ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. «ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ»

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΚΩΝΣΤΑΝΤΙΝΟΣ ΛΑΜΠΡΙΝΟΥΔΑΚΗΣ

ΕΠΙΜΕΛΕΙΑ: ΦΩΤΟΜΑΡΑΣ ΝΙΚΟΣ

ΔΗΜΙΟΥΡΓΙΑ AGENT ΓΙΑ ΑΝΑΚΑΛΥΨΗ ΕΥΠΑΘΕΙΩΝ

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Κωνσταντίνο Λαμπρινουδάκη για την ανάθεση της διπλωματικής εργασίας. Η εργασία αυτή δεν θα είχε ολοκληρωθεί χωρίς την καθοδήγηση και την πολύτιμη βοήθειά του.

Επιπλέον, θα ήθελα να ευχαριστήσω τα στελέχη του λόχου κυβερνοάμυνας του Κέντρου Έρευνας Πληροφορικής Υποστήριξης Ελληνικού Στρατού (ΚΕΠΥΕΣ) για την καθοδήγηση, καθώς και για τις συμβουλές τις οποίες μου παρείχαν κατά τη διάρκεια εκπόνησης της διπλωματικής αυτής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη στήριξή τους σε κάθε απόφασή μου όλα αυτά τα χρόνια και τη συμπαράστασή τους στην εκπλήρωση κάθε στόχου μου.

ΠΕΡΙΛΗΨΗ

Η ραγδαία τεχνολογική ανάπτυξη, ιδιαίτερα στον τομέα της πληροφορικής έχει επιφέρει τεράστια οφέλη τόσο στους καθημερινούς απλούς χρήστες όσο και στις επιχειρήσεις ή τους οργανισμούς που τα χρησιμοποιούν. Επιπλέον, όμως έχει επιφέρει και τεράστιους κινδύνους, καθώς τα δεδομένα που ανταλλάσσονται στις περισσότερες περιπτώσεις είναι ζωτικής σημασίας ή ακόμα και προσωπικού χαρακτήρα.

Επιπροσθέτως, είναι γνωστό πως με τη πάροδο των χρόνων τα κρούσματα κυβερνοεπιθέσεων ολοένα και αυξάνονται, αφού στις περισσότερες περιπτώσεις δεν δίνετε η απαραίτητη σημασία στην ασφάλεια, κατά τη δημιουργία ή την ανάπτυξη διαφόρων λογισμικών και συστημάτων.

Εξαιτίας αυτού, πολλοί χρήστες ή ακόμα και επιχειρήσεις μπορεί να είναι ευάλωτοι σε διαφόρου τύπου επιθέσεων. Οι τρόποι επιθέσεων ποικίλουν και έτσι είναι εφικτό, κάποιος κακόβουλος χρήστης να μπορέσει με διάφορες τεχνικές (π.χ. e-mail spoofing) να εγκαταστήσει κακόβουλο λογισμικό στον υπολογιστή του ανυποψίαστου θύματος.

Όπως είναι επίσης γνωστό, πολλές εφαρμογές έχουν αναπτυχθεί χωρίς να έχει δοθεί η απαραίτητη σημασία στην ασφάλειά τους, με αποτέλεσμα να παρατηρούνται πολλά κρούσματα εκμετάλλευσης κενών ασφάλειας σε εφαρμογές, τα οποία εκμεταλλεύονται οι κακόβουλοι χρήστες με σκοπό να αποκτούν πρόσβαση στις συσκευές των θυμάτων τους.

Στη παρούσα εργασία παρουσιάζετε η ανάπτυξη ενός εργαλείου (agent), το οποίο θα αναζητά ευπάθειες σε βάσεις δεδομένων ανοιχτού τύπου και στη συνέχεια θα παρουσιάζει τα ανάλογα στοιχεία σε μορφή log file καθώς επίσης και με ένα ανάλογο report. Στόχος του συγκεκριμένου εργαλείου είναι η έγκυρη πρόληψη ενός περιστατικού ασφάλειας, εξετάζοντας τις εγκατεστημένες εφαρμογές που υπάρχουν στον υπολογιστή για τυχόν ευπάθειες, σύμφωνα με τα ευρήματα από τις σχετικές βάσεις δεδομένων.

Αρχικά, αναφέρονται κάποιοι ορισμοί για το τι είναι ένας agent και με ποιον τρόπο δουλεύει. Επιπλέον, περιγράφονται οι τεχνολογίες που χρησιμοποιούνται για την εκπόνηση της παρούσας εργασίας, καθώς επίσης και κάποιες σημαντικές λεπτομέρειες γύρω από τις συγκεκριμένες τεχνολογίες με σκοπό την κατανόηση της σημασίας χρήσης τους. Τέλος, παρουσιάζονται κάποιες εικόνες από τη λειτουργία του εργαλείου ώστε να είναι εφικτή η κατανόηση του τρόπου λειτουργίας του εργαλείου.

Περιεχόμενα

Κεφάλαιο 1ο

Δημιουργία Agent για Ανακάλυψη Ευπαθειών

1.1 Εισαγωγή

Η έξαρση της τεχνολογικής ανάπτυξης, έχει επιφέρει τεράστιες αλλαγές στη ζωή τόσο των απλών καθημερινών ανθρώπων όσο και στις επιχειρήσεις και τους οργανισμούς. Οι αλλαγές αυτές έχουν επηρεάσει σημαντικά τη λειτουργία των επιχειρήσεων καθώς και τη καθημερινότητα των απλών χρηστών, κυρίως θετικά.

Στις περισσότερες περιπτώσεις, η ανάπτυξη αυτή έχει καταφέρει να συμβάλει δραστικά στην απλούστευση πολλών διαδικασιών ή ακόμα και στη πραγματοποίηση καθημερινών δραστηριοτήτων με εύκολο και γρήγορο τρόπο, όπως για παράδειγμα στη πραγματοποίηση συναλλαγών ή πληρωμών με τη βοήθεια του e-banking, της ηλεκτρονικής τραπεζικής δηλαδή. Με τη βοήθεια του προαναφερθέντος συστήματος και άλλων αρκετών συστημάτων, υπάρχει η δυνατότητα πραγματοποίησης διαφόρων δραστηριοτήτων μέσα σε πολύ μικρό χρονικό διάστημα καθώς επίσης δίνετε η δυνατότητα για χρήση των συστημάτων αυτών από οπουδήποτε με τη χρήση ηλεκτρονικού υπολογιστή.

Στα προαναφερθέντα, έχει συμβάλει σημαντικά και η τεράστια έξαρση του διαδικτύου, το οποίο πλέον υπάρχει σε κάθε οργανισμό, επιχείρηση ή ακόμα και μέσα σε κάθε σπίτι. Το διαδίκτυο αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων όλων των ηλικιών, αφού θεωρείτε ως ένα εργαλείο το οποίο μπορεί να παρέχει χιλιάδες ευκολίες, όπως την άντληση πληροφοριών, την πραγματοποίηση συναλλαγών ή ακόμα και την επικοινωνία με τη χρήση mail, social media κ.α.

Όλα τα παραπάνω, έχουν ως αποτέλεσμα τη ροή τεράστιου όγκου δεδομένων, τα οποία σε πολλές περιπτώσεις μπορεί να είναι άκρως σημαντικά, διότι μπορεί να αποτελούν δεδομένα προσωπικού χαρακτήρα ή ακόμα και κρίσιμα δεδομένα όπως είναι οι αριθμοί πιστωτικών καρτών, αριθμοί λογαριασμών κ.α.

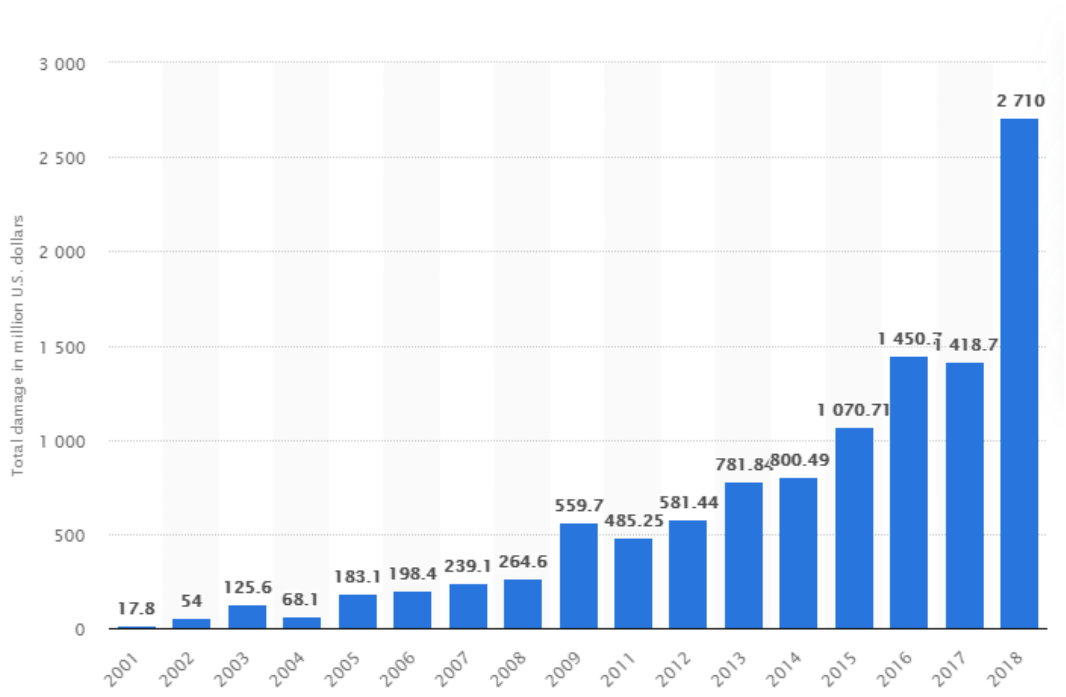
Επιπροσθέτως, η τεράστια τεχνολογική ανάπτυξη έχει επιφέρει εκτός των παραπάνω και σημαντική ανάπτυξη του ηλεκτρονικού εγκλήματος (ηλεκτρονικές απάτες, cyber bullying, phishing attacks), τα οποία τείνουν να ακολουθούν κυρίως νεαρά άτομα τα οποία στοχεύουν συνήθως στην απόσπαση πληροφοριών ή σημαντικών δεδομένων από το θύμα τους. Αυτό μπορεί να επιτευχθεί με διάφορους τρόπους και τεχνικές ή ακόμα και σε πολλές περιπτώσεις με την εκμετάλλευση αδυναμιών που μπορεί να έχουν διάφορες εφαρμογές που χρησιμοποιούνται καθημερινά εκατομμύρια χρήστες.

Έτσι λοιπόν μπορεί κανείς εύκολα να αντιληφθεί πως μία νέα απειλή έχει πλέον εισβάλει για τα καλά στη καθημερινότητα σχεδόν όλων όσων τείνουν να χρησιμοποιούν ηλεκτρονικό υπολογιστή ή ακόμα και έξυπνες συσκευές, είτε στον απλό καθημερινό τους χρόνο είτε στην επαγγελματική τους ζωή. Η νέα αυτή απειλή είναι το κυβερνοέγκλημα, το οποίο τείνει να αυξάνεται καθημερινά με τη πάροδο των χρόνων και τη ραγδαία έξαρση της τεχνολογίας. Δεν είναι λίγες οι φορές που έχουν

καταγραφεί σημαντικές επιθέσεις με στόχο το κέρδος, οι οποίες έχουν αποτελέσει σημαντικό παράγοντα για τη δυσλειτουργία αρκετών επιχειρήσεων ή ακόμα και για την καταστροφή τους. Αρκετά πρόσφατο και ίσως το πιο τρανταχτό παράδειγμα, είναι αυτό του ransomware wannacry, εξαιτίας του οποίου κατεγράφησαν τεράστιες απώλειες τόσο οικονομικές όσο και απώλειες δεδομένων από αρκετές επιχειρήσεις και οργανισμούς. Προκειμένου να πραγματοποιηθεί η επίθεση αυτή, οι επιτιθέμενοι αξιοποιήσαν μία ευπάθεια των συστημάτων windows, η οποία ονομάζετε EternalBlue και στη συνέχεια το εργαλείο DoublePulsar, ώστε να εγκατασταθεί το ransomware. Το wannacry ransomware στόχευε στη κρυπτογράφηση των όλων των αρχείων και των δεδομένων που βρίσκονταν στον υπολογιστή του εκάστοτε χρήστη, με σκοπό στη συνέχεια να ζητώνται λύτρα τα οποία έπρεπε να αποπληρωθούν σε bitcoin έτσι ώστε ο χρήστης να λάβει ένα κλειδί αποκρυπτογράφησης. Φυσικά κάτι τέτοιο δεν πραγματοποιήθηκε σε καμία περίπτωση με αποτέλεσμα πολλές εταιρίες να καταγράψουν τεράστιες απώλειες δεδομένων και έτσι πολλές από αυτές αναγκάστηκαν να τερματίσουν τη λειτουργία τους. Αξίζει να σημειωθεί, πως το ransomware εμφανίστηκε παγκοσμίως το Μάιο του 2017 και το πιο αξιοσημείωτο γεγονός είναι πως στόχευες σε μία αδυναμία στο SMB πρωτόκολλο των Windows, για το οποίο η Microsoft παρείχε ενημερώσεις συστήματος. Έτσι λοιπόν μπορεί κανείς να συμπεράνει πως αρκετοί ήταν αυτοί οι οποίοι δεν είχαν πραγματοποιήσει τις ενημερώσεις αυτές με αποτέλεσμα να καταγραφούν τεράστιοι αριθμοί επιθέσεων παγκοσμίως.

Εκτός από τη προαναφερθείσα επίθεση, η έξαρση του κυβερνοεγκλήματος βρίσκεται σε ραγδαία ανάπτυξη, αφού καθημερινά παρατηρούνται επιθέσεις ακόμα και σε δημόσιες υποδομές, με αποτέλεσμα να γίνεται σε αρκετές περιπτώσεις αρκετά δύσκολη η λειτουργία των υποδομών αυτών. Με βάση όλα τα παραπάνω, μπορεί κανείς να εξάγει το συμπέρασμα πως πλέον η τεράστια έξαρση της τεχνολογίας σε συνδυασμό με την κακόβουλη χρήση της έχει επιφέρει μία νέα μορφή πολέμου, τον κυβερνοπόλεμο.

Για αυτό το λόγο είναι σημαντικό να είναι κανείς αρκετά προσεκτικός στον τρόπο με τον οποίο χρησιμοποιεί τον ηλεκτρονικό υπολογιστή ή γενικότερα τις έξυπνες συσκευές, αφού πέρα από ένα εργαλείο που μπορεί να εξυπηρετήσει και να διευκολύνει αρκετά τη καθημερινή ζωή όλων, μπορεί επιπροσθέτως να αποτελέσει σημαντική απειλή και αιτία αρκετών προβλημάτων.



Εικόνα 1 Χρηματική αξία απωλειών σε εκατομμύρια δολάρια αμερικής (πηγή: https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/)

Σύμφωνα με το παραπάνω διάγραμμα (το οποίο προέρχετε από την πηγή statista https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/), οι χρηματικές απώλειες εξαιτίας του κυβερνοεγκλήματος ανέρχονται σε εκατοντάδες εκατομμύρια ανά χρόνο. Το πιο ανησυχητικό γεγονός όμως, είναι πως με τη πάροδο των χρόνων παρατηρεί κανείς τεράστια αύξηση των χρηματικών απωλειών, πράγμα που σημαίνει πως ολοένα και περισσότεροι τείνουν να στραφούν στη διάπραξη κυβερνοεγκλημάτων με σκοπό το κέρδος. Ειδικά για το 2018 παρατηρείται αύξηση σχεδόν 100% σε σχέση με το 2017, για τις χρηματικές απώλειες των επιχειρήσεων.

Για αυτό το λόγο, δημιουργείται η ανάγκη για καταπολέμηση όλων αυτών των περιστατικών. Προκειμένου να επιτευχθεί αυτό, είναι σημαντικό οι χρήστες να γνωρίζουν και να ενημερώνονται για τους κινδύνους του διαδικτύου και επιπλέον να υπάρχει η ύπαρξη εργαλείων που θα αυτοματοποιούν αρκετές διαδικασίες θωράκισης ενός συστήματος, ώστε να προστατευθούν ακόμα και οι απλοί χρήστες οι οποίοι δεν διαθέτουν γνώσεις περί ασφάλειας πληροφοριών.

1.2 Καθορισμός του Προβλήματος

Όπως αναφέρθηκε και προηγουμένως, η τεράστια τεχνολογική ανάπτυξη εκτός από πολλά οφέλη για τη καθημερινότητα όλων, επέφερε και αρκετούς κινδύνους σε ότι αφορά την προστασία των δεδομένων και των πληροφοριών, καθώς δεν είναι λίγοι εκείνοι οι οποίοι επιλέγουν να χρησιμοποιήσουν τα οφέλη της τεχνολογίας με κακόβουλο τρόπο έχοντας ως σκοπό συνήθως το κέρδος μέσα από αυτό.

Γενικότερα, παρατηρείται πως ο αριθμός των κυβερνοεπιθέσεων και των περιστατικών ασφάλειας ολοένα και αυξάνετε με τη πάροδο των χρόνων, καθώς πολλές από τις νέες τεχνολογίες που παρουσιάζονται είναι ευάλωτες σε διάφορους τύπους επιθέσεων, διότι δεν δίνετε η απαραίτητη σημασία στην ασφάλεια.

Εξάλλου αν κανείς περιηγηθεί σε βάσεις δεδομένων ανοιχτού τύπου όπως για παράδειγμα η exploit database (https://www.exploit-db.com/) μπορεί να εντοπίσει χιλιάδες ευπάθειες που έχουν κατά καιρούς ανακαλυφθεί για διάφορες εφαρμογές ή ακόμη και δημοφιλή λειτουργικά συστήματα όπως για παράδειγμα τα Windows.

Έτσι λοιπόν, είναι εμφανής η αναγκαιότητα ενημέρωσης σχετικά με το ποιες ευπάθειες έχουν ανακαλυφθεί στο παρελθόν για διάφορες εφαρμογές ή λειτουργικά συστήματα, ειδικά όταν το σύνολο των προαναφερθέντων χρησιμοποιούνται σχεδόν από κάθε χρήστη ηλεκτρονικού υπολογιστή ανεξαρτήτου ειδικότητας.

Παρόλα αυτά, η εν λόγο ενημέρωση αποτελεί μια χρονοβόρα διαδικασία ειδικά στην περίπτωση που κάποιος δεν είναι εξοικειωμένος με τον χώρο της ασφάλειας ώστε να γνωρίζει τις τεχνικές και τις διαδικασίες σχετικά με την ανακάλυψη ευπαθειών.

Για αυτό τον λόγο, δημιουργείτε η ανάγκη ύπαρξης ενός εργαλείου το οποίο θα μπορεί να ελέγχει ποιες εκδόσεις εφαρμογών είναι εγκατεστημένες στον ηλεκτρονικό υπολογιστή του χρήστη και να ψάχνει αυτομάτως σε διάφορες ανοιχτού τύπου βάσεις δεδομένων ώστε να ανακαλύψει αν τυχόν υπάρχουν περιστατικά ασφάλειας για τις συγκεκριμένες εφαρμογές. Στη συνέχεια, το εργαλείο αυτό θα εξάγει μία αναφορά σχετικά με το ποιες είναι οι ευπαθείς εφαρμογές καθώς επίσης και λεπτομέρειες σχετικά με την ευπάθεια που παρουσιάζουν. Επιπροσθέτως, στην αναφορά θα υπάρχουν και κάποια χαρακτηριστικά διαγράμματα σχετικά με το βαθμό επικινδυνότητα των ευπαθειών (cve scores).

Τέλος να αναφερθεί πως το εργαλείο αυτό θα πραγματοποιεί την αναζήτηση για ευπάθειες, στις πιο γνωστές βάσεις δεδομένων όπως οι eploit-db, rapid7 και cve-details.

Κεφάλαιο 2ο

Web Scraping

2.1 Τί είναι το Web Scraping

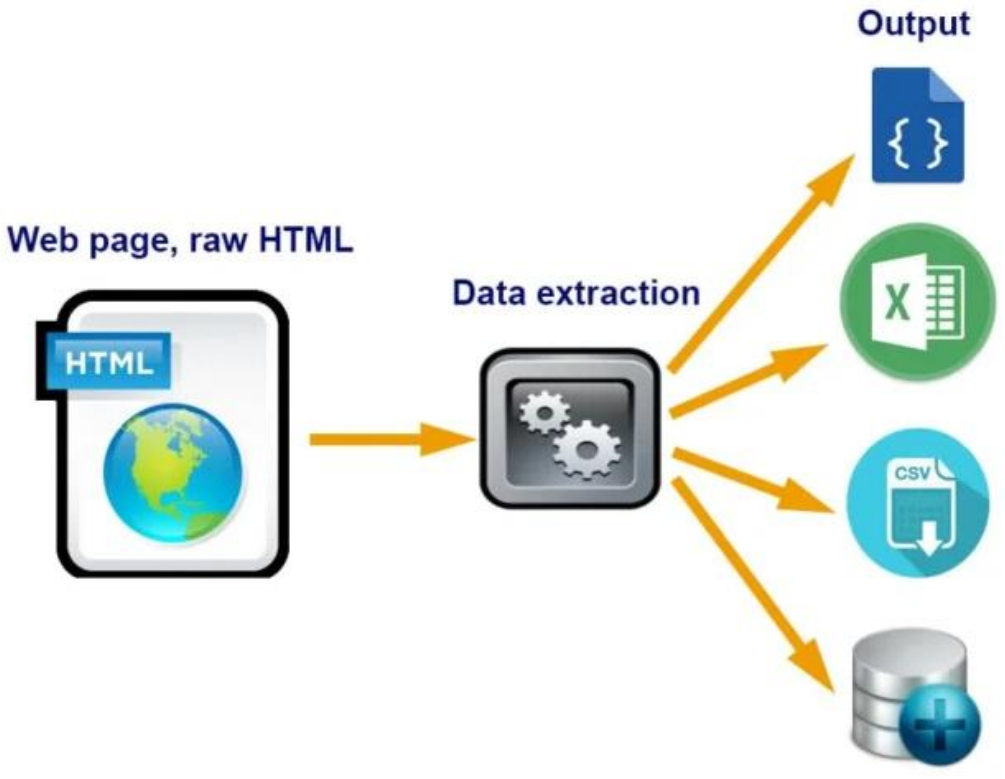
Ως Web Scraping ή Web Harvesting ορίζετε η διαδικασία κατά την οποία πραγματοποιείτε συλλογή δεδομένων και πληροφοριών μέσα από διάφορες ιστοσελίδες, μηχανές αναζήτησης ή ακόμα και κοινωνικά δίκτυα. Ένας γενικότερος ορισμός για το Web scraping θα μπορούσε να δοθεί ως η διαδικασία συλλογής δεδομένων και πληροφοριών μέσα από το διαδίκτυο με σκοπό την επεξεργασία τους.

Η παραπάνω διαδικασία μπορεί να πραγματοποιηθεί όπως προαναφέρθηκε με την βοήθεια των περιηγητών (browsers) ή με τη χρήση του πρωτοκόλλου HTTP (Hypertext Transfer Protocol). Αξίζει να σημειωθεί, πως η διαδικασία αυτή μπορεί να πραγματοποιηθεί από κάποιον χρήστη χειροκίνητα, αλλά συνήθως γίνετε χρήση αυτοματοποιημένων εργαλείων τα οποία ονομάζονται Web Crawlers. Οι Web Crawlers είναι υπεύθυνοι ώστε να πραγματοποιήσουν λήψη της ιστοσελίδας με σκοπό να επεξεργαστεί στη συνέχεια και να πραγματοποιηθεί η εξαγωγή της πληροφορίας με χρήση αυτοματοποιημένων και πάλι εργαλείων, τα οποία ονομάζονται Spiders (αναλύονται στη συνέχεια). Συνήθως τα δεδομένα που εξάγονται συλλέγονται είτε σε βάσεις δεδομένων, είτε σε τοπικά αρχεία όπως comma separated values (csv), xml αρχεία ή json αρχεία τα οποία αποτελούν τους δημοφιλέστερους τύπους αρχείων για το σκοπό αυτό.

Αξίζει επιπλέον να σημειωθεί, πως η αυτοματοποίηση της διαδικασίας του Web Scraping μπορεί να οδηγήσει στην επεξεργασία τεράστιου όγκου δεδομένων, αφού κατά τη διαδικασία αυτή, οι Web Crawlers αναλαμβάνουν να κατεβάσουν μεγάλο αριθμό ιστοσελίδων και στη συνέχεια επεξεργάζονται με τη βοήθεια των Web Spiders ώστε να πραγματοποιηθεί η εξαγωγή των επιθυμητών δεδομένων. Επιπροσθέτως, με τη χρήση των Web Scrapers είναι εφικτή η εξαγωγή μεγάλου όγκου δεδομένων από διάφορες πηγές ταυτόχρονα, γεγονός που μπορεί να εξοικονομήσει χρόνο για τον χρήστη, καθώς επίσης μπορεί να οδηγήσει στη μεγαλύτερη συλλογή δεδομένων από διάφορες πηγές.

Παρόλα τα οφέλη του Web Scraping μελανό σημείο μπορεί να αποτελέσει η νομιμότητα της διαδικασίας αυτής, αφού πολλές ιστοσελίδες δεν διαθέτουν τα δεδομένα τους προς αντιγραφή ή επεξεργασία. Το αν μία ιστοσελίδα επιτρέπει την εξαγωγή ή αντιγραφή των δεδομένων της με οποιοδήποτε τρόπο θα πρέπει να αναγράφετε στους όρους υπηρεσιών (Terms of Service) ή στο αρχείο robots.txt, το οποίο υπάρχει σε κάθε ιστοσελίδα και μέσα σε αυτό αναγέρνεται ρητά το τί επιτρέπετε και τί όχι. Με άλλα λόγια είναι ένα αρχείο το οποίο αναφέρει συγκεκριμένα, ποιες ενεργείς μπορεί να επιτρέπονται να πραγματοποιηθούν από διάφορα bots, μέσα στα οποία ανήκουν διάφορα εργαλεία ανάμεσά τους και οι Web Crawlers, και ποιες όχι.

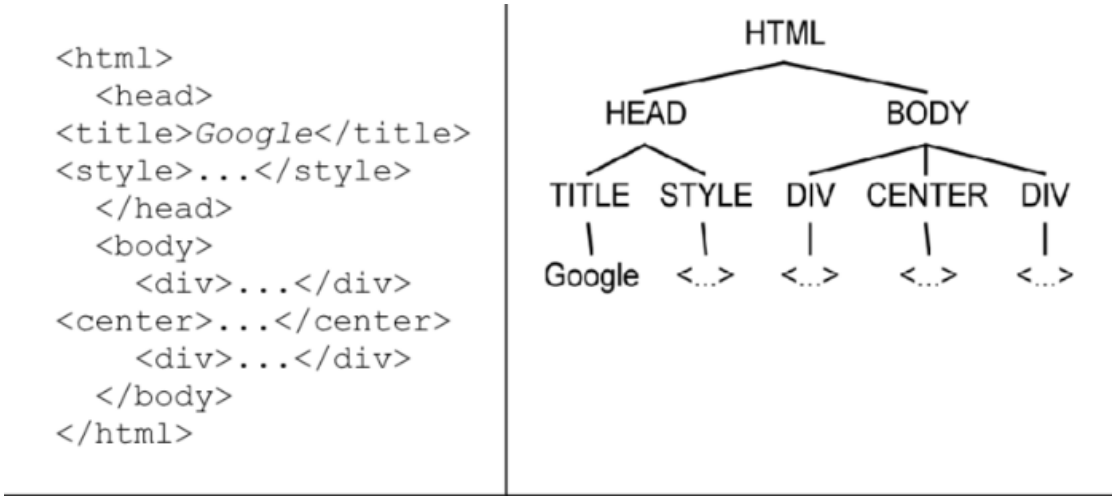
Το αρχείο αυτό, θα πρέπει να λαμβάνετε σοβαρά υπόψιν, καθώς σε περίπτωση που δεν αναφέρετε μέσα σε αυτό το Web Crawling, τότε οποιαδήποτε ενέργεια σχετικά με αυτό μπορεί να θωρηθεί παράνομη και να επιφέρει ανάλογες κυρώσεις.



Εικόνα 2 Web Scraping (πηγή: https://www.fiverr.com/categories/business)

Όπως φαίνεται και στη παραπάνω εικόνα, το Web Scraping ουσιαστικά εξυπηρετεί στην εξαγωγή διασκορπισμένων δεδομένων μέσα από διάφορους ιστότοπους, με σκοπό τη δόμησή ή την κατηγοριοποίησή τους και στη συνέχεια αποθήκευσή τους σε τοπικά αρχεία ή βάσεις δεδομένων. Προκειμένου όμως να επιτευχθεί αυτό είναι αρχικά αναγκαία η γνώση και η κατανόηση της γλώσσας Html, διότι μέσα από αυτή τη κατανόηση μπορεί να επιλεχθεί στη συνέχεια το κατάλληλο εργαλείο καθώς και η κατάλληλη τεχνική προκειμένου να ολοκληρωθεί επιτυχώς η διαδικασία του Web Scraping.

Πιο συγκεκριμένα, μία ιστοσελίδα αποτελείται από διάφορα Html elements, τα οποία καθορίζουν αν το περιεχόμενό τους θα είναι κάποια εικόνα, κείμενο, κάποιο μέσο αναπαραγωγής. Όλα αυτά, ανάλογα βέβαια και την ιστοσελίδα, δομούνται μεταξύ τους ώστε να μπορέσουν να αναπαραστήσουν σωστά τα δεδομένα μέσα στην ιστοσελίδα.



Εικόνα 3 Δεντρική απεικόνιση Html (πηγή: https://www.researchgate.net/publication/266611108_Using_XPaths_of_Inbound_Links_to_Cluster_Template-Generated_Web_Pages)

Προκειμένου κάποιος να μπορέσει να εξάγει δεδομένα μέσα από οποιαδήποτε ιστοσελίδα επιθυμεί, είναι αναγκαίο να μπορεί να κατανοήσει με ποιο τρόπο θα μπορέσει να φτάσει στη πληροφορία αυτή. Με άλλα λόγια θα πρέπει να είναι σε θέση να καθορίσει το μονοπάτι που θα ακολουθήσει, ανάλογα και που βρίσκονται τα δεδομένα, ώστε να εφαρμοστεί η εξόρυξη. Σύμφωνα με το παραπάνω παράδειγμα της εικόνας αν η πληροφορία βρίσκεται σε κάποιο από τα div elements, τότε ο χρήστης θα πρέπει να ακολουθήσει το μονοπάτι HTML → Body →Div→ Div's Child→…

Αυτό συμβαίνει, διότι ένας Web Scraper αρχικά πραγματοποιεί λήψη του κώδικα της ιστοσελίδας και στη συνέχεια, με χρήση κώδικα αλλά και τεχνικών όπως για παράδειγμα τα xpaths, ο χρήστης έχει τη δυνατότητα να φτάσει στη πληροφορία που αναζητά και να την εξάγει. Πρόκειται ουσιαστικά για τη διαδικασία της αναζήτησης η οποία μπορεί να οριστεί αποκλειστικά από τον εκάστοτε χρήστη, και βάσει αυτής καθορίζετε και η πολυπλοκότητα όπως και ο χρόνος της διαδικασίας της εξόρυξης δεδομένων. Για αυτό το σκοπό είναι αρκετά σημαντικό ο χρήστης να μπορεί να γνωρίζει και να επιλέγει τα κατάλληλα, ανάλογα με την περίσταση, εργαλεία και τεχνικές με σκοπό να πραγματοποιηθεί η όλη διαδικασία όσο το δυνατό πιο αποδοτικά.

2.2 Τεχνικές Web Scraping

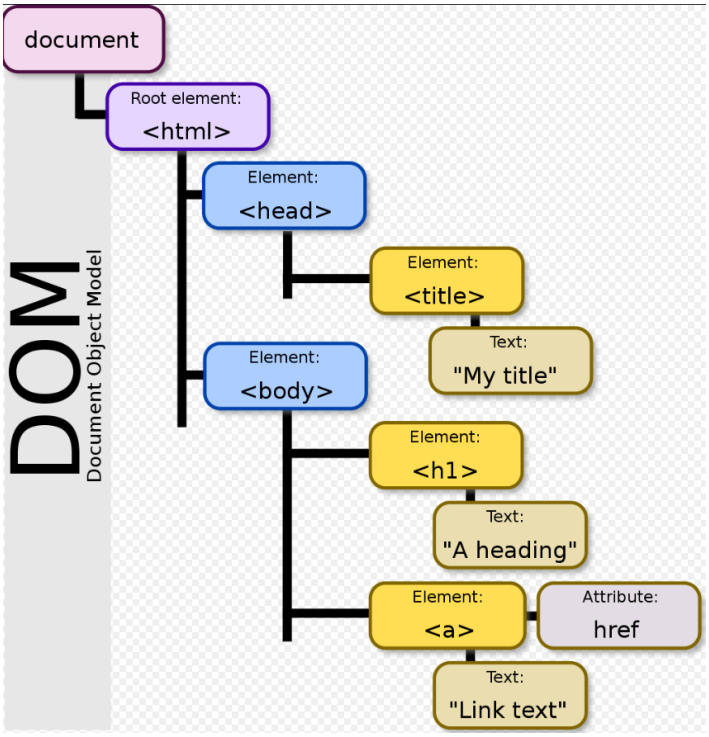
A. HTML Parsing

Μία από τις τεχνικές του Web Scraping είναι αυτή που ουσιαστικά αναφέρθηκε προηγουμένως, δηλαδή αυτή της άντλησης των δεδομένων ακολουθώντας το μονοπάτι ως το επιθυμητό στοιχείο. Προκειμένου ο χρήστης να διευκολυνθεί, συνήθως στη διαδικασία αυτή χρησιμοποιούνται γλώσσες ημιδομημένων δεδομένων όπως η HTQL. Πρόκειται για γλώσσα, η οποία στοχεύει στην άντληση πληροφοριών ανάλογα με τη δενδρική δομή του κώδικα της HTML . Επομένως, η διαδικασία της αναζήτησης του κατάλληλου στοιχείου μπορεί να απλοποιηθεί αρκετά με χρήση της γλώσσας αυτής, χωρίς ο χρήστης να χεριάζετε να αναζητήσει το κατάλληλο μονοπάτι. Το μόνο που χρειάζεται είναι να μπορέσει να δομήσει σωστά το κατάλληλο ερώτημα (Query) ώστε να αντλήσει την επιθυμητή πληροφορία.

B. Document Object Model (DOM)Parsing

Η γλώσσα DOM, είναι μία cross-platform διεπαφή που αντιμετωπίζει ένα έγγραφο XML ή HTML ως δομή δέντρου όπου κάθε κόμβος είναι ένα αντικείμενο που αντιπροσωπεύει ένα μέρος του εγγράφου. Το DOM αναπαριστά ένα έγγραφο με ένα λογικό δέντρο. Κάθε κλαδί του δέντρου καταλήγει σε έναν κόμβο και κάθε κόμβος περιέχει αντικείμενα. Οι μέθοδοι DOM επιτρέπουν την προγραμματική πρόσβαση στο δέντρο. μαζί τους μπορεί κανείς να αλλάξει τη δομή, το στυλ ή το περιεχόμενο ενός εγγράφου. Οι κόμβοι μπορούν να φέρουν χειριστές συμβάντων σε αυτές. Μόλις ενεργοποιηθεί ένα συμβάν, οι χειριστές συμβάντων εκτελούνται

(πηγή : https://en.wikipedia.org/wiki/Document_Object_Model).



Εικόνα 4 DOM display (πηγή: https://en.wikipedia.org/wiki/Document_Object_Model)

Γ. Εργαλεία Web Scraping

Πλέον υπάρχουν αρκετά εργαλεία τα οποία αυτοματοποιούν, αν όχι όλη τη διαδικασία, τότε σίγουρα ένα σημαντικό κομμάτι της διαδικασίας του Web Scraping, καθώς τα βήματα που πρέπει να ακολουθήσει κανείς είναι αρκετά προκειμένου να εξάγει τη πληροφορία αυτή, ειδικά αν πρόκειται για πληθώρα ιστοσελίδων και πληροφοριών. Για αυτό το σκοπό έχουν δημιουργηθεί ολοκληρωμένες λύσεις όπως είναι για παράδειγμα τα διάφορα frameworks με σκοπό να μειώσουν όσο γίνετε το χρόνο εκτέλεσης της διαδικασίας. Τα frameworks αυτά ποικίλουν ανάλογα και με το σε ποια γλώσσα προγραμματισμού επιθυμεί ο εκάστοτε χρήστης να χρησιμοποιήσει.

Δ. XPATH

Το XPath είναι μια γλώσσα που περιγράφει έναν τρόπο εντοπισμού και επεξεργασίας αντικειμένων σε έγγραφα XML (Extensible Markup Language) χρησιμοποιώντας μια σύνταξη διεύθυνσης που βασίζεται σε μια διαδρομή μέσω της λογικής δομής ή της ιεραρχίας του εγγράφου. Προκειμένου να χρησιμοποιήσει κανείς τα Xpaths είναι αναγκαίο να γνωρίζει το ανάλογο συντακτικό, ώστε να μπορέσει να εξάγει τη πληροφορία που επιθυμεί.

XPath:

```
/wikimedia/projects/project/editions/*[2]
```

XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
 <projects>
  <project name="Wikipedia" launch="2001-01-05">
   <editions>
    <edition language="English">en.wikipedia.org</edition>
    <edition language="German">de.wikipedia.org</edition>
    <edition language="French">fr.wikipedia.org</edition>
    <edition language="Polish">pl.wikipedia.org</edition>
   </editions>
  </project>
  <project name="Wiktionary" launch="2002-12-12">
   <editions>
    <edition language="English">en.wiktionary.org</edition>
    <edition language="French">fr.wiktionary.org</edition>
    <edition language="Vietnamese">vi.wiktionary.org</edition>
    <edition language="Trukish">tr.wiktionary.org</edition>
   </editions>
  </project>
 </projects>
</wikimedia>
```

Εικόνα 5 Παράδειγμα Xpath (πηγή: https://en.wikipedia.org/wiki/XPath)

Ε. APIs

Υπάρχουν αρκετά Api, τα οποία διατίθενται για συγκεκριμένους σκοπούς. Για παράδειγμα υπάρχουν έτοιμες λύσεις με σκοπό την εξαγωγή δεδομένων από το Youtube ή από διάφορα κοινωνικά δίκτυα. Μέσω αυτών των Api, δίνεται η δυνατότητα για απλή αποστολή Http αιτημάτων και λήψη των αντίστοιχων απαντήσεων με τα επιθυμητά δεδομένα. Ουσιαστικά, μέσα από αυτά τα Apis πραγματοποιείται αναζήτηση στις βάσεις δεδομένων των αντίστοιχων ιστοσελίδων ώστε να εξαχθούν τα δεδομένα που ζητούνται από τον χρήστη κατά το αίτημα.

ΣΤ. Μηχανική Μάθηση

Η μηχανική μάθηση (machine learning) σε συνδυασμό με τη τεχνητή νοημοσύνη έχουν σηματοδοτήσει μία νέα εποχή για το κόσμο της πληροφορίας και όχι μόνο. Πλέον ολοένα και περισσότερα λογισμικά τα οποία αναπτύσσονται στοχεύουν στο να υλοποιούνται σε συνδυασμό με τη τεχνητή νοημοσύνη και τη μηχανική μάθηση, έτσι ώστε να μπορούν να γίνονται ολοένα και εξυπνότερα. Αυτό γίνετε με σκοπό την αυτονομία των λογισμικών, καθώς με τη χρήση των προαναφερθέντων τεχνολογιών, παρέχεται η δυνατότητα στο να γίνονται πιο έξυπνοι οι αλγόριθμοι. Όσον αφορά, τους Web Scrapers, τόσο η τεχνητή νοημοσύνη όσο και η μηχανική μάθηση μπορούν να συμβάλουν σημαντικά στην εξέλιξή τους, ώστε να μπορούν μέσα από τις ιστοσελίδες και τα δεδομένα τα οποία αντλούν να γίνονται όλο και εξυπνότεροι. Βάσει αυτού ίσως στο μέλλον να μη χεριάζετε να πραγματοποιούνται ραγδαίες αλλαγές στην υλοποίηση των Web Scrapers, ώστε να αντλούν πληροφορίες ακόμα και αν μία ιστοσελίδα αλλάξει κάποια από τα στοιχεία της ή τη δομή της.

2.3 Λογισμικό Web Scraping

Εκτός από τις παραπάνω τεχνικές, οι οποίες μπορούν να χρησιμοποιηθούν, ώστε να αναπτυχθεί λογισμικό που στη συνέχεια μπορεί να χρησιμοποιηθεί για Web Scraping, υπάρχουν έτοιμα εργαλεία τα οποία πραγματοποιούν τις ενέργειες που έχουν αναφερθεί παραπάνω. Επιπλέον, παρέχονται και βιβλιοθήκες για διάφορες γλώσσες προγραμματισμού, οι οποίες αυτοματοποιούν αρκετές από τις απαιτούμενες διαδικασίες και έτσι εξοικονομούν χρόνο στον χρήστη αλλά μειώνουν ταυτόχρονα και την πολυπλοκότητα της ανάπτυξης του λογισμικού. Οι κατηγορίες λογισμικών σχετικά με το Web Scraping είναι οι εξής: Λογισμικό cloud, Desktop λογισμικό, Βιβλιοθήκες για γλώσσες προγραμματισμού, Browser Extensions.

A. Λογισμικό cloud

Σε αυτή τη κατηγορία ανήκουν οι λύσεις οι οποίες, παρέχουν μία διεπαφή για το χρήστη συνήθως μέσα από κάποιον browser, ενώ το backend κομμάτι της εφαρμογής βρίσκεται σε κάποιο cloud. Με αυτή τη λύση, μπορεί κανείς να διασφαλίσει το κομμάτι της απόδοσης, αφού το λογισμικό δεν βασίζετε στην υπολογιστική δύναμη που έχει ο εκάστοτε χρήστης, καθώς το κομμάτι που πραγματοποιεί τη διαδικασία εξόρυξης αλλά και επεξεργασίας των δεδομένων βρίσκετε στο cloud. Κατά αυτό τον τρόπο μπορούν να ολοκληρωθούν έργα με μεγάλες απαιτήσεις, σε αρκετά σύντομο χρονικό διάστημα. Παρόλα αυτά όμως, υπάρχουν και κάποια αρνητικά σε αυτή τη λύση. Το πιο σημαντικό από αυτά, είναι πως σε κάποιες περιπτώσεις ενδέχεται ο Web Scraper να μην έχει πρόσβαση στην ιστοσελίδα από την οποία θέλει ο χρήστης να αντλήσει δεδομένα. Αυτό μπορεί να συμβαίνει, διότι μερικές ιστοσελίδες δεν επιτρέπουν τη πρόσβαση σε χρήστες με ip διευθύνσεις από άλλες χώρες ή ηπείρους.

Δημοφιλείς Cloud-based Web Scrapers:

1. Scrapestack
2. Scrapinghub
3. Mozenda

B. Desktop λογισμικό

Οι λύσεις γύρω από το Desktop λογισμικό αναφορικά με τους Web Scrapers, βασίζονται κατά κύριο λόγο στο hardware που διαθέτει ο εκάστοτε χρήστης. Αυτό σημαίνει πως, σε αντίθεση με τη λύση των εφαρμογών cloud, σε αυτή τη περίπτωση απαιτείται αρκετά μεγάλη υπολογιστική δύναμη προκειμένου να μπορέσουν να ολοκληρωθούν απαιτητικά έργα. Αυτό συνεπάγεται, πως ο χρήστης θα πρέπει να διαθέσει ένα αρκετά σημαντικό ποσό για την αναβάθμιση του συστήματός του, παρόλα αυτά όμως του δίνετε η δυνατότητα για παραμετροποίηση, αφού σε αυτή τη περίπτωση μπορεί εκείνος να τροποποιεί για παράδειγμα την ip διεύθυνσή του ώστε να πραγματοποιήσει την άντληση πληροφοριών. Δημοφιλείς Desktop Scrapers:

1. ParseHub
2. FMiner

Γ. Βιβλιοθήκες για γλώσσες προγραμματισμού

Σε αυτή τη περίπτωση, δίνετε η δυνατότητα στον χρήστη να δημιουργήσει έναν εξολοκλήρου δικό του Web Scraper, σύμφωνα με τις ανάγκες και τις απαιτήσεις του. Είναι σημαντικό να αναφερθεί, πως οι λύσεις σε αυτή τη περίπτωση ποικίλουν, ανάλογα και με τις προτιμήσεις που έχει ο χρήστης σχετικά με τις γλώσσες προγραμματισμού. Σημαντικό είναι επίσης το γεγονός πως επιτρέπουν στους προγραμματιστές εξοικονομήσουν χρόνο, διότι επιτρέπουν την επαναχρησιμοποίηση κώδικα και επιπλέον προσφέρουν αυτοματοποιημένες λύσεις για πολλά από τα επιμέρους κομμάτια της διαδικασίας. Τέλος, αποτελούν την ιδανική λύση σε περίπτωση που κάποια από τις έτοιμες λύσεις που αναφέρθηκαν προηγουμένως δεν ταιριάζει.

	Dynamic Content	DOM navigation	DB integration	Programing language
Scrapy	YES	YES	YES	Python
Goutte	NO	YES	YES	PHP
Capybara	YES	YES	YES	Ruby
Jaunt	NO	YES	YES	Java
Web::Scraper	NO	YES	YES	Perl
Mechanize	NO	YES	YES	Ruby

Εικόνα 6 (πηγή: Web Scraping and Data Extraction From Websites, Author: Vojtech Draxl)

Δ. Browser Extensions

Σε αυτή τη κατηγορία ανήκουν λύσεις οι οποίες δίνουν τη δυνατότητα στο χρήστη να εξάγει χειροκίνητα κάποια δεδομένα. Μπορούν να χρησιμοποιηθούν μόνο σε μικρά έργα τα οποία δεν έχουν μεγάλες απαιτήσεις. Ο καθένας έχει πρόσβαση στα εργαλεία αυτά μέσα από τα ηλεκτρονικά καταστήματα των περιηγητών διαδικτύου.

Δημοφιλείς Browser Extension Web Scrapers:

1. Web Scraper
2. Data Scraper
3. Grepsr

2.4 Νομικές Πτυχές του Web Scraping

Η εξαγωγή δεδομένων με χρήση Web Scrapers, μπορεί να τεθεί υπό αμφισβήτηση αναφορικά με τη νομιμότητά της από τους κατόχους των δεδομένων που προβάλλονται στις ιστοσελίδες. Αυτό συμβαίνει, διότι αρκετοί οργανισμοί, βασίζουν τη λειτουργία και την επιβίωσή τους στις ιστοσελίδες τους και τα δεδομένα που αυτές προβάλλουν. Η εξαγωγή των δεδομένων αυτώ σε πολλές περιπτώσεις μπορεί να θεωρηθεί παράνομη, αφού προκειμένου να πραγματοποιηθεί η εξαγωγή δεδομένων, ο κάτοχος των δεδομένων αυτών δεν ερωτάτε όπως θα έπρεπε και επομένως μπορεί να θεωρηθεί πως η εξαγωγή πραγματοποιείται χωρίς τη συγκατάθεσή του.

Επιπλέον, σε πολλές περιπτώσεις η εξόρυξη και η εξαγωγή δεδομένων , μπορεί να θεωρηθεί κλοπή πνευματικής ιδιοκτησίας, καθώς μπορεί τα δεδομένα αυτά να προβληθούν σε κάποια άλλη ιστοσελίδα ή εφαρμογή. Για παράδειγμα, μπορεί κάποιος να χρησιμοποιήσει έναν Web Scraper ώστε να μπορέσει να αντλήσει όλες τις φωτογραφίες μέσα από μία ιστοσελίδα, με σκοπό στη συνέχεια να τις προβάλει σε κάποια άλλη. Αυτό θεωρείται κλοπή πνευματικής υπηρεσίας η οποία μπορεί να επιφέρει σημαντικές κυρώσεις.

Εκτός από τα παραπάνω, σχεδόν όλες οι ιστοσελίδες, αναφέρουν στον όρους χρήσης τους πως απαγορεύτε ρητά η οποιαδήποτε χρήση ή εξαγωγή των δεδομένων τους χωρίς ρητή συγκατάθεση. Επομένως, η οποιαδήποτε ενέργεια αντίθετη σε αυτό το κανονισμό μπορεί να επιφέρει σημαντικές κυρώσεις. Επιπροσθέτως, η εξαγωγή δεδομένων μπορεί να θεωρηθεί και αντιγραφή, γεγονός που σημαίνει πως μπορεί να οδηγήσει σε κατηγορίες για παραβίαση πνευματικών δικαιωμάτων. Το αν μια τέτοια αξίωση έχει κάποια αξία θα εξαρτηθεί από το συγκεκριμένο δεδομένου ότι δεν έχουν όλα τα δεδομένα που έχουν υποστεί αποξήρανση για την προστασία των δικαιωμάτων πνευματικής ιδιοκτησίας.

Εν συνεχεία, οι Web Scrapers σε πολλές περιπτώσεις, μπορεί να παραβιάσουν τα δικαιώματα περί βάσεων δεδομένων καθώς, όταν το σύνολο ή ένα σημαντικό τμήμα μιας βάσης δεδομένων εξάγεται ή επαναχρησιμοποιείται χωρίς τη συναίνεση του ιδιοκτήτη της, τότε αποτελεί αδίκημα. Η επανειλημμένη εξαγωγή ή επαναχρησιμοποίηση των επιμέρους τμημάτων μιας βάσης δεδομένων που έρχεται σε σύγκρουση με τη κανονική χρήση της βάσης ενδέχεται επίσης να παραβιάζει δικαιώματα βάσης δεδομένων. Παράβαση των δικαιωμάτων βάσης δεδομένων μπορεί επίσης να ισχύουν κατά την απόκρυψη καταλόγων ή καταλόγων ιστοσελίδων τρίτων εάν ο ιδιοκτήτης έχει πραγματοποιήσει δαπάνες για την ανάπτυξη και τη συντήρησή της.

Επιπροσθέτως, όσοι επιθυμούν να χρησιμοποιήσουν Web Scrapers για να συλλέξουν πληροφορίες από ιστοσελίδες που περιέχουν υλικό για φυσικά πρόσωπα (facebook, linkedin) πρέπει να γνωρίζουν ότι κινδυνεύουν να παραβιάσουν τη νομοθεσία γύρω από τα προσωπικά δεδομένα σε περίπτωση που αυτά δεν συναινούν ρητά με αυτή την ενέργεια.

2.5 Λόγοι και Σενάρια Χρήσης του Web Scraping

Όπως αναφέρθηκε και προηγουμένως, το Web scraping είναι μια ιδιαίτερη διαδικασία, από νομικής απόψεως, παρόλα αυτά αποτελεί μια διαδικασία αρκετά χρήσιμη ή ακόμα και σε αρκετές περιπτώσεις σημαντική. Αυτό συμβαίνει, διότι σε πολλές περιπτώσεις, αρκετά έργα προγράμματα ή ακόμα και εταιρίες στηρίζονται στην εξαγωγή και την επεξεργασία διαφόρων δεδομένων.

Για παράδειγμα, σε αρκετά ηλεκτρονικά καταστήματα (e-shops) διαφόρων ειδών, στον εκάστοτέ χρήστη κατά την αναζήτηση του επιθυμητού του προϊόντος, του δίνετε η δυνατότητα σύγκρισης τιμών του προϊόντος σε σχέση με ένα ίδιο προϊόν σε άλλα καταστήματα, που όπως είναι λογικό έχουν πιο ανεβασμένη τιμή σε σχέση με το κατάστημα στο οποίο βρίσκετε ο χρήστης. Η λειτουργία αυτή παρέχετε με τη βοήθεια των Web Scrapers, οι οποίοι αναζητούν σε συγκεκριμένα διαδικτυακά καταστήματα, τα οποία καθορίζονται από τον χρήστη, με σκοπό την εξαγωγή της πληροφορίας αναφορικά με την τιμή του προϊόντος. Κατά αυτόν τον τρόπο, το κατάστημα που παρέχει τη λειτουργία αυτή ευνοείται καθώς παρουσιάζει το φθηνότερο προϊόν σε σχέση με τους ανταγωνιστές του.

Μία άλλη περίπτωση χρήσης των Web Scrapers είναι η συλλογή δεδομένων γύρω από τα καιρικά φαινόμενα με σκοπό την ανάλυση τους και την εξαγωγή ασφαλούς συμπεράσματος γύρω από ένα δελτίο καιρού. Σχεδόν όλες οι εφαρμογές οι οποίες ασχολούνται με τα καιρικά φαινόμενα αντλούν τα δεδομένα τους χρησιμοποιώντας Web Crawlers για την άντληση πληροφοριών από το διαδίκτυο σχετικά με τα καιρικά φαινόμενα ανά τον κόσμο. Εκμεταλλεύονται δηλαδή, την ανάρτηση πληροφοριών από διάφορες ιστοσελίδες ή ακόμα και από ιστοσελίδες μεγάλων ειδησεογραφικών πρακτορείων σχετικά με τον καιρό και κατά αυτόν τον τρόπο παρουσιάζουν την έκβαση των καιρικών φαινομένων σχεδόν σε όλες τις χώρες και πόλεις ανά τον πλανήτη.

Όπως και στην προαναφερθείσα περίπτωση έτσι και σε πολλές άλλες περιπτώσεις, διάφορες εφαρμογές χρησιμοποιούν τους Web Scrapers, ώστε να συλλέξουν διάφορα δεδομένα από αρκετές ιστοσελίδες ή γενικότερα πηγές και να τα παρουσιάσουν στους χρήστες τους. Ένα από τα πιο κλασσικά παραδείγματα, είναι οι εφαρμογές που ασχολούνται με το στοίχημα γύρω από διάφορα αθλήματα. Οι εφαρμογές αυτές, συλλέγουν πληροφορίες γύρω από τις αποδώσεις μεγάλων στοιχηματικών εταιριών (π.χ. bwin, vistabet, κ.α.), καθώς επίσης και σημαντικές πληροφορίες αναφορικά με τον εκάστοτε αγώνα, όπως για παράδειγμα απώλειες λόγο τραυματισμών, βαθμολογική κατάταξη ή σε τί κατάσταση βρίσκεται η εκάστοτε ομάδα (σερί νικών, ηττών κ.α.).

Λαμβάνοντας υπόψιν τα παραπάνω παραδείγματα, μπορεί κανείς εύκολα να συμπεράνει πως οι Web Scrapers σε πολλές περιπτώσεις μπορούν να αποτελέσουν καθοριστικό παράγοντα για τη λειτουργία ενός οργανισμού ο οποίος θα πρέπει να αλληλοεπιδρά καθημερινά με τεράστιο όγκο πληροφοριών από όλο το διαδίκτυο. Τέτοιοι οργανισμοί μπορεί να είναι οι στοιχηματικές εταιρίες, ειδησιογραφία πρακτορεία ή ακόμα και διάφορα ηλεκτρονικά καταστήματα.

Έτσι λοιπόν ένας από τους σημαντικότερους λόγους χρήσης των Web Crawlers είναι η ταχύτητα που προσφέρουν στην άντληση και την επεξεργασία των

πληροφοριών. Για παράδειγμα, ένα ηλεκτρονικό κατάστημα θα μπορούσε να χρησιμοποιήσει Web Crawlers για τη δημιουργία περιγραφών των προϊόντων του, ή ακόμα και για λήψη διαφόρων φωτογραφιών σχετικά με αυτό, απλουστεύοντας την όλη διαδικασία.

Επιπλέον, οι Web Scrapers μπορούν να βοηθήσουν στη συλλογή πληροφοριών γύρω από το χρηματιστήριο και γενικότερα με την οικονομική κατάσταση διαφόρων εταιριών, καθώς στον κύκλο των οικονομολόγων και των οικονομικών αναλυτών υπάρχει η ανάγκη για εργαλεία τα οποία συλλέγουν διάφορες πληροφορίες σχετικά με όσα αναφέρθηκαν προηγουμένως σε σύντομο χρονικό διάστημα, έτσι ώστε να μπορέσουν να συμβουλέψουν αναλόγως τους πελάτες τους σχετικά με επενδύσεις στις ανάλογες εταιρίες.

Πολλή σημαντική επίσης θα μπορούσε να χαρακτηριστεί η συνεισφορά των Web Scrapers στην ανακάλυψη ή ακόμα και στην αντιμετώπιση των fake news, τα οποία αποτελούν μάστιγα του διαδικτύου, αφού καθημερινά πλέον αυτό κατακλύζετε από χιλιάδες fake news διαφόρων ειδών. Με τη χρήση των Web Scrapers, θα μπορούσε να δημιουργηθεί ένα εργαλείο το οποί εντοπίζει τα πιθανά fake news και στη συνέχεια πραγματοποιώντας μια έξυπνη αναζήτηση σε έμπιστα ειδησεογραφικά πρακτορεία να διασταυρώσει ή να χαρακτηρίσει την είδηση ως fake news.

Ένας άλλος τομέας ο οποίος χρησιμοποιεί και ωφελείται μέσα από Web Scrapers είναι αυτός της προβλεπτικής ανάλυσης, καθώς σε αυτό τον κλάδο η ύπαρξη και η καθημερινή ανανέωση των δεδομένων αποτελεί βασικό χαρακτηριστικό της λειτουργίας του. Με τη βοήθεια των Web Scrapers μπορούν να συλλεχθούν και να επεξεργαστούν αρκετά δεδομένα σε σύντομο χρονικό διάστημα, ώστε να μπορέσει να εξαχθεί ένα συμπέρασμα γύρω από αυτά.

Εκτός από όλα τα παραπάνω, η τεχνική του Web Scraping προσφέρει τα οφέλη του σε μία ταχέως αναπτυσσόμενη επιστήμη, αυτή της μηχανικής μάθησης (machine learning). Η μηχανική μάθηση, απαιτεί τεράστιες συλλογές δεδομένων, προκειμένου να δημιουργηθούν διάφορα μοντέλα τα οποία στη συνέχεια χρησιμοποιούν οι αλγόριθμοι ώστε να εκτελέσουν διάφορες ενέργειες. Έτσι λοιπόν, με τη χρήση των Web Scrapers είναι εφικτή όχι μόνο η συλλογή των απαραίτητων δεδομένων αλλά και ο συνεχής εμπλουτισμός τους ή η δημιουργία και άλλων σετ δεδομένων.

Επομένως, μπορεί κανείς εύκολα να συμπεράνει πως τα οφέλη των Web Scrapers είναι αναρίθμητα, καθώς συμβάλλουν στην απλούστευση αρκετών διαδικασιών σε πάρα πολλούς τομείς, διευκολύνοντας έτσι το έργο και τη λειτουργία τους όπως επίσης συμβάλλουν και στην ανάπτυξή τους.

2.6 Αρνητικά του Web Scraping

Όπως αναφέρθηκε και παραπάνω, το Web Scraping μπορεί να αποτελεί μία πολλή χρήσιμη τεχνική, παρόλα αυτά όμως σε πολλές περιπτώσεις δεν επιτρέπετε από αρκετές ιστοσελίδες. Αυτό γίνετε συνήθως για τη διαφύλαξη των δεδομένων, αφού συνήθως οι περισσότερες ιστοσελίδες δεν επιτρέπουν την αντιγραφή, την επεξεργασία ή την αναπαραγωγή των δεδομένων τους, με σκοπό τη διαφύλαξή τους. Εξάλλου, στη σύγχρονη εποχή η πληροφορία θεωρείται ως το πολυτιμότερο και πιο ακριβό αγαθό, καθώς αποτελεί κινητήριο δύναμη για τους περισσότερους κλάδους.

Επομένως, μπορεί κανείς εύκολα να συμπεράνει πως δεν είναι λίγες οι φορές οι οποίες το Web Scraping δεν μπορεί να εξάγει την πληροφορία, καθώς συνήθως οι ιστοσελίδες που δεν το επιτρέπουν έχουν διάφορους μηχανισμούς εντοπισμού και αντιμετώπισής του. Όπως είναι γνωστό, προκειμένου οι Web Scrapers να εξάγουν την πληροφορία, πραγματοποιούν HTTP αιτήματα (requests) προς την ιστοσελίδα, ώστε να πραγματοποιήσουν αρχικά λήψη της ιστοσελίδας και στη συνέχεια εξαγωγή των επιθυμητό δεδομένων. Με βάση τον όγκο των αιτημάτων οι ιστοσελίδες μπορεί να αποκλείσουν τη διεύθυνση ip από την οποία πραγματοποιούνται τα αιτήματα αυτά και ως αποτέλεσμα μπλοκάρουν τη λειτουργία των Scrapers ή ακόμα και τη δυνατότητα επίσκεψης της ιστοσελίδας. Κατά αυτόν τον τρόπο οι Scrapers αδυνατούν να πραγματοποιήσουν οποιαδήποτε λειτουργία πάνω στην ιστοσελίδα μέχρις ότου παρέλθει το χρονικό διάστημα για το οποίο αποκλείεται η διεύθυνση ip.

Ένα άλλο αρνητικό το οποίο μπορεί να συναντήσουν οι Web Scrapers κατά τη λειτουργία τους, είναι η αλλαγή του πηγαίου κώδικα της ιστοσελίδας. Όπως αναφέρθηκε προηγουμένως, οι Scrapers πραγματοποιούν λήψη της ιστοσελίδας και στη συνέχεια αναζητούν την επιθυμητή πληροφορία μέσα από τον πηγαίο κώδικά της. Επομένως, οποιαδήποτε αλλαγή στον πηγαίο κώδικα της ιστοσελίδας, μπορεί να διακόψει την αποτελεσματικότητα του Web Scraper στην περίπτωση που δεν ενημερωθεί από τον εκάστοτε developer αναλόγως. Έτσι λοιπόν, γεννάται η ανάγκη για συνεχή παρακολούθηση και ενδεχομένως και ενημέρωση των Web Scrapers, ώστε να μπορούν να είναι λειτουργικοί και αποδοτικοί ακόμα και σε ενδεχόμενες αλλαγές στον κώδικα της εκάστοτε ιστοσελίδας.

Τέλος, ένα άλλο αρνητικό των Web Scrapers είναι η διαθεσιμότητα της ιστοσελίδας την οποί προσπαθούν να επεξεργαστούν. Σε πολλές περιπτώσεις, διάφορες ιστοσελίδες μπορεί να παρουσιάσουν τεράστιο αριθμό επισκεψιμότητας και έτσι να είναι περιορισμένη η διαθεσιμότητά τους. Σε αυτή τη περίπτωση, ο Scraper θα χρειαστεί αρκετή ώρα προκειμένου να επεξεργαστεί και να εξάγει τα επιθυμητά δεδομένα. Επομένως, μπορεί κανείς να οδηγηθεί στο συμπέρασμα πως οι Scrapers εξαρτώνται σε πολύ μεγάλο βαθμό από την ίδια την ιστοσελίδα την οποία θα επεξεργαστούν.

Κεφάλαιο 3ο

Scrapy Python Framework

3.1 Τί είναι το Scrapy Framework

Το Scrapy είναι ένα Framework της γλώσσας προγραμματισμού python, το οποίο χρησιμοποιείται προκειμένου να αυτοματοποιηθεί και να πραγματοποιηθεί η διαδικασία του web scraping διάφορων ιστοσελίδων με σκοπό την εξαγωγή δομημένων δεδομένων τα οποία μπορούν εν συνεχεία, να επεξεργαστούν και να χρησιμοποιηθούν για διάφορους σκοπούς.

Επιπλέον το Scrapy Framework, μπορεί να χρησιμοποιηθεί είτε με σκοπό να εξάγει δεδομένα από διάφορες ιστοσελίδες ως ένας Web Scraper είτε χρησιμοποιώντας APIs όπως τα Amazon Web Services σύμφωνα με την επίσημη ιστοσελίδα του Scrapy Framework (https://docs.scrapy.org/en/latest/intro/overview.html).

Αξίζει επιπλέον να σημειωθεί, πως το Scrapy Framework παρέχει αρκετές λειτουργίες το οποίο το καθιστούν αρκετά εύχρηστο και αποτελεσματικό για οποιονδήποτε χρήστη. Πιο συγκεκριμένα, παρέχετε η δυνατότητα για επιλογή και εξαγωγή συγκεκριμένων δεδομένων χρησιμοποιώντας css-selector καθώς και xpath expressions. Για το σκοπό αυτό παρέχετε η δυνατότητα ενός interactive shell, μέσα από το οποίο ο χρήστης μπορεί να αλληλοεπιδράσει με την ιστοσελίδα, ώστε να δοκιμάσει τις προαναφερθείσες εκφράσεις και να δει εκείνη τη στιγμή το αποτέλεσμά τους.

Η πιο σημαντική λειτουργία που παρέχετε με χρήση του Scrapy Framework, είναι η διαχείριση διαφόρων περιπτώσεων όπως αυτή της απαγόρευσης των Web Scrapers σύμφωνα με τα αρχεία robots.txt τα οποία υπάρχουν σχεδόν σε κάθε ιστοσελίδα. Αναλυτικότερα, ο χρήστης μπορεί να καθορίσει αν ο Web Scraper θα παρακάμψει το αρχείο robots.txt αγνοώντας το ή όχι προκειμένου πραγματοποιήσει την εξαγωγή δεδομένων. Επιπροσθέτως, ο χρήστης μπορεί να αλλάξει τον user-agent που θα χρησιμοποιηθεί κατά το αίτημά του ώστε να πραγματοποιήσει την όποια επεξεργασία, καθώς συνήθως το default όνομα μπορεί να αποκλειστεί και έτσι να εμποδιστεί η λειτουργία του Scraper.

Εκτός από τα παραπάνω, το Scrapy Framework παρέχει τη δυνατότητα αυτοματοποιημένης διαδικασίας προκειμένου να πραγματοποιείτε λήψη εικόνων που ανακαλύπτονται κατά τη διαδικασία του scrapping, εξοικονομώντας έτσι χρόνο καθώς επίσης και πολυπλοκότητα κατά την υλοποίηση.

Τέλος, μία ιδιαίτερα σημαντική λειτουργία αποτελούν και τα Web Spiders, τα οποία θα αναλυθούν παρακάτω σχετικά με το τί είναι και πως ακριβώς λειτουργούν.

3.2 Web Spiders

Όπως έχει αναφερθεί και προηγουμένως, το Scrapy Framework παρέχει αρκετές λειτουργίες στους χρήστες του με σκοπό να βοηθήσει και να αυτοματοποιήσει τη διαδικασία του Web Scraping. Μία από αυτές τις λειτουργίες είναι και οι Web Spiders. Πιο συγκεκριμένα, οι Web Spiders μέσα από το Scrapy Framework, ορίζονται ως κολάσεις μέσα στις οποίες καθορίζετε ποιες ιστοσελίδες ή ένα σετ ιστοσελίδων θα επεξεργασθούν με σκοπό την εξαγωγή της πληροφορίας, καθώς επίσης και τον τρόπο με τον οποίο αυτή η ενέργεια θα πραγματοποιηθεί.

Αναλυτικότερα, μέσα στα Web Spiders μπορούν να οριστούν οι 'κανόνες', δηλαδή σύμφωνα με ποια κριτήρια θα καθοριστεί η επιθυμητή προς εξαγωγή πληροφορία. Αυτό επιτυγχάνετε με χρήση των εκφράσεων css-selectors και xpath expressions.

Προκειμένου να δημιουργηθεί ένα Web Spider, είναι αναγκαία η αρχικοποίηση των αιτημάτων που θα χρησιμοποιηθούν για τη συνέχεια της διαδικασίας. Αυτό επιτυγχάνετε με αυτοματοποιημένο τρόπο μέσα από το Scrapy Framework, καθώς ο χρήστης καλείται να δηλώσει κάποια URLs και στη συνέχεια με χρήση της μεθόδου start_requests() (η όλη διαδικασία θα δειχθεί με παραδείγματα) παράγονται τα απαιτούμενα για τη διαδικασία αιτήματα. Στη συνέχεια, με χρήση της μεθόδου parse(), πραγματοποιείται προσπέλαση της ιστοσελίδας, ώστε να εφαρμοστούν οι κανόνες που προαναφέρθηκαν για την εξαγωγή των δεδομένων. Σε αυτό το σημείο αξίζει να σημειωθεί πως παρέχεται η δυνατότητα εφαρμογής και άλλων Framework όπως το BeautifulSoup ή lxml για την εξαγωγή των δεδομένων.

Τελικά, πραγματοποιείται η εξαγωγή των δεδομένων, τα οποία με τη σειρά τους καταχωρούνται σε βάση δεδομένων ή σε τοπικό αρχείο μέσα στον υπολογιστή. Επιπροσθέτως, είναι σημαντικό να αναφερθεί πως ο σκοπός των Web Spiders είναι η απλούστευση της διαδικασίας, αλλά και η μείωση της πολυπλοκότητας της υλοποίησης. Αυτό συμβαίνει, διότι η λογική των Web Spiders βασίζετε στη λογική του Object Oriented Porgramming. Πιο συγκεκριμένα, παρέχεται η δυνατότητα επαναχρησιμοποίησης του κώδικα και των αντικειμένων, με σκοπό τη βελτίωση της αποδοτικότητας του λογισμικού, καθώς και η γρηγορότερη υλοποίηση επιπλέον λειτουργιών ώστε να μπορούν ταυτόχρονα να λειτουργούν πολυάριθμα Web Spiders με σκοπό την εξαγωγή τεράστιου όγκου δεδομένων από διάφορες ιστοσελίδες.

Τέλος αξίζει να σημειωθεί πως σε αρκετές περιπτώσεις ενδέχεται να μπορούν να επαναχρησιμοποιηθούν και οι κανόνες που ορίζονται μέσα στα Web Spiders, καθώς σε αρκετές ιστοσελίδες ίσως μπορούν να εφαρμοστούν οι εκφράσεις που έχουν αναφερθεί παραπάνω ώστε να εξαχθούν τα επιθυμητά δεδομένα. Εκτός από αυτά, το Scrapy Framework παρέχει αρκετά default Web Spiders, τα οποία μπορούν να χρησιμοποιηθούν χωρίς παραλλαγές ή ενδεχομένως να μπορεί να αναπτυχθεί κώδικας πάνω σε αυτές με σκοπό την διευκόλυνση των developers.

3.3 Αντικείμενα

Τα αντικείμενα αποτελούν μια αρκετή σημαντική λειτουργία του Scrappy Framework, καθώς ορίζοντας αντικείμενα, ο χρήστης έχει τη δυνατότητα να δομήσει αυτόματα το τελικό αρχείο το οποίο όπως έχει αναφερθεί και προηγουμένως μπορεί να είναι σε μορφή csv ή json. Με αυτό τον τρόπο και ιδιαίτερα σε περιπτώσεις όπου ο όγκος δεδομένων είναι τεράστιος ο χρήστης έχει τη δυνατότητα να δημιουργήσει αυτόματα τις 'κολόνες' σε ένα αρχείο csv για παράδειγμα. Έτσι, τα τελικά δεδομένα δομούνται χωρίς ο χρήστης να μπορεί να κάνει κάποιο λάθος ενδεχόμενος κατά την κατηγοριοποίησή τους.

Προκειμένου ένας χρήστης να δημιουργήσει ένα αντικείμενο, χρειάζεται να ανοίξει το σχετικό αρχείο (items.py) και στη συνέχεια δημιουργώντας κλάσεις να δημιουργήσει τα αντικείμενα μέσα στα οποία θα δηλώσει τα πεδία και την ονομασία που θα τους δοθεί. Επιπλέον, με αυτόν τον τρόπο μπορούν να οριστούν και μεταδεδομένα (metadata) με τη βοήθεια των αντικειμένων μέσα από το Scrapy Framework. Έτσι παρέχεται η δυνατότητα στον χρήστη να εξάγει και να καταχωρήσει όχι μόνο δεδομένα αλλά και αρκετά μεγάλο όγκο χρήσιμων, για τα δεδομένα, πληροφοριών. Σημαντικό είναι επίσης να αναφερθεί, πως δεν υπάρχουν συγκεκριμένες λέξεις κλειδιά για τον ορισμό των μεταδεδομένων, γεγονός που σημαίνει πως ο εκάστοτε χρήστης μπορεί να προσθέσει τα δικά του επιθυμητά μεταδεδομένα για κάθε κατηγορία δεδομένων που εξάγονται.

Με βάση τα όσα αναφέρονται παραπάνω, μπορεί κανείς εύκολα να συμπεράνει πως το Scrapy Framework αποτελεί ένα extensible εργαλείο μέσα από το οποίο παρέχονται αρκετές ανέσεις στον χρήστη, αφού αυτοματοποιεί αρκετές χρονοβόρες διαδικασίες. Συγκεκριμένα όσο αναφορά τα αντικείμενα, προσφέρει τη δυνατότητα ομαλής κατηγοριοποίησης των δεδομένων που εξάγονται, χωρίς να χεριάζετε να σπαταληθεί επιπλέον χρόνος για ομαδοποίηση των δεδομένων. Επιπροσθέτως, τα αντικείμενα δημιουργούνται μέσα από κλάσεις, γεγονός που τα καθιστά επαναχρησιμοποιήσιμα, μειώνοντας έτσι το χρόνο και τη πολυπλοκότητα ανάπτυξης κώδικα.

```
>>> product = Product(name='Desktop PC', price=1000)
>>> print(product)
Product(name='Desktop PC', price=1000)
```

Εικόνα 7 Παράδειγμα δημιουργίας αντικειμένου πηγή: https://docs.scrapy.org/en/latest/topics/items.html

3.4 Logging and Stats Collection

Το Scrapy Framework, χρησιμοποιεί logging system βασισμένο στη γλώσσα προγραμματισμού python. Η λειτουργία αυτή μπορεί να βοηθήσει αρκετά στη διαδικασία τόσο της ανάπτυξης του κώδικα όσο και στη διαδικασία του debugging. Επιπλέον, το python based logging system είναι αρκετά ευκολότερο στην ανάγνωση και την κατανόηση από τον εκάστοτε χρήστη και έτσι μπορεί να γίνει η όλη διαδικασία γρηγορότερα και αποτελεσματικότερα.

Τα επίπεδα logging είναι τα εξής:

1. logging.CRITICAL - for critical errors (highest severity)
2. logging.ERROR - for regular errors
3. logging.WARNING - for warning messages
4. logging.INFO - for informational messages
5. logging.DEBUG - for debugging messages (lowest severity)

Προκειμένου να χρησιμοποιηθεί το logging system είναι αναγκαία η εισαγωγή της βιβλιοθήκης logging χρησιμοποιόντας την εντολή import logging και στη συνέχεια με χρήση της εντολής logging.log(logging.WARNING, "This is my warning"). Τέλος, ο χρήστης έχει τη δυνατότητα να τροποποιήσει τις ρυθμίσεις των loggers.

Επιπροσθέτως, το Scrapy Framework παρέχει τη δυνατότητα για συλλογή στατιστικών, επιστρέφωντας τα δεδομένα σε μορφή κλειδί, τιμή. Αυτή η λειτουργία μπορεί να αποδειχθεί ιδιαιτέρως χρήσιμη, καθώς μπορεί για παράδειγμα ο χρήστης να ορίσει να αποθηκεύει μετρητές για τα δεδομένα που συλλέγει, ώστε να μπορέσει στη συνέχεια να εξάγει για αυτά ορισμένα συμπεράσματα. Οι διαθέσιμοι συλλογείς δεδομένων είναι δύο:

1) MemoryStatsCollector: Κρατάει στη προσωρινή μνήμη τα στατιστικά στοιχεία της λειτουργίας της εκάστοτε Spider. Αυτός είναι ο προεπιλεγμένος συλλέκτης στατιστικών στοιχείων που χρησιμοποιείται στο Scrapy.

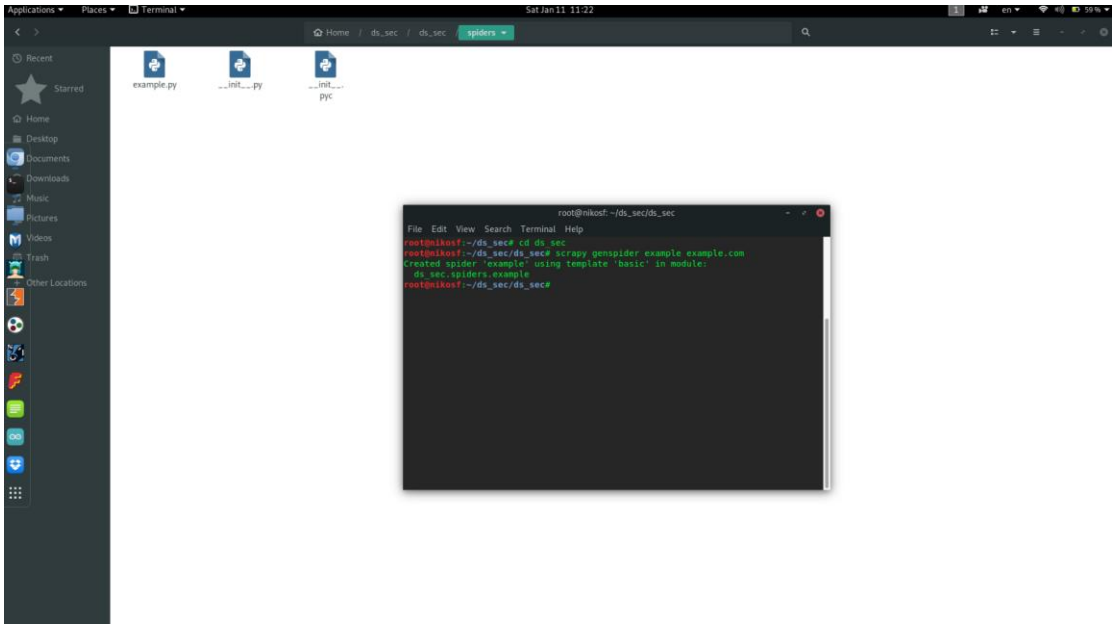
2) DummyStatsCollector: Ένας συλλέκτης Στατιστικών που δεν κάνει τίποτα αλλά είναι πολύ αποδοτικός (επειδή δεν κάνει τίποτα). Αυτός ο συλλέκτης στατιστικών στοιχείων μπορεί να οριστεί μέσω της ρύθμισης STATS_CLASS, για να απενεργοποιηθεί η συλλογή στατιστικών στοιχείων για τη βελτίωση της απόδοσης

3.5 Δημιουργία και Ανάπτυξη Αράχνης

Όπως έχει αναφερθεί και παραπάνω, οι αράχνες είναι εκείνες που αναλαμβάνουν τη διαδικασία να ψάξουν στις ιστοσελίδες που ορίζει ο χρήστης προκειμένου να μπορέσουν να συλλέξουν τα δεδομένα. Επιπλέον, παρέχουν τη δυνατότητα για συλλογή δεδομένων από διάφορες ιστοσελίδες ταυτόχρονα προκειμένου να μειωθεί η δυσκολία στην ανάπτυξή τους αλλά και για να εξυπηρετήσουν όσο το δυνατόν περισσότερες απαιτήσεις του χρήστη.

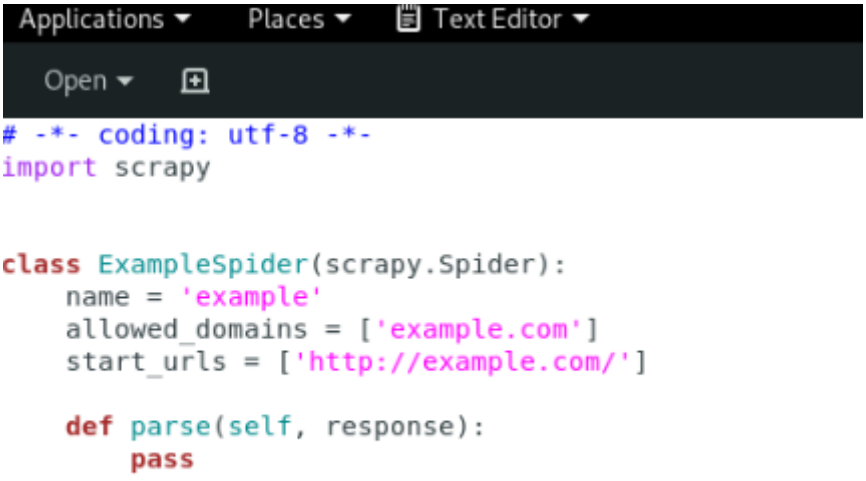
Έτσι λοιπόν μπορεί κανείς να συμπεράνει πως η δημιουργία και η ανάπτυξη μιας αράχνης είναι η σημαντικότερη διαδικασία προκειμένου να χρησιμοποιήσει κανείς αποτελεσματικά το Scrapy Framework.

Προκειμένου να δημιουργήσει ένας χρήστης αράχνες χρειάζετε να πληκτρολογήσει την εντολή scrapy genspider spider example, η οποία δημιουργεί το αρχείο example.py μέσα στον φάκελο spiders.



Εικόνα 8 Δημιουργία Αράχνης

Μέσα στο αρχείο υπάρχει ο παρακάτω κώδικας.



Εικόνα 9 Παράδειγμα πηγαίου κώδικα αράχνης

Στη παραπάνω εικόνα φαίνονται τη default κλάση και μέθοδο που δημιουργούνται με την εκτέλεση της προηγούμενης εντολής. Πιο συγκεκριμένα, κάθε κλάση που δημιουργείται είναι υποχρεωτικό να την κληρονομεί, διότι περιέχει την υλοποίηση της μεθόδου start_requests() το οποίο στέλνει αιτήματα από το χαρακτηριστικό start_urls spider και καλεί την ανάλυση της μεθόδου αράχνης για κάθε μία από τις απαντήσεις. Μέσα στη κλάση ExampleSpider του παραπάνω παραδείγματος, μπορούν να δηλωθούν αρκετές παράμετροι, μία από τις οποίες είναι η παράμετρος name. Κατά τη δήλωση της παραμέτρου αυτής ορίζεται από τον χρήστη το όνομα της, το οποίο θα πρέπει να είναι αυτό που θα πρέπει να χρησιμοποιήσει κατά την κλήση της αράχνης (π.χ. scrapy crawl example) άσχετα με το όνομα που έχει το αρχείο py. Για αυτό το λόγο, το όνομα που θα δηλώσει ο χρήστης θα πρέπει να είναι μοναδικό.

Η επόμενη παράμετρος που καλείται να παραμετροποιήσει ο χρήστης είναι η start_urls, η οποία αποτελεί μία python list, στην οποία ο χρήστης δηλώνει τα url των ιστοσελίδων που επιθυμεί. Αξίζει να σημειωθεί πως ο χρήστης μπορεί να δώσει από ένα μέχρι όσα ορίσματα επιθυμεί προκειμένου να αντληθούν δεδομένα από διάφορα Urls. Σημαντικό είναι να αναφερθεί πως καλό θα ήταν να δηλωθούν Urls της ίδιας ιστοσελίδας προκειμένου να μπορούν να αντληθούν δεδομένα, καθώς αναλόγως με ποιους τρόπους θα επιλέξει ο χρήστης να δομήσει τα δεδομένα που θα εξάγει (xpath, css selectors) ενδέχεται τα html elements να διαφέρουν από ιστοσελίδα σε ιστοσελίδα. Για αυτό τον σκοπό υπάρχει και η παράμετρος allowed_domain, η οποία καθορίζει ποιες ιστοσελίδες επιτρέπετε να εξερευνηθούν από την αράχνη. Η ύπαρξη της παραμέτρου αυτής είναι να μπορεί να εξυπηρετήσει τον χρήστη εάν επιθυμεί να εναλλάσσει τις ιστοσελίδες χωρίς να χεριάζετε να αλλάξει την παράμετρο start_urls.

```python
import scrapy


class MySpider(scrapy.Spider):
    name = 'example.com'
    allowed_domains = ['example.com']
    start_urls = [
        'http://www.example.com/1.html',
        'http://www.example.com/2.html',
        'http://www.example.com/3.html',
    ]

    def parse(self, response):
        self.logger.info('A response from %s just arrived!', response.url)
```

Εικόνα 10 πηγή https://docs.scrapy.org/en/latest/topics/spiders.html

Επιπροσθέτως, μια πολύ σημαντική συνάρτηση είναι η parse(self, response), η οποία δέχεται σαν ορίσματα τον εαυτό της και την απάντηση από την ιστοσελίδα. Πιο συγκεκριμένα, η συνάρτηση αυτή είναι υπεύθυνη για την επεξεργασία της απάντησης που λαμβάνει από την επιθυμητή ιστοσελίδα προκειμένου να εξάγει τα επιθυμητά δεδομένα. Για αυτό το λόγο η συνάρτηση αυτή καλείται (by default) σε κάθε περίπτωση που υπάρχει απάντηση από την ιστοσελίδα και μέσα σε αυτή τη συνάρτηση ο χρήστης μπορεί να ορίσει τον τρόπο με τον οποίο θα εξάγει τα δεδομένα που επιθυμεί να κρατήσει. Αναλυτικότερα, μέσα στη συνάρτηση parse μπορούν να δημιουργηθούν μέθοδοι με χρήση των οποίων η αράχνη θα κρατάει ή θα αγνοεί δεδομένα. Επιπλέον, μπορεί να οριστεί μέσα σε αυτή και σε ποια αντικείμενα θα αποθηκεύονται (τα αντικείμενα που αναφέρθηκαν παραπάνω).

Εκτός από τα παραπάνω, το Scrapy Framework παρέχει generic αράχνες, προκειμένου να εξυπηρετήσει τις ανάγκες του εκάστοτε χρήστη. Αυτές αφορούν σε γενικού τύπου περιπτώσεις και προβλήματα, όπως για παράδειγμα τη συλλογή δεδομένων από έναν ισότοπο ακολουθώντας συγκεκριμένους συνδέσμους αλλά και κανόνες που μπορεί να ορίσει ο χρήστης. Οι κανόνες αυτοί δίνονται σαν ορίσματα στην κλάση που αναφέρθηκε προηγουμένως. Οι κανόνες αυτοί αφορούν σε διάφορες λειτουργίες της αράχνης όπως για παράδειγμα αν θα υπάρχει κάποια callback συνάρτηση ή το ποια ορίσματα θα πρέπει να δέχεται η συνάρτηση αυτή. Επιπλέον, καθορίζετε αν θα πραγματοποιείται επεξεργασία των συνδέσμων των οποίων η αράχνη συλλέγει ή το αν θα καλείται η μέθοδος process_request η οποί καθορίζει που θα καλείται για κάθε Αίτημα που εξάγεται από αυτόν τον κανόνα. Αυτό το καλούμενο θα έπρεπε να πάρει το αίτημα ως πρώτο επιχείρημα και η απάντηση από την οποία προέκυψε το αίτημα ως το δεύτερο επιχείρημα. Πρέπει να επιστρέψει ένα αντικείμενο αίτησης ή κανένα (για να φιλτράρει το αίτημα).

(πηγη: https://docs.scrapy.org/en/latest/topics/spiders.html)

Κεφάλαιο 4ο

Σκέψη και Τρόπος Υλοποίησης

4.1 Συνολική Ιδέα

Όπως αναφέρθηκε και προηγουμένως, σκοπός της δημιουργίας του εργαλείου αυτού είναι η αναζήτηση σε ανοιχτές βάσεις δεδομένων προκειμένου να ανακαλυφθούν, εφόσον υπάρχουν αδυναμίες σε εγκατεστημένες εφαρμογές του προσωπικού υπολογιστή. Αυτό επιτυγχάνετε, με τη χρήση του Framework Scrapy το οποίο παρέχει τη δυνατότητα για συλλογή δεδομένων από διάφορες πηγές που καθορίζονται από τον χρήστη, με χρήση css selectors ή xpath εκφράσεων.

Αναλυτικότερα, ο agent που υλοποιήθηκε, ξεκινάει τη λειτουργία του χρησιμοποιώντας το αρχείο start.py μέσα από το οποίο δημιουργούνται αρχικά τα αρχεία .csv που θα χρησιμοποιηθούν ώστε να αποθηκευτούν τα δεδομένα που συλλέγονται από τις αράχνες. Ο λόγος για τον οποίο δημιουργούνται εξαρχής τα προαναφερθέντα αρχεία είναι προκειμένου να αποφευχθεί ένα σφάλμα, κατά το οποίο οι αράχνες αν δεν βλέπουν τα αρχεία στον φάκελο τότε δεν εκτελούνται.

Αφότου πραγματοποιηθεί η παραπάνω διαδικασία, τότε καλείται η πρώτη αράχνη, αυτή που συλλέγει δεδομένα από τη βάση δεδομένων rapid7. Σε αυτό το αρχείο, καλείται η εντολή dpkg, ώστε να ληφθούν όλα τα εγκατεστημένα πακέτα, τα οποία στη συνέχεια τοποθετούνται με τη βοήθεια μιας επανάληψης στα start_urls ώστε να ξεκινήσει η διαδικασία αναζήτησής τους μέσα στη βάση δεδομένων. Για όσα πακέτα βρίσκονται αδυναμίες, αυτά αποθηκεύονται μέσα στο αρχείο rapid7.csv. Αφότου, πραγματοποιηθεί η διαδικασία για όλα τα πακέτα τότε τερματίζετε και στη συνέχεια επιστρέφει πίσω στο αρχικό αρχείο start.py, το οποίο στη συνέχεια καλεί το αρχείο search.py.

Το αρχείο search.py είναι υπεύθυνο για τη λήψη αρχικά του files_exploits.csv μέσα από το repository της offensive security και περιέχει όλες τις ευπάθειες που έχουν καταγραφεί και καταγράφονται καθημερινά, καθώς το αρχείο αυτό ανανεώνετε κάθε μέρα. Επομένως, με τη λήψη του αρχείου αυτού, το οποίο παρέχει όλα τα δεδομένα που μπορεί κανείς να δει στην exploitdb, μπορεί να αναζητηθούν οι ευπάθειες για τα εγκατεστημένα πακέτα και ταυτόχρονα να μειωθεί ο χρόνος της αναζήτησης διότι πρόκειται για τοπικό αρχείο και όχι ιστοσελίδα. Επομένως με την εκτέλεση του αρχείου search.py αρχικά πραγματοποιείται λήψη του αρχείου csv που αναφέρθηκε και στη συνέχεια πραγματοποιείται η αναζήτηση μέσα σε αυτό. Με το πέρας της αναζήτησης στο αρχείο files_exploits.csv, πραγματοποιείται φιλτράρισμα των αποτελεσμάτων από το αρχείο rapid7.csv, ώστε όσες ευπάθειες βρεθούν να καταγραφούν συνολικά και στο τοπικό αρχείο agent.log.

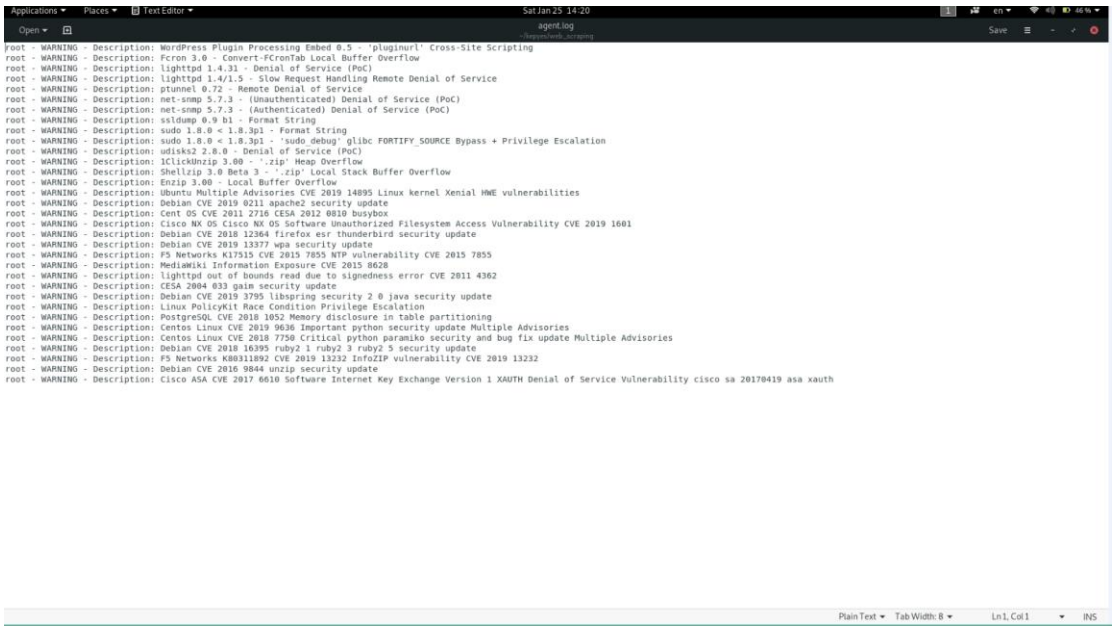
Μετά την εκτέλεση του προαναφερθέντος αρχείου, ο agent επιστρέφει στο αρχείο start.py το οποίο με τη σειρά του καλεί την αράχνη get cves. Αυτή είναι υπεύθυνη, ώστε να πάρει τα ονόματα των ευπαθειών που κατεγράφησαν στο αρχείο

agent.log προκειμένου να αναζητηθούν στη βάση δεδομένων cve-details οι κωδικοί CVE, αφού στη συνέχεια θα χρησιμοποιηθούν για να αναζητηθούν κάποια επιπλέον στοιχεία. Η αναζήτηση πραγματοποιείται όπως σε όλες τις αράχνες, έτσι όσα αποτελέσματα βρίσκονται για τις αδυναμίες που κατεγράφησαν, αποθηκεύονται στο αρχείο get_cve.csv. Κατά αυτόν τον τρόπο ολοκληρώνεται η διαδικασία της συλλογής και των κωδικών CVE.

Η τελευταία αράχνη η οποία καλείται είναι η details, η οποία είναι υπεύθυνη προκειμένου να συλλέξει δεδομένα, γύρω από τις ευπάθειες που έχουν καταγραφεί και για όσα cve έχουν επίσης καταγραφεί. Αυτό υλοποιείται πραγματοποιώντας στην βάση δεδομένων cve-details αναζήτηση με βάση του κωδικούς CVE που αναφέρθηκαν παραπάνω. Πιο συγκεκριμένα, με τη χρήση της αράχνης αυτής συλλέγονται οι βαθμοί επικινδυνότητας των ευπαθειών με σκοπό να κατηγοριοποιηθούν στη συνέχεια.

Αφότου πραγματοποιηθεί και η εκτέλεση της τελευταίας αράχνης, τότε καλείται το αρχείο create_Assesment.py, το οποίο αναλαμβάνει να εξερευνήσει τα .csv αρχεία στα οποία έχουν αποθηκευτεί οι ευπάθειες, ώστε να ομαδοποιήσει τα δεδομένα σε κατηγορίες όπως xss, code execution κ.α. Στη συνέχεια δημιουργούνται δύο γραφήματα, ένα barchart και ένα piechart με τα ομαδοποιημένα στοιχεία ανάλογα τη κατηγορία στην οποία ανήκουν. Τέλος, δημιουργείται ένα τυπικό assessment μέσα στο οποίο συμπεριλαμβάνονται οι εικόνες .png με τα γραφήματα που δημιουργήθηκαν κατά τη προηγούμενη διαδικασία.

Με αυτόν τον τρόπο ολοκληρώνεται η όλη διαδικασία η οποία δίνει στον χρήστη μία συνοπτική εικόνα με τις πιθανές ευπάθειες που μπορεί να υπάρχουν στο σύστημά του, παρέχοντάς του τη δυνατότητα να δει ποιες είναι οι ευάλωτες εφαρμογές ώστε στη συνέχεια να πραγματοποιήσει την ερεύνα του και να αντιμετωπίσει τα όποια προβλήματα. Αξίζει να σημειωθεί πως είναι αναγκαία η περαιτέρω διερεύνηση διότι σε πολλές περιπτώσεις μπορεί να μην υφίσταται λόγος για ανησυχία.



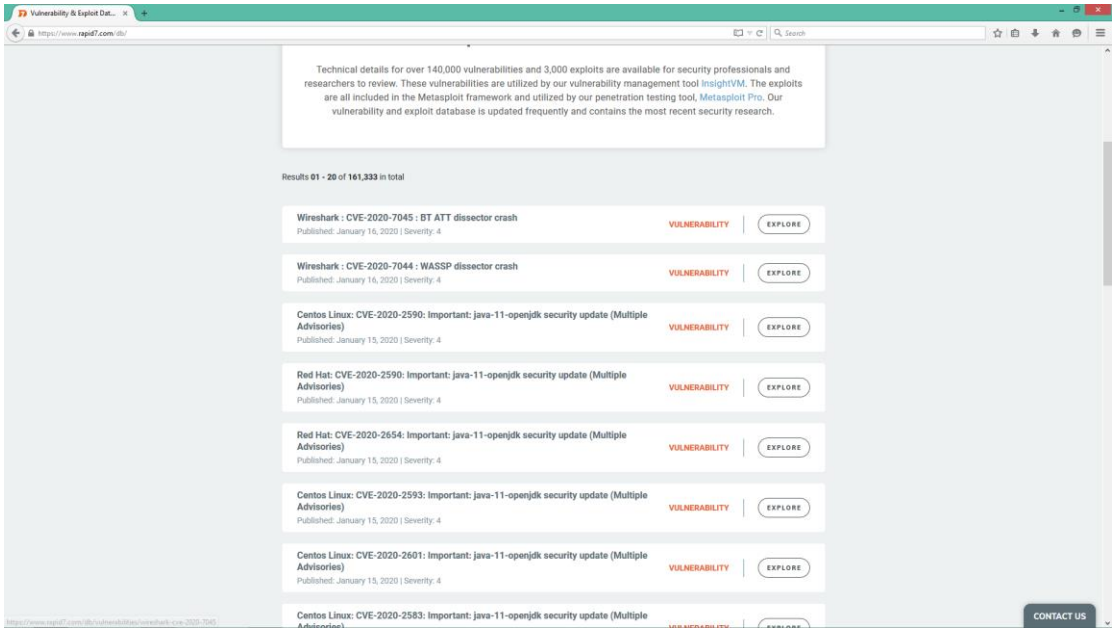
Εικόνα 11 Agent.log αρχείο

4.2 Διαδικασία Εξαγωγής των Δεδομένων

Όπως αναφέρθηκε και προηγουμένως, στόχος της δημιουργίας του agent είναι η εξαγωγή πληροφοριών γύρω από ευπάθειες που ενδεχομένως να υπάρχουν στις εγκατεστημένες εφαρμογές σε έναν υπολογιστή. Προκειμένου να πραγματοποιηθεί εξόρυξη δεδομένων, πρέπει αρχικά να καθοριστούν οι ιστοσελίδες-βάσεις μέσα από τις οποίες θα πραγματοποιηθεί η εξαγωγή αυτή. Στη συνέχεια, δημιουργούνται οι ανάλογες αράχνες, μία για τη κάθε βάση δεδομένων που πρόκειται να πραγματοποιηθεί η αναζήτηση. Αυτό συμβαίνει καθώς η κάθε ιστοσελίδα έχει διαφορετική δομή και έτσι απαιτούνται διαφορετικοί κανόνες και μέθοδοι ώστε να αντληθεί η επιθυμητή πληροφορία.

Παρόλα αυτά, όλες οι αράχνες θα πρέπει να έχουν ένα κοινό χαρακτηριστικό, τη παράμετρο start_urls μέσα στην οποία δηλώνονται οι διευθύνσεις από τις οποίες θα πραγματοποιηθεί η αναζήτηση. Από τη στιγμή που η διαδικασία αναζήτησης θα πρέπει να είναι αυτοματοποιημένη, δηλαδή να πραγματοποιείται εξολοκλήρου από τον agent, τότε οι αράχνες θα πρέπει να υλοποιηθούν με τέτοιο τρόπο ώστε να μπορούν να γνωρίζουν αν υπάρχουν παραπάνω από μια σελίδες μέσα στον ίδιο ιστότοπο και στη περίπτωση που κάτι τέτοιο ισχύει να μπορεί να πραγματοποιήσει ανακατεύθυνση έτσι ώστε να λάβει τα δεδομένα από όλες τις σελίδες.

Προκειμένου να επιτευχθεί αυτό θα πρέπει να ευρεθεί ένας τρόπος, ώστε να δίνονται τα urls στην αράχνη δυναμικά. Αρχικά, αν πραγματοποιήσει κανείς μία αναζήτηση στη βάση δεδομένων rapid7 μπορεί να παρατηρήσει το εξής:



Το url της ιστοσελίδας έχει τη μορφή https://www.rapid7.com/db/?q=nmap&type=.

Έτσι λοιπόν, μπορεί κανείς εύκολα να καταλάβει πως αν αντικατασταθεί η τιμή της παραμέτρου q στο παραπάνω url, είναι εφικτό να πραγματοποιηθεί αναζήτηση για διαφορετικό πακέτο. Επομένως, για να δημιουργηθούν τα urls δυναμικά, αρκεί να συμπεριληφθεί μέσα σε μια δομή επανάληψης for, η οποία θα εκτελείται κάθε φορά

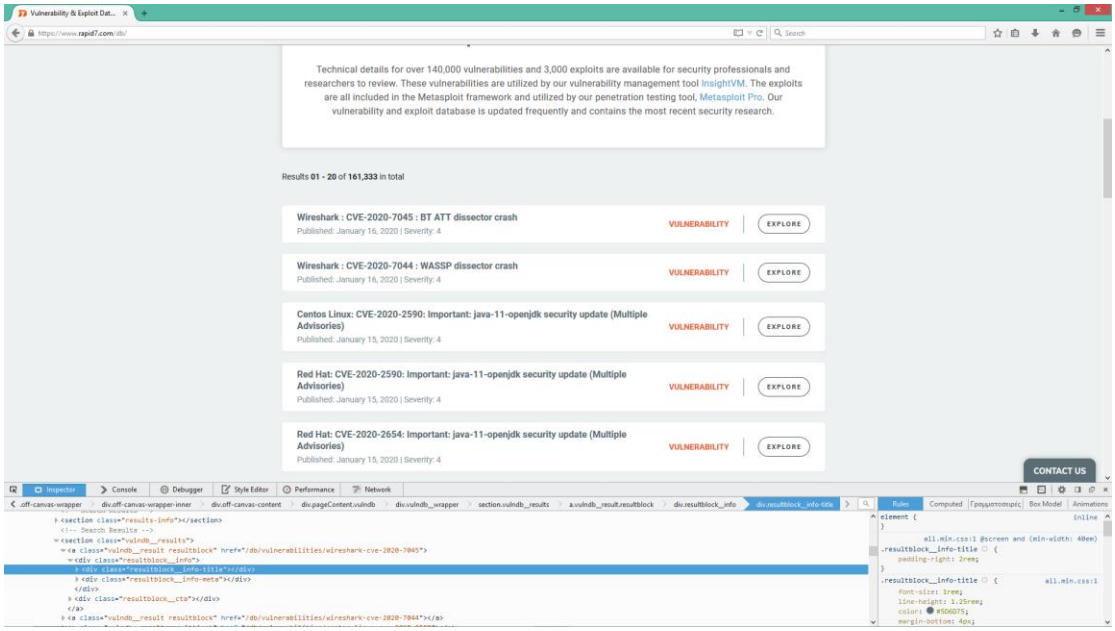
για το εκάστοτε πακέτο το οποίο θα συμπεριλαμβάνετε στο url και έτσι θα αναζητούνται οι ευπάθειες για το συγκεκριμένο πακέτο.

```
# This is a spider that collects data rapod7 database

import scrapy,os,sys,re
from web_scraping.items import WebScrapingItem

packages = os.popen("dpkg -l | grep '^ii' | awk '{print $2}'").read().split('\n')   #get the packages' name using dpkg
versions = os.popen("dpkg -l | grep '^ii' | awk '{print $3}'").read().split('\n')   #get the packages' version
kernel_version= os.popen('uname -r').read() #get the linux kernel version

kernel_version=kernel_version[:6]

class QuotesSpider(scrapy.Spider):
    results=()
    start_urls=[ 'https://www.rapid7.com/db/?q=linux kernel'+ ' ' + kernel_version]
    for i,j in zip(packages,versions):
        j=j[:3]
        results=(i+" "+(re.sub(r'([^\w\s]|_)+(?=\s|$)', '', j)))

        name = "rapid7"

        if str(results) != str(' '):
            start_urls.append('https://www.rapid7.com/db/?q='+str(results))


    def parse(self, response):
        item = WebScrapingItem()

        name = response.css('.search-query > input:nth-child(1)::attr(value)').get()

        for quote in response.css('section.vulndb__results'):
            item['name'] = quote.css('div.resultblock__info-title::text').get() # save the result into item in order to store it in a better csv format
            vul = re.sub('[\W_]+', ' ', str(item)) #remove special characters
            vul = vul[12:] #remove some bad characters at the start of the string
            vul =  vul[:-4] #remove some bad characters at the end of the string
            item['vulnerability']=vul
            item ['name'] = name
            yield item



        next_page = response.css('.large-12 > ul:nth-child(1) > li:nth-child(3) > a::attr(href)').get()
      # if item != ' ':
       #    yield item

        if next_page is not None:

            yield response.follow(next_page, callback=self.parse)
```

Εικόνα 12 Πηγαίος Κώδικας για εξαγωγή δεδομένων από τη βάση Rapid7

Όπως φαίνεται και στη παραπάνω εικόνα, χρησιμοποιώντας τη δομή επανάληψης for, μπορεί να διασφαλιστεί η δυναμική παραγωγή των urls προκειμένου να πραγματοποιηθεί η αναζήτηση για τα εγκατεστημένα πακέτα. Επιπροσθέτως, χρησιμοποιώντας το scrapy framework, δίνετε η δυνατότητα στον χρήστη να μπορεί να πραγματοποιήσει redirect, καθώς η κλάση parse καλείται σε κάθε επανάληψη της for και έτσι συλλέγεται η πληροφορία από όλες τις σελίδες.

Αρκετά σημαντικός είναι ο ορισμός των κανόνων ώστε να γνωρίζει η αράχνη σε ποια στοιχεία του κώδικα html βρίσκεται η πληροφορία που επιθυμεί ο χρήστης. Για να καθοριστεί αυτό θα πρέπει να εξεταστεί ο κώδικας της σελίδας και να εντοπιστούν τα εν λόγο στοιχεία.



Εικόνα 13 Κώδικας Html της σελίδας Rapid7

Στη παραπάνω εικόνα, μπορεί κανείς να παρατηρήσει πως για να πραγματοποιηθεί η διαδικασία που αναλύθηκε προηγουμένως, αρκεί να επιλέξει ένα από τα αποτελέσματα και να επιλέξει έλεγχος στοιχείου πατώντας δεξί κλικ.

4.3 Ιδιαιτερότητα της Exploit-Db

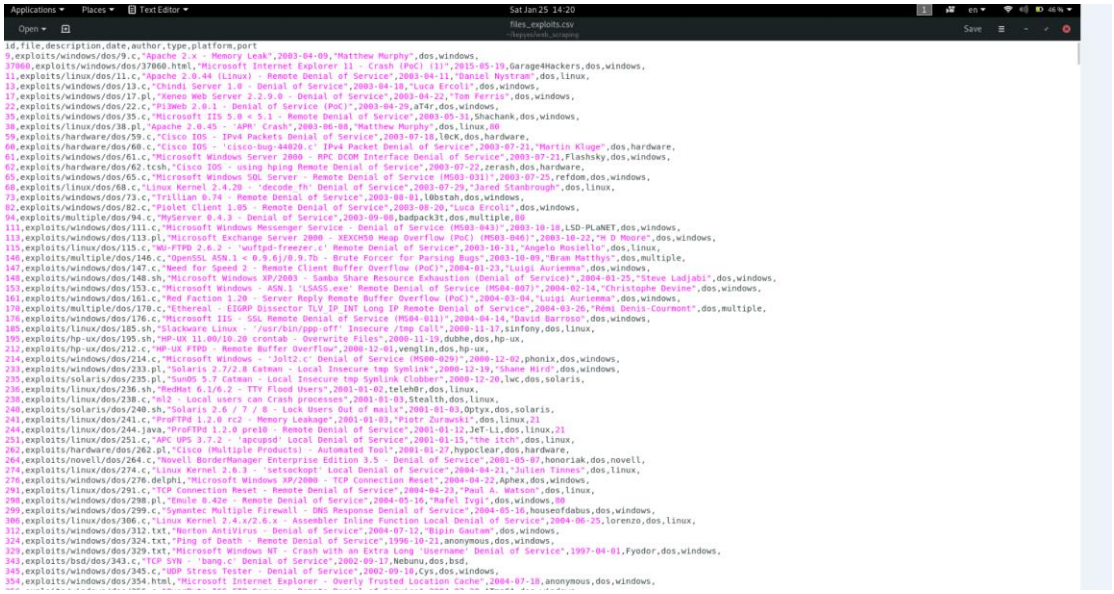
Όπως αναφέρθηκε προηγουμένως, για να πραγματοποιηθεί η εξόρυξη θα πρέπει να δοθούν τα σωστά ορίσματα στη παράμετρο start urls, με σκοπό να μπορούν να παραχθούν δυναμικά οι αναζητήσεις. Στο προηγούμενο παράδειγμα σχετικά με τη βάση δεδομένων rapid7, ο τρόπος με τον οποίο μπορεί κάποιος να δημιουργήσει τα url δυναμικά, είναι με χρήση εκφράσεων xpath ή με χρήση css selectors σε συνδυασμό με το μονοπάτι στο οποίο βρίσκετε το επιθυμητό στοιχείο.

Σε αντίθεση με τη προηγούμενη περίπτωση της βάσης δεδομένων rapid7, η βάση δεδομένων exploit-db, η οποία είναι ίσως η πιο γνωστή και αξιόπιστη βάση σχετικά με την αναζήτηση ευπαθειών, έχει μια ιδιαιτερότητα. Συγκεκριμένα, αν παρατηρήσει κανείς το url κατά την αναζήτηση αυτό που θα συμπεράνει είναι πως το url δεν μεταβάλλεται, δηλαδή δεν χρησιμοποιεί κάποια παράμετρο στο request του προς τον κεντρικό server. Αυτό πιθανώς συμβαίνει, προκειμένου να αποφευχθούν τυχόν επιθέσεις τύπου sql injection. Επομένως, δεν είναι η εφικτή η υλοποίηση κάποιας αράχνης ώστε να μπορέσει να εξάγει τα επιθυμητά δεδομένα, καθώς δεν γίνετε να πραγματοποιηθεί αναζήτηση για τα εγκατεστημένα πακέτα από τη στιγμή που δεν μπορεί να δοθεί κάποια παράμετρος κατά το αίτημα.

Επιπλέον, μία άλλη δυσκολία είναι πως αν κανείς εξετάσει τον κώδικα html της ιστοσελίδας θα παρατηρήσει πως δεν υπάρχει κάποιος τρόπος μέσα από τη χρήση web scraper, προκειμένου να τεθεί μία τιμή στο πλαίσιο της αναζήτησης ώστε να μπορούν να αντληθούν τα επιθυμητά δεδομένα.

Για αυτό το λόγο, μία εναλλακτική λύση μπορεί να χρησιμοποιηθεί στη προκειμένη περίπτωση. Αναλυτικότερα, στο σύνδεσμο (https://github.com/offensive-security/exploitdb) μπορεί κανείς να βρει ένα αρχείο το οποίο ονομάζετε files_exploits.csv. Το αρχείο αυτό, περιέχει όλες τις ευπάθειες για όλα τα λειτουργικά συστήματα ή τις εφαρμογές που έχουν καταγραφεί από το καιρό της δημιουργίας της βάσης δεδομένων exploit-db έως και σήμερα. Επιπροσθέτως, το αρχείο αυτό ανανεώνεται καθημερινά περίπου μεταξύ 10-11π.μ. ώρα Ελλάδος. Επομένως, προκειμένου να αντληθούν δεδομένα και να αναζητηθούν πιθανές ευπάθειες, αρκεί να πραγματοποιεί κανείς λήψη αυτού του αρχείου, οποτεδήποτε χρησιμοποιεί τον agent ώστε να ενημερώνεται το εν λόγο αρχείο.

Στη συνέχεια, μέσα από το αρχείο search.py, πραγματοποιείται η λήψη του αρχείου και στη συνέχεια πραγματοποιείται δυναμικά μία αναζήτηση για τα εγκατεστημένα πακέτα που ενδιαφέρουν τον χρήστη. Με αυτό τον τρόπο, επιλύεται το αρχικό πρόβλημα, καθώς επίσης μειώνεται ο χρόνος εκτέλεσης του αρχείου search.py μειώνεται σημαντικά, αφού πλέον το μόνο που χεριάζετε είναι να εξετάσει ένα τοπικό αρχείο και όχι να κατεβάσει την ιστοσελίδα και να ψάξει μέσα σε αυτή.



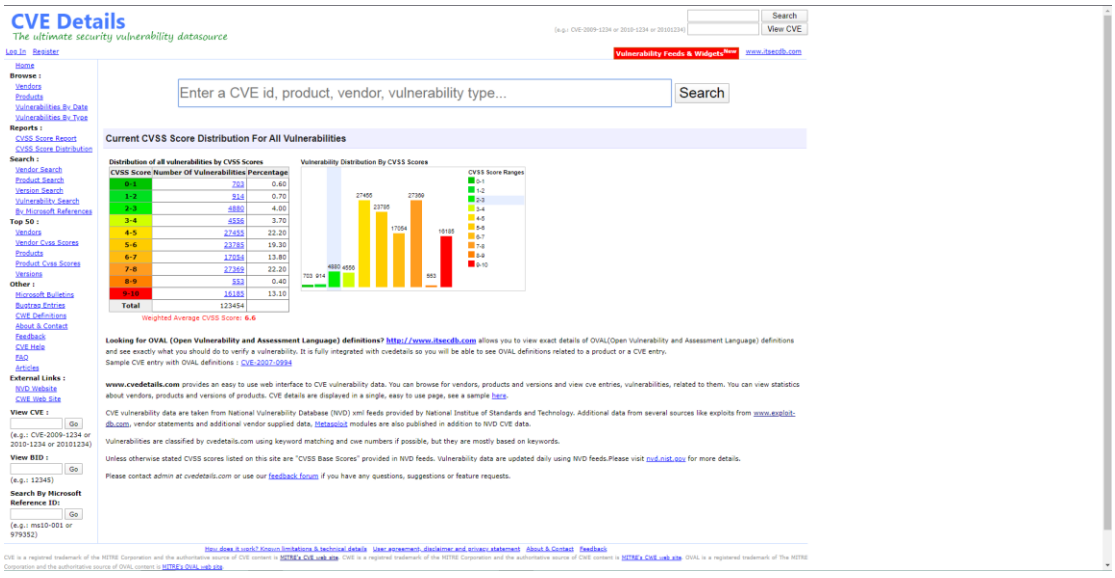

```
id,file,description,date,author,type,platform,port
9,exploits/windows/dos/9.c,"Apache 2.x - Memory Leak",2003-04-09,"Matthew Murphy",dos,windows,
37060,exploits/windows/dos/37060.html,"Microsoft Internet Explorer 11 - Crash (PoC) (1)",2015-05-19,Garage4Hackers,dos,windows,
11,exploits/linux/dos/11.c,"Apache 2.0.44 (Linux) - Remote Denial of Service",2003-04-11,"Daniel Nystram",dos,linux,
13,exploits/windows/dos/13.c,"Chindi Server 1.0 - Denial of Service",2003-04-18,"Luca Ercoli",dos,windows,
17,exploits/windows/dos/17.pl,"Xeneo Web Server 2.2.9.0 - Denial of Service",2003-04-22,"Tom Ferris",dos,windows,
22,exploits/windows/dos/22.c,"PiWeb 2.0.1 - Denial of Service (PoC)",2003-04-29,at4r,dos,windows,
35,exploits/windows/dos/35.c,"Microsoft IIS 5.0 < 5.1 - Remote Denial of Service",2003-05-31,Shachank,dos,windows,
38,exploits/linux/dos/38.pl,"Apache 2.0.45 - 'APR' Crash",2003-06-08,"Matthew Murphy",dos,linux,80
59,exploits/hardware/dos/59.c,"Cisco IOS - IPv4 Packets Denial of Service",2003-07-10,l0ck,dos,hardware,
60,exploits/hardware/dos/60.c,"Cisco IOS - 'cisco-bug-44020.c' IPv4 Packet Denial of Service",2003-07-21,"Martin Kluge",dos,hardware,
61,exploits/windows/dos/61.c,"Microsoft Windows Server 2000 - RPC DCOM Interface Denial of Service",2003-07-21,Flashsky,dos,windows,
62,exploits/hardware/dos/62.tcsh,"Cisco IOS - using hping Remote Denial of Service",2003-07-22,zerash,dos,hardware,
65,exploits/windows/dos/65.c,"Microsoft Windows SQL Server - Remote Denial of Service (MS03-031)",2003-07-25,refdom,dos,windows,
68,exploits/linux/dos/68.c,"Linux Kernel 2.4.20 - 'decode_fh' Denial of Service",2003-07-29,"Jared Stanbrough",dos,linux,
73,exploits/windows/dos/73.c,"Trillian 0.74 - Remote Denial of Service",2003-08-01,l0bstah,dos,windows,
82,exploits/windows/dos/82.c,"Piolet Client 1.05 - Remote Denial of Service",2003-08-20,"Luca Ercoli",dos,windows,
94,exploits/multiple/dos/94.c,"MyServer 0.4.3 - Denial of Service",2003-09-08,badpack3t,dos,multiple,80
111,exploits/windows/dos/111.c,"Microsoft Windows Messenger Service - Denial of Service (MS03-043)",2003-10-18,LSD-PLaNET,dos,windows,
113,exploits/windows/dos/113.pl,"Microsoft Exchange Server 2000 - XEXCH50 Heap Overflow (PoC) (MS03-046)",2003-10-22,"H D Moore",dos,windows,
115,exploits/linux/dos/115.c,"WU-FTPD 2.6.2 - 'wuftpd-freezer.c' Remote Denial of Service",2003-10-31,"Angelo Rosiello",dos,linux,
140,exploits/multiple/dos/146.c,"OpenSSL ASN.1 < 0.9.6j/0.9.7b - Brute Forcer for Parsing Bugs",2003-10-09,"Bram Matthys",dos,multiple,
147,exploits/windows/dos/147.c,"Need for Speed 2 - Remote Client Buffer Overflow (PoC)",2004-01-23,"Luigi Auriemma",dos,windows,
148,exploits/windows/dos/148.sh,"Microsoft Windows XP/2003 - Samba Share Resource Exhaustion (Denial of Service)",2004-01-25,"Steve Ladjabi",dos,windows,
153,exploits/windows/dos/153.c,"Microsoft Windows - ASN.1 'LSASS.exe' Remote Denial of Service (MS04-007)",2004-02-14,"Christophe Devine",dos,windows,
161,exploits/windows/dos/161.c,"Red Faction 1.20 - Server Reply Remote Buffer Overflow (PoC)",2004-03-04,"Luigi Auriemma",dos,windows,
170,exploits/multiple/dos/170.c,"Ethereal - EIGRP Dissector TLV IP_INT Long IP Remote Denial of Service",2004-03-26,"Rémi Denis-Courmont",dos,multiple,
176,exploits/windows/dos/176.c,"Microsoft IIS - SSL Remote Denial of Service (MS04-011)",2004-04-14,"David Barroso",dos,windows,
185,exploits/linux/dos/185.sh,"Slackware Linux - '/usr/bin/ppp-off' Insecure /tmp Call",2000-11-17,sinfony,dos,linux,
195,exploits/hp-ux/dos/195.sh,"HP-UX 11.00/10.20 crontab - Overwrite Files",2000-11-19,dubhe,dos,hp-ux,
212,exploits/hp-ux/dos/212.c,"HP-UX FTPD - Remote Buffer Overflow",2000-12-01,venglin,dos,hp-ux,
214,exploits/windows/dos/214.c,"Microsoft Windows - 'Jolt2.c' Denial of Service (MS00-029)",2000-12-02,phonix,dos,windows,
233,exploits/windows/dos/233.pl,"Solaris 2.7/2.8 Catman - Local Insecure tmp Symlink",2000-12-19,"Shane Hird",dos,windows,
235,exploits/solaris/dos/235.pl,"SunOS 5.7 Catman - Local Insecure tmp Symlink Clobber",2000-12-20,lwc,dos,solaris,
236,exploits/linux/dos/236.sh,"RedHat 6.1/6.2 - TTY Flood Users",2001-01-02,teleh0r,dos,linux,
238,exploits/linux/dos/238.c,"ml2 - Local users can Crash processes",2001-01-03,Stealth,dos,linux,
240,exploits/solaris/dos/240.sh,"Solaris 2.6 / 7 / 8 - Lock Users Out of mailx",2001-01-03,Optyx,dos,solaris,
241,exploits/linux/dos/241.c,"ProFTPd 1.2.0 rc2 - Memory Leakage",2001-01-03,"Piotr Zurawski",dos,linux,21
244,exploits/linux/dos/244.java,"ProFTPd 1.2.0 pre10 - Remote Denial of Service",2001-01-12,JeT-Li,dos,linux,21
251,exploits/linux/dos/251.c,"APC UPS 3.7.2 - 'apcupsd' Local Denial of Service",2001-01-15,"the itch",dos,linux,
262,exploits/hardware/dos/262.pl,"Cisco (Multiple Products) - Automated Tool",2001-01-27,hypoclear,dos,hardware,
264,exploits/novell/dos/264.c,"Novell BorderManager Enterprise Edition 3.5 - Denial of Service",2001-05-07,honoriak,dos,novell,
274,exploits/linux/dos/274.c,"Linux Kernel 2.6.3 - 'setsockopt' Local Denial of Service",2004-04-21,"Julien Tinnes",dos,linux,
276,exploits/windows/dos/276.delphi,"Microsoft Windows XP/2000 - TCP Connection Reset",2004-04-22,Aphex,dos,windows,
291,exploits/linux/dos/291.c,"TCP Connection Reset - Remote Denial of Service",2004-04-23,"Paul A. Watson",dos,linux,
298,exploits/windows/dos/298.pl,"Emule 0.42e - Remote Denial of Service",2004-05-10,"Rafel Ivgi",dos,windows,80
299,exploits/windows/dos/299.c,"Symantec Multiple Firewall - DNS Response Denial of Service",2004-05-10,houseofdabus,dos,windows,
306,exploits/linux/dos/306.c,"Linux Kernel 2.4.x/2.6.x - Assembler Inline Function Local Denial of Service",2004-06-25,lorenzo,dos,linux,
312,exploits/windows/dos/312.txt,"Norton AntiVirus - Denial of Service",2004-07-12,"Bipin Gautam",dos,windows,
324,exploits/windows/dos/324.txt,"Ping of Death - Remote Denial of Service",1996-10-21,anonymous,dos,windows,
329,exploits/windows/dos/329.txt,"Microsoft Windows NT - Crash with an Extra Long 'Username' Denial of Service",1997-04-01,Fyodor,dos,windows,
343,exploits/bsd/dos/343.c,"TCP SYN - 'bang.c' Denial of Service",2002-09-17,Nebunu,dos,bsd,
345,exploits/windows/dos/345.c,"UDP Stress Tester - Denial of Service",2002-09-10,Cys,dos,windows,
354,exploits/windows/dos/354.html,"Microsoft Internet Explorer - Overly Trusted Location Cache",2004-07-18,anonymous,dos,windows,
```

Εικόνα 14 Αρχείο files_exploits.csv

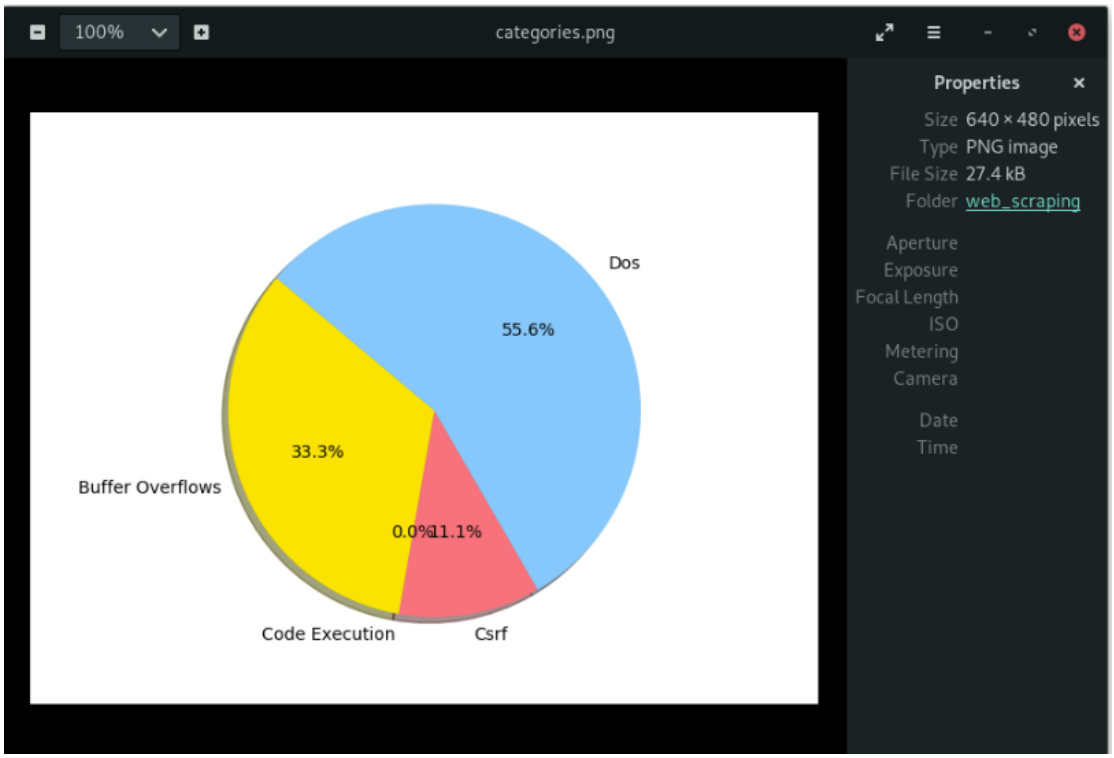
4.4 Αναζήτηση κωδικών CVE στη Βάση Δεδομένων CVE-DETAILS

Αφότου πραγματοποιηθεί η αναζήτηση στις βάσεις δεδομένων rapid7 και exploit-db, όπως εξηγήθηκε προηγουμένως, η εκτέλεση του προγράμματος συνεχίζετε με την εξαγωγή των κωδικών cve από τις πιθανές ευπάθειες που έχουν βρεθεί. Πιο συγκεκριμένα, κατά την εκτέλεση της αναζήτησης στη βάση δεδομένων exploit-db, πέρα από την ονομασία της ευπάθειας ο agent αποθηκεύει και τον κωδικό cve που τη συνοδεύει. Αυτό γίνετε, προκειμένου στη συνέχεια να εκτελεστεί η αράχνη get_cves.py, η οποία πραγματοποιεί αναζήτηση των ευπαθειών που έχουν συλλεχθεί από τη βάση δεδομένων rapid7 στη βάση δεδομένων exploit-db. Με αυτό τον τρόπο, είναι εφικτό να βρεθούν οι κωδικοί cves, με σκοπό στη συνέχεια να πραγματοποιηθεί αναζήτηση στη βάση δεδομένων cve-details. Αυτό θα συμβάλλει, στη συγκέντρωση στατιστικών γύρω από την επικινδυνότητα των ευπαθειών. Αναλυτικότερα, οι ευπάθειες μπορούν να χωριστούν σε κατηγορίες από μικρής σημασίας μέχρι μέγιστης επικινδυνότητας.

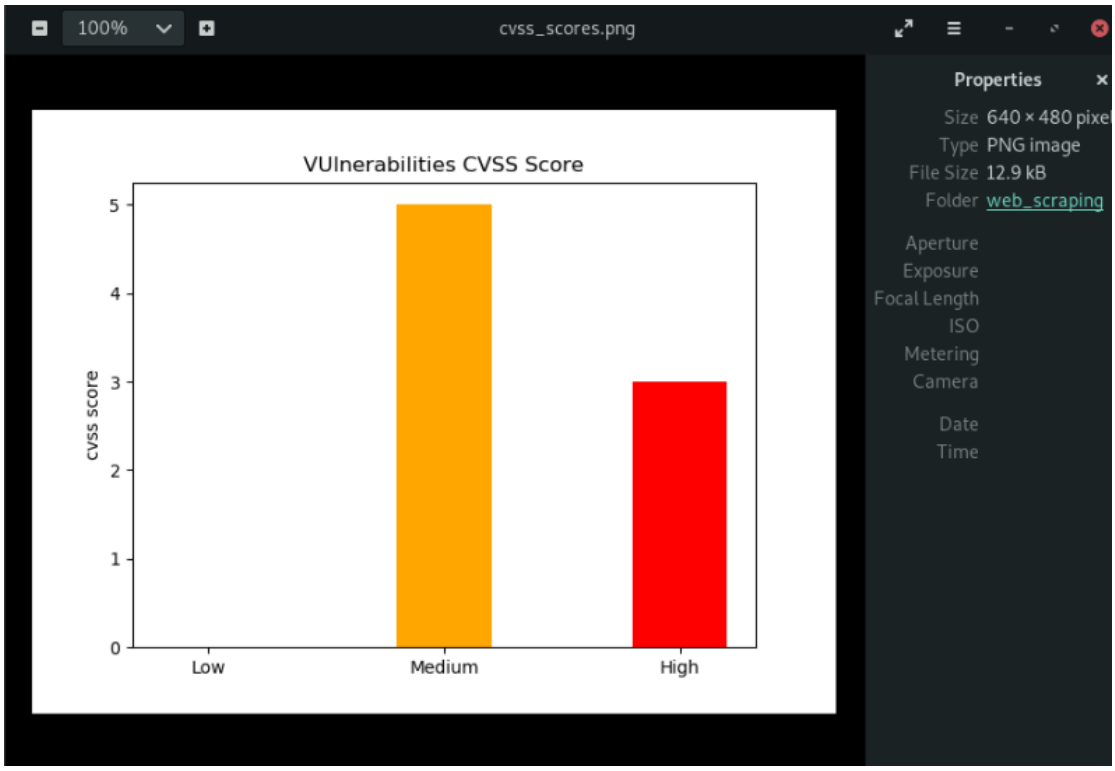
Επομένως, ο χρήστης θα πρέπει να είναι ενήμερος για το πόσο επικίνδυνες είναι ή όχι οι πιθανές ευπάθειες που αντιμετωπίζει το σύστημά του. Επιπλέον, συγκεντρώνοντας τις παραπάνω πληροφορίες, είναι εφικτή η δημιουργία διαγραμμάτων, προκειμένου ο χρήστης να μπορεί με μία ματιά να αντιληφθεί τι οφείλει να διορθώσει άμεσα.



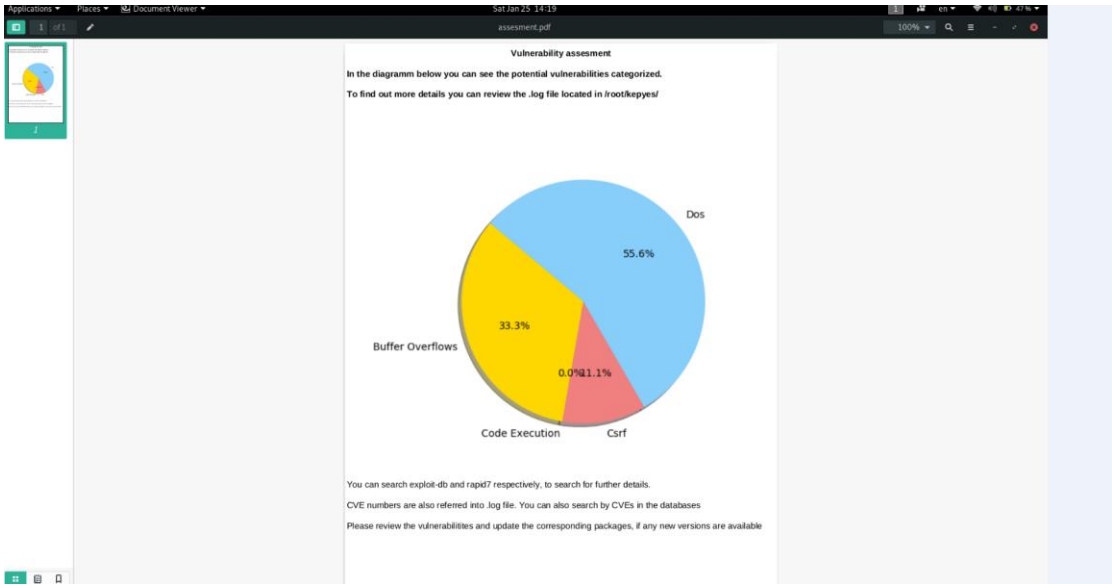
Εικόνα 15 Βάση Δεδομένων Cve Details



Εικόνα 16 Κατηγοριοποίηση των πιθανών ευπαθειών



Εικόνα 17 Κατηγορίες CVSS Score



Εικόνα 18 Agent's report

Κεφάλαιο 5ο

Συμπεράσματα

Η τεχνική του Web Scraping, σίγουρα αποτελεί μία μέθοδο με αρκετά οφέλη, αλλά ταυτόχρονα έχει αποδειχτεί πονοκέφαλος για επιχειρήσεις ή οργανισμούς που στηρίζονται αποκλειστικά στη παραγωγή και τη διάθεση πληροφοριών μέσα από το διαδίκτυο. Για παράδειγμα, υπάρχουν εταιρίες οι οποίες εξάγουν δεδομένα για το χρηματιστήριο και πως αυτό αναμένεται να κινηθεί κατά τη διάρκεια της ημέρας. Πάνω σε αυτό, βασίζονται και διάφορες στατιστικές μελέτες που μπορεί να πραγματοποιηθούν ώστε να εξαχθούν δεδομένα για διάφορους ενδιαφερόμενους. Σε αυτή τη περίπτωση, η εταιρία που παράγει όλα αυτά τα δεδομένα βασίζετε αποκλείστηκα σε αυτά με σκοπό το κέρδος και την επιβίωσή της. Αν σε αυτή τη περίπτωση κάποιος χρήστης με τη βοήθεια των Web Scrapers, παρέμβει τότε καταστρέφει την αποκλειστικότητα της πληροφορίας.

Επιπλέον, σε πολλές περιπτώσεις το Web Scraping μπορεί να ενισχύσει τόσο τον αθέμητο ανταγωνισμό, όσο και τη παραπλάνηση των καταναλωτών. Είναι σύνηθες πλέον φαινόμενο, πολλά ηλεκτρονικά καταστήματα να διαθέτουν μηχανισμούς σύγκρισης τιμών με άλλα παρόμοια καταστήματα ανταγωνιστές. Όπως είναι φυσικό σε αυτές τις περιπτώσεις, παρουσιάζονται καταστήματα, τα οποία διαθέτουν παρόμοια προϊόντα σε ακριβότερες τιμές. Τα δεδομένα, από τα υπόλοιπα καταστήματα, εξάγονται με χρήση των Web scrapers και κατά τον τρόπο τον οποίο αναλύθηκε παραπάνω, ενισχύεται ο αθέμητος ανταγωνισμός.

Εκτός από τα παραπάνω, θα πρέπει κανείς να λάβει υπόψιν του, το γεγονός πως σε αρκετές περιπτώσεις, η χρήση των Web Scrapers μπορεί να επιφέρει νομικές κυρώσεις, καθότι παραβιάζονται αρκετοί νόμοι οι οποίοι σχετίζονται με κλοπή πνευματικής ιδιοκτησίας ή ακόμα και με αντιγραφή δεδομένων. Σε οποιαδήποτε από τις περιπτώσεις αυτές, εννοείτε πως ο ιδιοκτήτης των δεδομένων έχει τη δυνατότητα να κινηθεί νομικά εναντίων του χρήστη του Web Scraper. Έτσι λοιπόν, είναι αναγκαίο όσοι χρησιμοποιούν Web Scrapers να μπορούν να γνωρίζουν για ποιο λόγο και σε ποιες περιπτώσεις θα χρησιμοποιηθούν τα εργαλεία αυτά.

Επιπροσθέτως, προκειμένου να υλοποιηθούν οι Web Scrapers, θα πρέπει να χρησιμοποιηθούν εκφράσεις τύπου xpath ή css selector, με βάση τη δομή και τον κώδικα Html της ιστοσελίδας, αυτό σημαίνει πως, οποιαδήποτε αλλαγή πραγματοποιηθεί στον κώδικα ενδεχομένως να επηρεάσει την αποτελεσματική λειτουργία του scraper. Λόγω αυτού του προβλήματος, θα πρέπει να προβλεφθεί η συνεχής υποστήριξη και βελτίωσή του ώστε να μην διαταραχθεί η λειτουργία του. Αυτό φυσικά μπορεί να σημαίνει επιπλέον κόστος για μία επιχείρηση η οποία βασίζετε στην εξαγωγή και τροποποίηση δεδομένων και σίγουρα θα επηρεάσει σε σημαντικό βαθμό τη λειτουργία της.

Παρόλα αυτά, ένας Web Scraper μπορεί επιπλέον να αποδειχθεί ιδιαιτέρως χρήσιμος σε πολλές περιπτώσεις, ειδικά όταν καλείται να αντλήσει πληροφορίες από ανοιχτές βάσεις δεδομένων, όπου τα δεδομένα είναι διαθέσιμα στο ευρύ κοινό.

Με αυτόν το τρόπο, μπορούν να συγκεντρωθούν χρήσιμες πληροφορίες για διάφορα ζητήματα, οι οποίες μπορούν να βοηθήσουν στη πρόβλεψη κάποιων γεγονότων ή στη διευκόλυνση των χρηστών. Αναλυτικότερα, ο λόγος που χρησιμοποιήθηκε η τεχνική του Web Scrraping στην εργασία αυτή, ήταν προκειμένου να αντληθούν πληροφορίες από ανοιχτές βάσεις δεδομένων, ώστε να μπορέσουν να αποφευχθούν τυχόν κενά ασφάλειας λόγω συγκεκριμένων εγκατεστημένων εφαρμογών. Με τη δημιουργία ενός τέτοιου μηχανισμού, είναι δυνατόν να μπορούν να ανιχνευθούν κάποιες ενδεχομένως, ευάλωτες εφαρμογές οι οποίες είναι δυνατόν να χρησιμοποιηθούν από κακόβουλους χρήστες οι οποίοι επιθυμούν να πραγματοποιήσουν μία επίθεση και να βλάψουν τον χρήστη. Επομένως, μπορεί κανείς εύκολα να συμπεράνει πως, η τεχνική του Web Scraping μπορεί να αποδειχθεί ένα πολύ χρήσιμο εργαλείο το οποίο μπορεί να χρησιμοποιηθεί με τη βοήθεια των ανοιχτών βάσεων δεδομένων, ως εργαλείο για την αποφυγή τυχόν ευπαθειών. Επιπροσθέτως, θα μπορούν να δημιουργηθούν και άλλα εργαλεία τα οποία θα μπορούσαν να εξυπηρετήσουν στην επίλυση διαφόρων προβλημάτων με χρήση των Web Scrapers, καθώς πολλοί τομείς βασίζονται στα δεδομένα και τη πληροφορία και επομένως η πληροφορία αποτελεί σήμερα το πιο πολύτιμο αγαθό.

Σε κάθε περίπτωση όμως, είναι αναγκαίο κανείς προτού χρησιμοποιήσει κάποιον Web Scraper, να ορίσει τη πηγή από όπου θα αντλήσει τις πληροφορίες του και στη συνέχεια, οφείλει να εξετάσει αναλυτικά τι δεδομένα θα συλλέξει, πως θα τα επεξεργαστεί και θα τα χρησιμοποιήσει και τέλος εάν αυτό επιτρέπετε από τον κάτοχο των δεδομένων. Επιπλέον, ιδιαίτερη σημασία πρέπει να δοθεί στα αρχεία robots.txt τα οποία διαθέτουν οι ιστοσελίδες, καθώς μέσα σε αυτά αναφέρετε αν επιτρέπετε ή όχι η εξαγωγή των δεδομένων.

Εν κατακλείδι, σαφώς οι Web Scrapers μπορούν να παρέχουν αρκετά οφέλη προς τους χρήστες τους και όχι μόνο, αλλά είναι αναγκαίο να καθοριστεί ο τρόπος και ο σκοπός για τον οποίο θα χρησιμοποιηθούν. Σε αρκετές περιπτώσεις, μπορούν να χρησιμοποιηθούν ώστε να εξάγουν παράνομα δεδομένα, όπου σε αυτή τη περίπτωση ο χρήστης μπορεί να έρθει αντιμέτωπος με νομικές κυρώσεις. Σε αντίθετη περίπτωση, με τη βοήθεια των Web Scrapers, μπορούν να αναπτυχθούν αρκετά εργαλεία τα οποία θα βοηθήσουν στη συλλογή και την επεξεργασία πολύτιμων δεδομένων για αρκετούς τομείς, όπως για παράδειγμα ο τομέας της ασφάλειας των πληροφοριών.

Τέλος, είναι σημαντικό να αναφερθεί το γεγονός, πως τα Frameworks, όπως είναι το Scrapy, βοηθούν σημαντικά στην υλοποίηση τέτοιων εργαλείων, διότι αυτοματοποιούν αρκετές διαδικασίες και έτσι δίνουν τη δυνατότητα στο χρήστη να μπορεί να αναπτύσσει πολύπλοκα εργαλεία σε μικρό χρονικό διάστημα. Τα Frameworks αυτά ποικίλουν, γεγονός που σημαίνει πως καθένας μπορεί να επιλέξει ποιο από αυτά θέλει να χρησιμοποιήσει ανάλογα με τα οφέλη του ή σύμφωνα με τη γλώσσα προγραμματισμού που επιθυμεί.

Αναφορές

1. Scrapy Framework Documentation

2. An Overview On Web Scraping Techniques And Tools, Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode

3. Web Scraping Data Extraction from websites, Vojtech Draxl

4. Study & Analysis of Web Content Mining Tools to Improve Techniques of Web Data Mining, Arvind Kumar Sharma1, P.C. Gupta

5. Using XPaths of Inbound Links to Cluster Template-Generated Web Pages, Tomas Grigalis1 and Antanas Čenys1

6. G. Srivastava, K. Sharma, V. Kumar," Web Mining: Today and Tomorrow", in the Proceedings of 2011 3rd International Conference on Electronics Computer Technology (ICECT), pp.399-403, April 2011.

7. Algorithms for Web Scraping, Patrick Hagge Cording

8. Information Extraction Using Web Usage Mining, Web Scrapping and Semantic Annotation, Sanjay Kumar Malik, SAM Rizvi

9. Web Scraping With Python, Richard Lawson

10. Comparison of Web Scraping Techniques : Regular Expression, HTML DOM and Xpath, Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, Firman Firdaus

11. A dive into Web Scraper world, Deepak Kumar Mahto, Lisha Singh

12. A Comparative Study on Web Scraping, De S Sirisuriya, SCM

13. Legality and Ethics of Web Scraping, Vlad Krotov, Leiser Silva

14. Data Extraction for Decision-Support Systems: Application in Labour Market Monitoring and Analysis Maxim Bakaev and Tatiana Avdeenko