αποτέλεσμα συνάρτησης βάσει παραμέτρων για την επιτυχή εκμετάλλευση της ευπάθειας		Exploitability Score	number	[0,1]	OXI
				Severity	string	[0,1]	OXI
		Αποτέλεσμα βάσει παραμέτρων για την επίπτωση της επιτυχούς		Impact Score	number	[0,1]	OXI

		εκμετάλλευσης της ευπάθειας					
				acInsuInfo	Boolean	[0,1]	OXI
		Δυνατότητα απόκτησης των προνομίων όλων των ομάδων χρηστών		Obtain All Privileges	Boolean	[0,1]	OXI
		Δυνατότητα απόκτησης των προνομίων των απλών χρηστών		Obtain User Privileges	Boolean	[0,1]	OXI
		Δυνατότητα απόκτησης των προνομίων άλλων ομάδων χρηστών		Obtain Other Privileges	Boolean	[0,1]	OXI
		Χρειάζεται ή όχι η αλληλοεπίδραση με χρήστη για την εκμετάλλευση της ευπάθειας		User Interaction Required	boolean	[0,1]	OXI
5	CVSS_3	Μέτρηση Score CVSS (Βλ. «Χρήσιμοι Ορισμοί») Έκδοσης 3					
		CVSS_3 ID	Name		string	1	NAI
		Η έκδοση του CVSS		Version	string	1	NAI
		Σε μια γραμμή τα αποτελέσματα των βασικών μετρικών		Vector String	string	1	NAI
		Τρόπος - Μέσο εκμετάλλευσης της ευπάθειας		Attack Vector	string	[0,1]	OXI
		Πολυπλοκότητα εκμετάλλευσης της ευπάθειας		Attack Complexity	string	[0,1]	OXI
		Προνόμια ομάδων χρηστών που απαιτούνται για την επιτυχή εκμετάλλευση της ευπάθειας		Privileges	string	[0,1]	OXI
		Χρειάζεται ή όχι η αλληλοεπίδραση με χρήστη για την εκμετάλλευση της ευπάθειας		User Interaction	string	[0,1]	OXI
				Scope	string	[0,1]	OXI
		Επίπτωση στην Εμπιστευτικότητα		Confidentiality Impact	string	[0,1]	OXI

		Επίπτωση στην Ακεραιότητα		Integrity Impact	string	[0,1]	OXI
		Επίπτωση στην Διαθεσιμότητα		Availability Impact	string	[0,1]	OXI
		Αποτέλεσμα συνάρτησης βάσει των παραπάνω μετρικών		Base Score	number	1	NAI
				Base Severity	string	1	NAI
		Αποτέλεσμα συνάρτησης βάσει παραμέτρων για την επιτυχή εκμετάλλευση της ευπάθειας		Exploitability Score	number	[0,1]	OXI
		Αποτέλεσμα βάσει παραμέτρων για την επίπτωση της επιτυχούς εκμετάλλευσης της ευπάθειας		Impact Score	number	[0,1]	OXI
6	Reference_Data	Δημόσιες Αναφορές του CVE					
		Δημόσια Αναφορά του CVE στο Διαδίκτυο	url		string	1	NAI
		Το Όνομα της Δημόσιας Αναφοράς		Name	string	[0,1]	OXI
		Φορέας/Φυσικό Πρόσωπο που την Εξέδωσε		refSource	string	[0,1]	OXI
7	General_Info_CWE	Γενικές Πληροφορίες του αρχείου του Dataset					
		Το όνομα του καταλόγου	Name		string	1	NAI
		Έκδοση του καταλόγου	Version		string	1	NAI
		Ημερομηνία δημοσίευσης	Date		string	1	NAI
		Σύνδεσμος της οντολογίας του καταλόγου	Schema		string	[0,1]	OXI
8	CWE	(Βλ. «Χρήσιμοι Ορισμοί»)					
		Το αναγνωριστικό ID της ευπάθειας	Name		String	1	NAI
		Το όνομα της ευπάθειας		Extended Name	String	1	NAI

		Μετρική για το επίπεδο γενικότητας της ευπάθειας		Abstraction	String	1	NAI
		Δομή της ευπάθειας		Structure	String	1	NAI
		Κατάσταση της ευπάθειας		Status	String	1	NAI
		Σύντομη περιγραφή		Description	String	1	NAI
		Εκτενής Περιγραφή		Extended Description	String	[0,1]	OXI
		Πιθανότητα επιτυχούς εκμετάλλευσης της ευπάθειας		Likelihood Of Exploit	String	[0,1]	OXI
		Επιμέρους πληροφορίες		Background Details	String	[0,1]	OXI
		Πηγές της ευπάθειας		Modes Of Introduction	String Array	[0,N]	OXI
		Ημ/νία Έκδοσης		Submission Date	String	[0,1]	OXI
		Όνομα Έκδοσης		Submission Name	String	1	NAI
		Φορέας Έκδοσης		Submission Organization	String	1	NAI
		Τροποποιήσεις		Modifications	String Array	[0,N]	OXI
		Διαφορετικοί όροι έκφρασης της ευπάθειας		Alternate Terms	String	[0,1]	OXI
		Σημειώσεις		Notes	String Array	[0,N]	OXI
		Πόροι που θα επηρεαστούν από την επιτυχή εκμετάλλευση της ευπάθειας		Affected Resources	String Array	[0,N]	OXI
		Πιθανά σημεία στο πληροφοριακό σύστημα όπου μπορεί να εμφανιστεί η ευπάθεια		Functional Areas	String Array	[0,N]	OXI
9	Applicable_Platform	Που εντοπίζονται συνήθως τέτοιες ευπάθειες					
		Ο τύπος της πλατφόρμας που εντοπίζονται	Type		String	1	NAI
		Πόσο συχνά εμφανίζονται στον συγκεκριμένο τύπο	Prevalence		String	1	NAI

		Συγκεκριμένο όνομα πλατφόρμας που εμπίπτει η ευπάθεια		Name	String	[0,1]	OXI
		Εξειδικευμένη κατηγοριοποίηση στον εντοπισμό των πλατφόρμων που εμφανίζεται η ευπάθεια, διαφορετικές κατηγορίες ανά τύπο		Class	String	[0,1]	OXI
10	Demonstrative_Example	Παράδειγμα εκμετάλλευσης της ευπάθειας					
		Περιγραφή του παραδείγματος	Intro Text		String	1	NAI
11	Consequence	Συνέπειες της ευπάθειας					
		Πεδία του Πληρ. Συστήματος τα οποία επηρεάζει η ευπάθεια	Scope		String Array	[1,N]	NAI
12	Detection_Method	Τρόποι Ανίχνευσης της ευπάθειας					
		Μέθοδος ανίχνευσης	Method		String	1	NAI
13	Mitigation	Τρόποι μετρίασης της ευπάθειας					
		Περιγραφή	Description		String	1	NAI
		Φάση του κύκλου ζωής του επηρεαζόμενου προϊόντος		Phase	String Array	[0,N]	OXI
		Στρατηγικές χρήσης του τρόπου μετρίασης της ευπάθειας		Strategy	String	[0,1]	OXI
		Αποτελεσματικότητα του τρόπου απέναντι στην ευπάθεια		Effectiveness	String	[0,1]	OXI
		Σημειώσεις για την αποτελεσματικότητα		Effectiveness Notes	String	[0,1]	OXI
		Αναγνωριστικό της μετρίασης		Mitigation ID	String	1	NAI

14	External_Reference_CWE	Δημόσιες αναφορές του CWE					
		Αναγνωριστικό της αναφοράς	Reference ID		String	1	ΝΑΙ
		Συγγραφέας		Author	String Array	[0,N]	ΟΧΙ
		Τίτλος αναφοράς		Title	String	1	ΝΑΙ
		Έκδοση		Edition	String	[0,1]	ΟΧΙ
		Σύνδεσμος αναφοράς		URL	String	[0,1]	ΟΧΙ
		Έτος έκδοσης αναφοράς		Publication Year	String	[0,1]	ΟΧΙ
		Φορέας αναφοράς		Publisher	String	[0,1]	ΟΧΙ
15	CWE	Κατηγορίες CWE					
		Το αναγνωριστικό ID της κατηγορίας ευπάθειας	Name		String	1	ΝΑΙ
		Το όνομα της κατηγορίας ευπάθειας		Extended Name	String	1	ΝΑΙ
		Κατάσταση της κατηγορίας ευπάθειας		Status	String	1	ΝΑΙ
		Σύντομη Περιγραφή		Summary	String	1	ΝΑΙ
		Σημειώσεις		Notes	String	[0,1]	ΟΧΙ
		Όνομα έκδοσης		Submission Name	String	[0,1]	ΟΧΙ
		Ημερομηνία Έκδοσης		Submission Date	String	[0,1]	ΟΧΙ
		Φορέας Έκδοσης		Submission Organization	String	[0,1]	ΟΧΙ
		Τροποποιήσεις		Modification	String Array	[0,N]	ΟΧΙ
16	CWE_VIEW	Διαφορετική κατηγοριοποίηση των CWE					
		Αναγνωριστικό ID	ViewID		String	1	ΝΑΙ
		Χαρακτηριστικό όνομα		Name	String	1	ΝΑΙ
		Κατηγοριοποίηση σχετικά με την σχέση των μελών του view μεταξύ τους		Type	String	1	ΝΑΙ
		Κατάσταση του view		Status	String	1	ΝΑΙ
		Στόχος κατηγοριοποίησης		Objective	String	1	ΝΑΙ

		Για συγκεκριμένα Types επιπλέον κατηγοριοποίηση		Filter	String	[0,1]	OXI
		Σημειώσεις		Notes	String	[0,1]	OXI
		Όνομα έκδοσης		Submission Name	String	[0,1]	OXI
		Ημερομηνία Έκδοσης		Submission Date	String	[0,1]	OXI
		Φορέας Έκδοσης		Submission Organization	String	[0,1]	OXI
		Τροποποιήσεις		Modification	String Array	[0,N]	OXI
17	Stakeholder	Κοινό που επηρεάζεται από τις κατηγορίες CWE					
		Τύπος του κοινού που επηρεάζεται από τις ευπάθειες που ανήκουν στο view	Type		String	1	NAI
18	GeneralInfo_CAPEC	Γενικές Πληροφορίες του αρχείου του Dataset					
		Το όνομα του καταλόγου	Name		String	1	NAI
		Έκδοση του καταλόγου	Version		String	1	NAI
		Ημερομηνία δημοσίευσης	Date		String	1	NAI
		Σύνδεσμος της οντολογίας του καταλόγου	Schema		String	[0,1]	OXI
19	CAPEC	(Βλ. «Χρήσιμοι Ορισμοί»)					
		Αναγνωριστικό ID	Name		String	1	NAI
		Όνομα του CAPEC		Extended Name	String	1	NAI
		Μετρική για το επίπεδο γενικότητας του CAPEC		Abstraction	String	1	NAI
		Κατάσταση του CAPEC		Status	String	1	NAI
		Περιγραφή		Description	String	1	NAI
		Πιθανότητα επίτευξης της επίθεσης		Likelihood of Attack	String	[0,1]	OXI
				Typical Severity	String	[0,1]	OXI

		Εναλλακτικοί όροι έκφρασης του CAPEC		Alternate Terms	String Array	[0,N]	OXI
		Προϋποθέσεις επίτευξης της επίθεσης		Prerequisites	String Array	[0,N]	OXI
		Τι ικανότητες χρειάζονται για την επίτευξη της επίθεσης		Skills Required	String Array	[0,N]	OXI
		Περιγραφή ικανοτήτων		Skills Required Description	String Array	[0,N]	OXI
		Τρόποι μετρίασης της επίθεσης		Mitigations	String Array	[0,N]	OXI
		Παραδείγματα χρήσης του CAPEC		Examples	String Array	[0,N]	OXI
		Σημειώσεις		Notes	String Array	[0,N]	OXI
		Όνομα έκδοσης		Submission Name	String	[0,1]	OXI
		Ημερομηνία Έκδοσης		Submission Date	String	[0,1]	OXI
		Φορέας Έκδοσης		Submission Organization	String	[0,1]	OXI
		Τροποποιήσεις		Modifications	String Array	[0,N]	OXI
		Πόροι που απαιτούνται για την επίτευξη της επίθεσης		Resources Required	String Array	[0,N]	OXI
		Στοιχεία που καταδεικνύουν ότι γίνεται ή έγινε επίθεση με αυτόν τον τρόπο		Indicators	String Array	[0,N]	OXI
20	External_Reference_CAPEC	Δημόσιες αναφορές του CAPEC					
		Αναγνωριστικό της αναφοράς	Reference ID		String	1	NAI
		Συγγραφέας		Author	String Array	[0,N]	OXI
		Τίτλος αναφοράς		Title	String	1	NAI
		Έκδοση		Edition	String	[0,1]	OXI
		Σύνδεσμος αναφοράς		URL	String	[0,1]	OXI
		Έτος έκδοσης αναφοράς		Publication Year	String	[0,1]	OXI
		Φορέας αναφοράς		Publisher	String	[0,1]	OXI
21	CAPEC	Κατηγορίες CAPEC					

		Αναγνωριστικό ID	Name		String	1	NAI
		Όνομα του CAPEC		Extended Name	String	1	NAI
		Κατάσταση του CAPEC		Status	String	1	NAI
		Σύντομη περιγραφή		Summary	String	1	NAI
		Σημειώσεις		Notes	String Array	[0,N]	OXI
		Όνομα έκδοσης		Submission Name	String	[0,1]	OXI
		Ημερομηνία Έκδοσης		Submission Date	String	[0,1]	OXI
		Φορέας Έκδοσης		Submission Organization	String	[0,1]	OXI
		Τροποποιήσεις		Modifications	String Array	[0,N]	OXI
22	CAPEC_ VIEW	Διαφορετική κατηγοριοποίηση των CAPEC					
		Αναγνωριστικό ID	ViewID		String	1	NAI
		Χαρακτηριστικό όνομα		Name	String	1	NAI
		Κατηγοριοποίηση σχετικά με την σχέση των μελών του view μεταξύ τους		Type	String	1	NAI
		Κατάσταση του view		Status	String	1	NAI
		Στόχος κατηγοριοποίησης		Objective	String	1	NAI
		Για συγκεκριμένα Types επιπλέον κατηγοριοποίηση		Filter	String	[0,1]	OXI
		Σημειώσεις		Notes	String	[0,1]	OXI
		Όνομα έκδοσης		Submission Name	String	[0,1]	OXI
		Ημερομηνία Έκδοσης		Submission Date	String	[0,1]	OXI
		Φορέας Έκδοσης		Submission Organization	String	[0,1]	OXI
		Τροποποιήσεις		Modification	String Array	[0,N]	OXI

	Πίνακας Σχέσεις						
#	**Ετικέτα**	**Από → Προς**	**Αναγνωριστικό**	**Χαρακτηριστικά**	**Τύπος**	**Πλήθος**	**Required**
1	parentOf	CPE → CPE					

2	belongsTo	CVE → GeneralInfo_CVE					
3	applicableIn	CVE → CPE	Vulnerable		Boolean	1	NAI
4	CVSS2_Impact	CVE → CVSS_2					
5	CVSS3_Impact	CVE → CVSS_3					
6	referencedBy	CVE → Reference_Data					
7	Problem_Type	CVE → CWE					
8	belongsTo	CWE → General_Info_CWE					
9	Related_Weakness	CWE → CWE	Nature		string	1	NAI
10	Applicable_Platform	CWE → Applicable_Platform					
11	hasExample	CWE → Demonstrative_Example					
		Περαιτέρω εξήγηση του παραδείγματος		Body Text	String Array	[0,N]	OXI
		Παράδειγμα της ευπάθειας σε κώδικα		Example Code	String Array	[0,N]	OXI
12	hasConsequence	CWE → Consequence					
		Τι επηρεάζει σε τεχνικό επίπεδο η εκμετάλλευση της ευπάθειας		Impact		[0,N]	OXI
		Επιπλέον Σημειώσεις		Note	String	[0,1]	OXI
		Πιθανότητα να εμφανιστεί η επίπτωση όταν γίνει εκμετάλλευση της ευπάθειας		Likelihood	String	[0,1]	OXI

13	canBeDetected	CWE → Detection_Method	Description		String	1	NAI
		Τα επίπεδα αποτελεσματικότητας της μεθόδου στην ανίχνευση της ευπάθειας		Effectiveness	String	[0,1]	OXI
		Σημειώσεις για την αποτελεσματικότητα		Effectiveness Notes	String	[0,1]	OXI
		Αναγνωριστικό της μεθόδου		Detection Method ID	String	1	NAI
14	hasMitigation	CWE → Mitigation					
15	RelatedAttackPattern	CWE → CAPEC					
16	hasExternal_Reference	CWE → External_Reference_CWE					
17	hasMember	CWE → CWE	ViewID		String	1	NAI
18	usefulFor	CWE_VIEW → Stakeholder					
		Περιγραφή του Stakeholder		Description	String	[0,1]	OXI
19	hasMember	CWE_VIEW → CWE					
20	hasConsequence	CAPEC → Consequence					
		Τι επηρεάζει σε τεχνικό επίπεδο η εκμετάλλευση της ευπάθειας		Impact		[0,N]	OXI
		Επιπλέον Σημειώσεις		Note	String	[0,1]	OXI
		Πιθανότητα να εμφανιστεί η επίπτωση όταν γίνει εκμετάλλευση της ευπάθειας		Likelihood	String	[0,1]	OXI
21	belongsTo	CAPEC → GeneralInfo_CAPEC					

22	hasMitigation	CAPEC → Mitigation					
23	Related_Attack_Pattern	CAPEC → CAPEC					
24	hasExternal_Reference	CAPEC → External_Reference_CAPEC					
25	hasMember	CAPEC → CAPEC					
26	usefulFor	CAPEC_VIEW → Stakeholder					
		Περιγραφή του Stakeholder		Description	String	[0,1]	OXI
27	hasMember	CAPEC_VIEW → CAPEC					
28	hasExternal_Reference	CAPEC_VIEW → External_Reference_CAPEC					