Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάλυση και συσχέτιση αζήτητης ηλεκτρονικής επικοινωνίας** **Spam Analysis and Correlation Engine**
Ονοματεπώνυμο Φοιτητή	**Γεωργία Μιχαλά**
Πατρώνυμο	**Βασίλειος**
Αριθμός Μητρώου	**ΜΠΚΣΑ 18016**
Επιβλέπων	**Κωνσταντίνος Πατσάκης, Αναπληρωτής Καθηγητής**

Ημερομηνία Παράδοσης **Δεκέμβριος 2020**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Πατσάκης Κωνσταντίνος	Αλέπης Ευθύμιος	Σακκόπουλος Ευάγγελος
Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής	Επίκουρος Καθηγητής

«Το παρόν έγγραφο με τίτλο: «SACE: Μηχανή ανάλυσης και συσχέτισης μηνυμάτων αζήτητης ηλ. Επικοινωνίας (spam)», πραγματοποιήθηκε με σκοπό την εκπόνηση της διατριβής για το μεταπτυχιακό πρόγραμμα σπουδών «Κατανεμημένα Συστήματα, Ασφάλεια Και Αναδυόμενες Τεχνολογίες Πληροφορίας» στο τμήμα Πληροφορικής του Πανεπιστημίου Πειραιά.

Θα ήθελα να ευχαριστήσω ιδιαιτέρως τον κ. Χρυσάνθου Ανάργυρο που με εμπιστεύτηκε να συνεχίσουμε την συνεργασία μας η οποία ξεκίνησε το 2017 με σκοπό την περάτωση της διπλωματικής εργασίας που πραγματοποιήθηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Δημοκρίτειου Πανεπιστημίου Θράκης. Επίσης θα ήθελα να εκφράσω τις ευχαριστίες μου στον κ.Πατσάκη Κωνσταντίνο για την αποδοχή και την θέληση να συνεργαστούμε στην διάρκεια της προσπάθειας με σκοπό την εκπόνηση της παρούσας διατριβής. Ευχαριστώ και τους δύο για ανταπόκριση τους σ' όλες μου τις απορίες, ανεξαρτήτως ημέρας και ώρας.»

Πίνακας περιεχομένων

Περίληψη

Η παρούσα διατριβή πραγματεύεται την δημιουργία μιας εφαρμογής η οποία συλλέγει κεφαλίδες μηνυμάτων ηλεκτρονικού ταχυδρομείου με σκοπό την ανάλυση τους και την εύρεση συνθηκών υπό των οποίων μπορεί να σημανθεί ένα μήνυμα ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητο. Η εφαρμογή υλοποιήθηκε σε γλώσσα python. Οι κεφαλίδες των e-mails στέλνονται στις επιμέρους υπηρεσίες της εφαρμογής με τη χρήση του JSON Web Token και κατά την αποστολή αυτών με σκοπό την ανάλυση τους τοποθετούνται σε ουρά εργασιών μέσω του Celery. Η αποθήκευση των δεδομένων έγινε σε MySQL, MongoDB βάση δεδομένων και στο ElasticSearch. Επιπρόσθετη παραμετροποίηση των δεδομένων έγινε με τη βοήθεια του Logstash και η αναπαράσταση αυτών πραγματοποιήθηκε με τη συμβολή του Kibana.

Abstract

This dissertation focuses on the creation of an application that collects e-mail headers in order to analyze them and find conditions from which an e-mail may be labeled as spam. The application was implemented in python language. The e-mail headers are sent to the individual services using the JSON Web Token and when sent for analyzing purposes they are placed in Celery Task Queue. The data was stored in MySQL, MongoDB database and ElasticSearch. Additional data configuration was performed with the help of Logstash and Kibana was used for data visualization capabilities.

Εισαγωγή

Το ηλεκτρονικό ταχυδρομείο (e-mail) αποτελεί μια από τις ταχύτερες και οικονομικότερες μορφές επικοινωνίας. Δεδομένου ότι είναι εξαιρετικά εύκολο να αποσταλεί, έχει κερδίσει τεράστια δημοτικότητα όχι μόνο ως μέσο για την ανταλλαγή μηνυμάτων μεταξύ φίλων και συναδέλφων αλλά και ως μέσο για τη διεξαγωγή ηλεκτρονικού εμπορίου. Δυστυχώς, οι ίδιες αρετές που έχουν κάνει το e-mail δημοφιλές στους χρήστες, έχουν επίσης δελεάσει διάφορους εμπόρους να βομβαρδίζουν τα ηλεκτρονικά ταχυδρομεία με ανεπιθύμητα μηνύματα (spam) σχετικά με την εταιρεία ή το διαφημιζόμενο προϊόν τους. Επιπρόσθετα, το e-mail χρησιμοποιείται και από κακόβουλους χρήστες, οι οποίοι αποστέλλουν μηνύματα ηλ. ψαρέματος (phishing) ή κακόβουλα αρχεία (ως scripts, documents, ακόμα και συνημμένα εκτελέσιμα αρχεία). Μέσω των παραπάνω, επιτήδειοι επιδιώκουν την υποκλοπή δεδομένων ή / και τη μη εξουσιοδοτημένη πρόσβαση σε υπολογιστικά συστήματα με σκοπό τον εκβιασμό ανυποψίαστων χρηστών και συχνά το παράνομο κέρδος.

Η κεφαλίδα ενός μηνύματος ηλεκτρονικού ταχυδρομείου (Mail Header), όπως θα περιγραφεί στη συνέχεια, περιέχει αρκετή πληροφορία ώστε να διαπιστώσει κανείς την αξιοπιστία ενός email. Έτσι, ο κυριότερος σκοπός της εργασίας αυτής είναι να δημιουργηθεί μια εφαρμογή μέσω της οποίας θα είναι εφικτή η ανάλυση ενός ή περισσότερων e-mail Headers μέσω της αξιοποίησης πολλών από τα πεδία τους.

Έτσι κατέστη απαραίτητο η δημιουργία ενός collector ο οποίος θα λαμβάνει τα mail headers με τη χρήση του JSON Web Token για την διασφάλιση της ακεραιότητας του μηνύματος και την εγκυρότητα του αποστολέα. Έπειτα για περισσότερα από ένα e-mail header τα μηνύματα εισέρχονται σε ουρά εργασιών μέσω του Celery. Επόμενη διαδικασία είναι η μεταφορά των κεφαλίδων μέσω της ουράς προς τον analyzer. Η γλώσσα προγραμματισμού που επιλέχθηκε για τη δημιουργία του collector και του analyzer είναι η python. Σκοπός του τελευταίου είναι η ανάλυση των πεδίων και εξαγωγή χρήσιμων στοιχείων από το Mail Header και η αποθήκευσή τους σε MySQL και Mongodb βάση δεδομένων καθώς και σε ευρετήρια στο ElastiSearch. Παράλληλα, επιπρόσθετη παραμετροποίηση των πεδίων πραγματοποιήθηκε μέσω του Logstash. Τέλος η οπτικοποίηση των δεδομένων έγινε μέσω της δημιουργίας dashboard μέσω του Kibana.

Το Kibana προσφέρει τη δυνατότητα ανάλυσης των δεδομένων που έχουν εισαχθεί στην εφαρμογή με σκοπό την εύρεση ανεπιθύμητων μηνυμάτων που ως στόχο έχουν την προώθηση ιδεών ή προϊόντων όπου ο αποστολέας δεν αναμένει τέτοιου είδους μηνύματα. Παράλληλα μπορεί να γίνει αναζήτηση πληροφορίων αναφορικά με την εύρεση ύποπτης/ων: IP του αποστολέα, των hashes πιθανόν συνημμένων αρχείων ή και κακόβουλων υπερσυνδέσμων, θέτοντας σε εκμετάλλευση OSINT πηγές. Συνοπτικά ένας αναλυτής δύναται μέσω των δημιουργημένων dashboards να συμπεράνει αν κάποιο e-mail έχει σταλεί με κακόβουλο σκοπό.

1. Ηλεκτρονικό Ταχυδρομείο (E-mail)

Το ηλεκτρονικό ταχυδρομείο έχει καταστεί αναπόσπαστο μέρος των καθημερινών επιχειρηματικών δραστηριοτήτων σε όλες σχεδόν τις πτυχές του εμπορίου. Το τεράστιο εύρος των πληθυσμών που χρησιμοποιούν τακτικά το ηλεκτρονικό ταχυδρομείο μπορεί να αποδοθεί, σε μεγάλο βαθμό, στην προσβασιμότητα και τη γενική χρησιμότητα του e-mail. Το πρώτο παράδειγμα ηλεκτρονικού ταχυδρομείου μπορεί να βρεθεί στους υπολογιστές του MIT σε ένα πρόγραμμα που ονομάζεται "MAILBOX" το 1965 [1]. Οι χρήστες υπολογιστών του MIT θα μπορούσαν να αφήσουν μηνύματα με αυτό το πρόγραμμα σε υπολογιστές του πανεπιστημίου για άλλους χρήστες, οι οποίοι θα έβλεπαν τα μηνύματα την επόμενη φορά που θα συνδεόταν στον υπολογιστή.

Το Hotmail, το Yahoo και το Gmail είναι κάποιες από τις υπηρεσίες που δημιούργησαν ένα υπόβαθρο του Διαδικτύου και του ηλεκτρονικού ταχυδρομείου. Επένδυσαν σε πολλά δολάρια με σκοπό το μάρκετινγκ ώστε να βελτιώσουν την προσβασιμότητα και να γνωρίσουν σ' ένα ευρύτερο κοινό τα οφέλη του World Wide Web.

1.1 Δομή του e-mail

Ένα μήνυμα ηλεκτρονικού ταχυδρομείου αποτελείται από τρία βασικά στοιχεία: **τον φάκελο (envelope), την κεφαλίδα (Header) και το σώμα του μηνύματος**.

- Ο φάκελος είναι κάτι που ένας χρήστης ηλεκτρονικού ταχυδρομείου δεν θα δει ποτέ, δεδομένου ότι αποτελεί μέρος της εσωτερικής διαδικασίας μέσω της οποίας δρομολογείται ένα μήνυμα ηλεκτρονικού ταχυδρομείου.
- Το σώμα του μηνύματος είναι το μέρος που βλέπει ο χρήστης καθώς είναι το πραγματικό περιεχόμενο του μηνύματος που περιέχεται στο μήνυμα ηλεκτρονικού ταχυδρομείου.
- Το header, η τρίτη συνιστώσα ενός e-mail, θα αναλυθεί σ' αυτό το κεφάλαιο και είναι αναμφισβήτητα το πιο ενδιαφέρον κομμάτι ενός μηνύματος ηλεκτρονικού ταχυδρομείου.

Ορισμένα πεδία στις κεφαλίδες (headers) είναι υποχρεωτικά, όπως τα πεδία FROM, TO και DATE. Άλλα είναι προαιρετικά, όπως το SUBJECT και το CC, αλλά πολύ συχνά εμφανίζονται στο header. Κάθε φορά που ένα μήνυμα μεταφέρεται από έναν χρήστη σ' έναν άλλον (δηλ. όταν αποστέλλεται ή προωθείται), το μήνυμα συνοδεύεται από την ημερομηνία/ώρα (date/time) σύμφωνα με το MTA(Mail transer Agent). Το Date/Time, το FROM, το TO και το SUBJECT, είναι κάποια από τα πολλά πεδία που περιέχονται στην κεφαλίδα του εκάστοτε e-mail και προηγούνται του σώματος ενός ηλεκτρονικού ταχυδρομείου. Το E-mail Header θα πρέπει πάντα να διαβάζεται από κάτω προς τα πάνω.

Πρόσβαση στο Mail Header:

Οι περισσότερα mail clients επιτρέπουν την πρόσβαση στην κεφαλίδα του μηνύματος. Ένα δημοφιλές πρόγραμμα ηλεκτρονικού ταχυδρομείου είναι το Google Mail (GMail) Webmail. Για την προβολή της κεφαλίδας ενός e-mail ο χρήστης πρέπει να συνδεθεί στο λογαριασμό του και έπειτα να ανοίξει το μήνυμα. Στη συνέχεια, πρέπει να κάνει κλικ στο "κάτω βέλος" στην επάνω δεξιά γωνία του μηνύματος και να επιλέξει "Προβολή αρχικού" ή "Show Original". Τότε, θα δει την πλήρη κεφαλίδα του μηνύματος.

2. Πεδία ενός Mail Header σύμφωνα με το IANA

Τα επιμέρους πεδία των κεφαλίδων που θα τύχουν επεξεργασίας στα πλαίσια της εργασίας έχουν προσδιοριστεί σύμφωνα με την καταγραφή τους στην ιστοσελίδα του **IANA** με βάση τα σχετικά RFCs (τα RFC καλύπτουν πολλές πτυχές της δικτύωσης ηλεκτρονικών υπολογιστών, συμπεριλαμβανομένων των πρωτοκόλλων, των διαδικασιών και των προγραμμάτων). Η IANA είναι μία μη κερδοσκοπική ιδιωτική αμερικανική εταιρεία, η οποία είναι υπεύθυνη για τον συντονισμό ορισμένων βασικών στοιχείων του διαδικτύου. Μερικά από αυτά είναι η εποπτεία της καθολικής κατανομής διευθύνσεων IP, η κατανομή αριθμών αυτόνομου συστήματος, η διαχείριση του root zone στο Domain Name System (DNS) κι άλλων στοιχείων που σχετίζονται με το Πρωτόκολλο Διαδικτύου. [2]

Μερικά από τα σημαντικότερα πεδία της κεφαλίδας αλφαβητικά είναι:

Authentication-Results: περιέχει κυρίως τα πεδία spf, dkim και dmarc που περιγράφονται στην ενότητα 4.

Παράδειγμα δομής του πεδίου:

```
Authentication-Results: example.com; spf=pass smtp.mailfrom=example.net

Received: from dialup-1-2-3-4.example.net

        (dialup-1-2-3-4.example.net [192.0.2.200])

        by mail-router.example.com (8.11.6/8.11.6)

        with ESMTP id g1G0r1kA003489;

        Fri, Feb 15 2002 17:19:07 -0800

From: sender@example.net

Date: Fri, Feb 15 2002 16:54:30 -0800

To: receiver@example.com

Message-Id: <12345.abc@example.net>

Subject: here's a sample
```

Σ’ αυτό το mail header παρατηρείται ότι το πρωτόκολλο επικύρωσης (spf) του ηλεκτρονικού ταχυδρομείου επαληθεύει ότι η εισερχόμενη αλληλογραφία από ένα domain προέρχεται από μια διεύθυνση IP που είναι εξουσιοδοτημένη από τους διαχειριστές αυτού του domain. Έτσι, χαρακτηρίζεται ως ‘pass’ σε αντίθετη περίπτωση το υποπεδίο spf θα χαρακτηριζόταν ως ‘fail’ [3].

Bcc: (Blind Carbon Copy) περιέχει τις διευθύνσεις των παραληπτών του μηνύματος των οποίων οι διευθύνσεις δεν αποκαλύπτονται σε άλλους παραλήπτες του μηνύματος.

```
Bcc: [address-list ]
```

Cc: (Carbon Copy) περιέχει τις διευθύνσεις άλλων που πρόκειται να λάβουν το μήνυμα, αν και το περιεχόμενο του μηνύματος μπορεί να μην απευθύνεται σε αυτούς.

Οι διευθύνσεις παραληπτών ενός μηνύματος ηλεκτρονικού ταχυδρομείου παρατίθενται σε ένα ή περισσότερα από τα πεδία "From", "CC" και "BCC". Συνήθως οι spammers προτιμούν να χρησιμοποιούν το πεδίο "BCC" για να αποστέλλουν μηνύματα spam σε μεγάλο αριθμό παραληπτών, ενώ ταυτόχρονα κανένας από τους παραλήπτες δεν μπορεί να λάβει τη λίστα των διευθύνσεων που συλλέγονται από τους αποστολείς ανεπιθύμητης αλληλογραφίας, επειδή ο διακομιστής SMTP στέλνει ξεχωριστό μήνυμα ηλεκτρονικού ταχυδρομείου σε καθέναν από τους παραλήπτες που παρατίθενται στο πεδίο "BCC" και κάθε παραλήπτης δεν έχει πληροφορίες σχετικά με τους άλλους παραλήπτες.

Date: αντιπροσωπεύει την ημερομηνία και την ώρα του μηνύματος ηλεκτρονικού ταχυδρομείου όταν αυτό αποστέλλεται από τον αποστολέα στο Agent User Mail (MUA). Πρέπει να αναφερθεί ότι ο χρόνος που καταγράφεται σε αυτό το πεδίο βασίζεται στη θέση του διακομιστή αλληλογραφίας του αποστολέα ο οποίος θα μπορούσε να ανήκει σε μια ζώνη ώρας διαφορετική από εκείνη του παραλήπτη. Για παράδειγμα: Date: Wed, 14 Mar 2018 16:10:09 +0000 EET.

Dkim: η χρησιμότητα αυτού του πεδίου αναλύεται στην ενότητα 4.2. Παράδειγμα ως προς την δομή του καταγράφεται παρακάτω:

```
DKIM-Signature: v=1; a=rsa-sha256; d=example.net; s=brisbane;

c=relaxed/simple; q=dns/txt; l=1234; t=1117574938;

x=1118006938; h=from:to:subject:date:keywords:keywords;
bh=MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0NTY3ODkwMTI=;
b=dzdVyOfAKCdLXdJOc9G2q8LoXSlEniSbav+yuU4zGeeruD00lszZVoG4ZHRNiYzR
```

Όπου ‘v’ είναι η έκδοση υπογραφής, ‘a’ είναι ο αλγόριθμος υπογραφής, ‘d’ είναι το domain, ‘s’ είναι ο επιλογέας(selector), ‘c’ είναι ένας αλγόριθμος κανονικοποίησης για το header και το body του e-mail, ‘q’ είναι η προεπιλεγμένη μέθοδος ερωτήματος, ‘l’ είναι το μήκος του κανονικοποιημένου μέρους του σώματος που έχει υπογραφεί, ‘t’ είναι η χρονική σήμανση υπογραφής, ‘x’ είναι ο χρόνος λήξης του, και ‘h’ είναι η λίστα των υπογεγραμμένων πεδίων κεφαλίδας που επαναλαμβάνονται για πεδία που εμφανίζονται πολλές φορές.

From: τα πεδία δημιουργίας ενός μηνύματος αποτελούνται από το πεδίο ‘from’, το πεδίο ‘sender’ (όταν υπάρχει) και προαιρετικά από το πεδίο ‘reply-to’. Το πεδίο ‘from’ αποτελείται από το όνομα πεδίου “From” και μια λίστα διαχωρισμένη με κόμματα με μία ή περισσότερες διευθύνσεις e-mail. Το πεδίο "From:" προσδιορίζει τους συντάκτες του μηνύματος, δηλαδή τα mailbox των προσώπων ή των συστημάτων που είναι υπεύθυνα για τη σύνταξη του μηνύματος.

Message-ID: περιέχει ένα μοναδικό αναγνωριστικό μηνύματος. Η μοναδικότητα του αναγνωριστικού μηνύματος είναι εγγυημένη από τον κεντρικό υπολογιστή (host) που το δημιουργεί. Το πεδίο "Message-ID" είναι αναγνωριστικό από το μηχάνημα, το οποίο λαμβάνει το όνομα του μηχανήματος, την ημερομηνία και ώρα του μηνύματος ηλεκτρονικού ταχυδρομείου όταν αποστέλλεται. Για παράδειγμα:

```
From: John Doe <jdoe@machine.example>

To: Mary Smith <mary@example.net>

Subject: Saying Hello

Date: Fri, 21 Nov 1997 09:55:06 -0600

Message-ID: <1234@local.machine.example>
```

Received: κάθε μήνυμα ηλεκτρονικού ταχυδρομείου μπορεί να περιέχει περισσότερα από ένα πεδία "Received". Αυτό το πεδίο χρησιμοποιείται συνήθως για την παρακολούθηση της πορείας ενός μηνύματος ηλεκτρονικού ταχυδρομείου, διαβάζοντάς το από κάτω προς τα πάνω. Το κάτω μέρος αντιπροσωπεύει τον πρώτο διακομιστή αλληλογραφίας που εμπλέκεται στη μεταφορά του μηνύματος και η κορυφή αντιπροσωπεύει την πιο πρόσφατη, ενώ κάθε ενδιάμεσο πεδίο 'Received' αντιπροσωπεύει την μεταβίβαση μεταξύ των μηχανών. Ως εκ τούτου, κάθε νέο πεδίο 'Received' προστίθεται στην κορυφή της στοίβας για κάθε κεντρικό υπολογιστή που λαμβάνει το μήνυμα ηλεκτρονικού ταχυδρομείου και θα το μεταφέρει και σε όποιον φιλοξενεί το μήνυμα θα παραδοθεί, συνοδευόμενο από την ώρα και την ημερομηνία παραλήφθηκε.

```
Received: from node.example by x.y.test; 21 Nov 1997 10:01:22 -0600
```

Received-SPF: είναι το πεδίο αποτελεσμάτων που καθορίστηκε αρχικά για το SPF. Προορίζεται να συμπεριλάβει αρκετές πληροφορίες για να επιτρέψει την ανασυγκρότηση της αξιολόγησης SPF του μηνύματος, ενώ το 'Authentication-Results' έχει σχεδιαστεί μόνο για να αναμεταδίδει το ίδιο το αποτέλεσμα και τις σχετικές λεπτομέρειες πιθανής χρήσης στους τελικούς χρήστες. Για παράδειγμα:

```
Received-SPF: pass (mybox.example.org: domain of
                   myname@example.com designates
                   192.0.2.1 as permitted sender)
  receiver=mybox.example.org; client-ip=192.0.2.1;
  envelope-from="myname@example.com";     helo=foo.example.com;
```

Return-Path: μερικές φορές αποκαλείται "bounce address" ή "envelope sender address". Αυτή πρέπει να είναι η διεύθυνση από την οποία προέρχεται ένα μήνυμα και είναι η διεύθυνση στην οποία αποστέλλονται τυχόν ανεπιθύμητες ειδοποιήσεις μηνυμάτων ("bounces").

Ένα πράγμα που συχνά μπερδεύει, είναι η διαφορά μεταξύ της διεύθυνσης "Return-Path" και της διεύθυνσης κεφαλίδας "From:". Πρόκειται για δύο διαφορετικά πεδία, καθώς οι διακομιστές καθορίζουν το Return-Path. Οι διακομιστές επιβάλλουν το πεδίο "Return-Path" για δύο κύριους λόγους.

Πρώτα απ 'όλα, εάν οποιοσδήποτε χρήστης έστελνε μηνύματα χρησιμοποιώντας άλλο domain name, πολλοί διακομιστές λήψης θα θεωρούσαν ότι τα μηνύματά μας ήταν ανεπιθύμητα με σκοπό την πλαστογράφηση κάποιας διεύθυνσης. Για παράδειγμα αν υποθέσουμε ότι μόνο οι διακομιστές της AOL πρέπει να στέλνουν μηνύματα από διευθύνσεις που τελειώνουν στο "@aol.com", τότε πολλοί διακομιστές θα απέρριπταν το μήνυμα το οποίο είχε διαφορετικό domain.

Ο δεύτερος λόγος είναι ότι συμβάλλει στην ελαχιστοποίηση του spam που αποστέλλεται από το δίκτυό ενός χρήστη. Περιστασιακά, οι αποστολείς ανεπιθύμητης αλληλογραφίας προσπαθούν να εκμεταλλευτούν το μη-ασφαλές λογισμικό των server ώστε να στέλνουν χιλιάδες μηνύματα spam. Όταν το κάνουν αυτό, χρησιμοποιούν σχεδόν πάντοτε πλαστές διευθύνσεις "Return-Path". Όμως, τα φίλτρα εξερχόμενης αλληλογραφίας ανιχνεύουν το μη έγκυρο Return-Path και σταματούν την αλληλογραφία πριν βγει από το δίκτυό μας, διασφαλίζοντας ότι οι άλλοι ISP δεν αποκλείουν τους διακομιστές αλληλογραφίας μας (και το νόμιμο e-mail).

Subject: περιέχει έναν περιορισμένο αριθμό χαρακτήρων όπως περιγράφεται στο RFC 822 (*βλ.* [4]) και στο RFC 2822 (*βλ.* [5]). Περιέχει το θέμα ή μια σύνοψη του μηνύματος ηλεκτρονικού ταχυδρομείου.

To: τα πεδία προορισμού ενός μηνύματος αποτελούνται από τρία πιθανά πεδία: το όνομα πεδίου, το οποίο είναι είτε "To", "Cc" ή "Bcc", ακολουθούμενο από μία ή περισσότερες διευθύνσεις. Το πεδίο "To:" περιέχει τη διεύθυνση του κύριου παραλήπτη του μηνύματος.

X-Mailer: είναι ένα προαιρετικό πεδίο στην κεφαλίδα του μηνύματος ηλεκτρονικού ταχυδρομείου το οποίο περιέχει το πρόγραμμα ηλεκτρονικού ταχυδρομείου που χρησιμοποιείται για τη δημιουργία του μηνύματος ηλεκτρονικού ταχυδρομείου. Σε αυτό το πεδίο, καταγράφεται ο πελάτης ηλεκτρονικού ταχυδρομείου ή το όνομα και η έκδοση MUA.

3. Ανεπιθύμητη αλληλογραφία (Spam e-mail)

Ανεπιθύμητη αλληλογραφία (spam e-mail) ονομάζεται η μαζική αποστολή ηλεκτρονικών μηνυμάτων και χαρακτηρίζονται ως unsolicited messages. Με την φράση unsolicited messages, ορίζονται τα ανεπιθύμητα μηνύματα, δηλαδή τα μηνύματα που δεν έχει ζητήσει ένας χρήστης ή δεν περιμένει να λάβει από τον αποστολέα του μηνύματος.

Τα περισσότερα spam μηνύματα αφορούν κυρίως διαφημιστικούς λόγους, για την προώθηση προϊόντων ή ιδεών. Πολλά από αυτά μπορεί να περιέχουν κακόβουλα links που φαίνονται να είναι οικείες ιστοσελίδες αλλά στην πραγματικότητα οδηγούν σε ιστότοπους ηλεκτρονικού "ψαρέματος" (phishing) ή ιστότοπους που φιλοξενούν κακόβουλα προγράμματα. Τα διαφημιστικά μηνύματα που δεν έχουμε ζητήσει μπορεί να μας «ενοχλήσουν» σε πολλά σημεία επαφής της ιδιωτικής μας ζωής (κινητό τηλέφωνο, ηλεκτρονικό ταχυδρομείο, ιστοσελίδα κοινωνικής δικτύωσης). Μπορεί να είναι:

- μια διαφημιστική καμπάνια προώθησης ενός προϊόντος
- μηνύματα που αγγίζουν ευαίσθητες πτυχές της προσωπικής μας ζωής (π.χ. το δικαίωμα του «εκλέγειν»), όπως μηνύματα από υποψήφιους βουλευτές, τα οποία ως επί το πλείστον βομβαρδίζουν τα κινητά μας τηλέφωνα κυρίως παραμονή των εκάστοτε εκλογών.

Το ηλεκτρονικό ταχυδρομείο ανεπιθύμητης αλληλογραφίας μπορεί επίσης να περιλαμβάνει κακόβουλα προγράμματα ως scripts ή άλλα συνημμένα εκτελέσιμα αρχεία με σκοπό την εξαγωγή προσωπικών δεδομένων (phishing), όπως ονόματα χρήστη, κωδικούς, αριθμούς πιστωτικής κάρτας, κ.λπ.. Μπορεί να περιλαμβάνει επίσης, ένα κακόβουλο μήνυμα με στόχο την οικονομική εξαπάτηση (scamming) μηνύματα φαινομενικά εμπορικά που παραπέμπουν σε ιστοσελίδες με κακόβουλο κώδικα (malware).

Σε αρκετές χώρες η αποστολή spam διώκεται όχι μόνο δικαστικά αλλά και διοικητικά. Για τις ευρωπαϊκές χώρες πχ. αρμόδιες για το spam (εκτός δικαστηρίων) είναι οι Αρχές Προστασίας Δεδομένων Προσωπικού Χαρακτήρα [6].

4. Τεχνικές για την αποφυγή και την ανίχνευση ανεπιθύμητων μηνυμάτων

4.1 Sender Policy Framework (SPF)

Το **Sender Policy Framework (SPF)** είναι ένα πρωτόκολλο επικύρωσης ηλεκτρονικού ταχυδρομείου που έχει σχεδιαστεί για να ανιχνεύει και να αποκλείει την πλαστογράφηση ηλεκτρονικού ταχυδρομείου, παρέχοντας έναν μηχανισμό που επιτρέπει στους συνδρομητές ανταλλαγής αλληλογραφίας να επαληθεύουν ότι η εισερχόμενη αλληλογραφία από ένα domain προέρχεται από μια διεύθυνση IP εξουσιοδοτημένη από τους διαχειριστές αυτού του domain. Η λίστα με εξουσιοδοτημένους κεντρικούς υπολογιστές αποστολής (sending hosts) και διευθύνσεις IP για ένα domain δημοσιεύεται στις εγγραφές του συστήματος ονομάτων τομέα (Domain Name System) για το συγκεκριμένο domain [7].

Τα μηνύματα ηλεκτρονικού ταχυδρομείου και το ηλεκτρονικό "ψάρεμα" (phishing) συχνά χρησιμοποιούν πλαστά "from" στην δομή του Mail Header και απατηλά domain για την αποφυγή ψεύτικων διευθύνσεων ηλεκτρονικού ταχυδρομείου του αποστολέα. Το σύστημα μπορεί να ανιχνεύσει αν ο διακομιστής αλληλογραφίας, ο οποίος θέλει να στείλει ένα μήνυμα στην αλληλογραφία των παραληπτών, είναι έγκυρος για τη διεύθυνση ηλεκτρονικού ταχυδρομείου του αποστολέα. Ο έλεγχος με SPF μπορεί να θεωρηθεί ως μία από τις πιο αξιόπιστες και απλές μεθόδους αντιμετώπισης spam.

Το αποτέλεσμα μπορεί να είναι:

Received-SPF: neutral

Received-SPF: pass

Εάν αποτύχει:

Received-SPF: fail

Το μήνυμα θα πρέπει να απορριφθεί από το mail exchanger του παραλήπτη [8].

4.2 DomainKeys Identified Mail (DKIM)

Το **DomainKeys Identified Mail (DKIM)** είναι μια μέθοδος επαλήθευσης μέσω ηλεκτρονικού ταχυδρομείου που έχει σχεδιαστεί για την ανίχνευση απατηλού ηλεκτρονικού ταχυδρομείου. Επιτρέπει στον παραλήπτη να ελέγξει ότι ένα μήνυμα ηλεκτρονικού ταχυδρομείου που ισχυρίζεται ότι προέρχεται από συγκεκριμένο domain εξουσιοδοτήθηκε πράγματι από τον κάτοχο αυτού του domain. Αποσκοπεί δηλαδή, να αποτρέψει τις πλαστές διευθύνσεις αποστολέων. Αυτή η τεχνική συχνά χρησιμοποιείται στο phishing και στο spam.

Πρακτικά, το DKIM επιτρέπει σε ένα domain να συσχετίζει το όνομά του με ένα μήνυμα ηλεκτρονικού ταχυδρομείου, τοποθετώντας σε αυτό μια ψηφιακή υπογραφή. Η επαλήθευση πραγματοποιείται χρησιμοποιώντας το δημόσιο κλειδί του υπογράφοντος που δημοσιεύτηκε στο DNS. Με την έγκυρη υπογραφή εγγυάται ότι ορισμένα τμήματα του μηνύματος ηλεκτρονικού ταχυδρομείου (ενδεχομένως συμπεριλαμβανομένων των συνημμένων) δεν έχουν τροποποιηθεί από την στιγμή που τοποθετήθηκε η υπογραφή. Συνήθως, οι υπογραφές DKIM δεν είναι ορατές στους τελικούς χρήστες και έχουν

τοποθετηθεί ή επαληθεύονται από την υποδομή του συστήματος και όχι από τους αποστολείς και τους παραλήπτες του μηνύματος [9].

4.3 Domain-based Message Authentication (DMARC)

Domain-based Message Authentication, Reporting and Conformance (DMARC) είναι ένα σύστημα επικύρωσης μέσω ηλεκτρονικού ταχυδρομείου που έχει σχεδιαστεί για τον εντοπισμό και την αποτροπή της πλαστογράφησης ηλεκτρονικού ταχυδρομείου. Σκοπός του είναι να καταπολεμήσει ορισμένες τεχνικές που χρησιμοποιούνται συχνά σε phishing και σε spam μηνύματα, όπως μηνύματα ηλεκτρονικού ταχυδρομείου με παραμορφωμένες διευθύνσεις αποστολέων που φαίνονται να προέρχονται από νόμιμους οργανισμούς.

Το DMARC έχει δημιουργηθεί με βάση δύο υπάρχοντες μηχανισμούς, το Sender Policy Framework (SPF) και το DomainKeys Identified Mail (DKIM) που αναφέρθηκαν παραπάνω. Επιτρέπει στον κάτοχο διαχειριστή του domain να δημοσιεύει μια πολιτική σχετικά με τον μηχανισμό (DKIM ή SPF ή και τα δύο) που χρησιμοποιείται κατά την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου από το domain και τον τρόπο με τον οποίο ο παραλήπτης πρέπει να αντιμετωπίσει αποτυχίες.

Επιπλέον, παρέχει έναν μηχανισμό αναφοράς των ενεργειών που εκτελούνται στο πλαίσιο αυτών των πολιτικών. Επομένως, συσχετίζει τα αποτελέσματα των DKIM και SPF και καθορίζει υπό ποιες συνθήκες το πεδίο FROM, το οποίο είναι συχνά ορατό στους τελικούς χρήστες, πρέπει να θεωρείται νόμιμο.

Σκοπός του dmarc είναι να βοηθήσει τους παραλήπτες του e-mail να καθορίσουν αν το υποτιθέμενο μήνυμα "συμβαδίζει" ή “ευθυγραμμίζεται” με αυτό που ο παραλήπτης γνωρίζει για τον αποστολέα. Εάν όχι, το DMARC περιλαμβάνει μία καθοδήγηση για το πως να διαχειρίζεται τα "μη ευθυγραμμισμένα" μηνύματα [10].

Παρόμοια με το SPF και το DKIM, αυτή η πολιτική βρίσκεται στο DNS. Μια τυπική εγγραφή DMARC στο DNS θα έχει την εξής μορφή:

```
_dmarc.domain.com TXT v=DMARC1\; p=reject\; pct=100\; rua=mailto:dmarc-reports@domain.com\;
```

Η παραπάνω εγγραφή ορίζει μια πολιτική απόρριψης (p = απόρριψη) 100% (pct = 100) εάν το e-mail δεν περάσει το DKIM ή το SPF.

Υπάρχουν 3 πιθανές πολιτικές DMARC: None (monitoring only), Quarantine και Reject.

1. Πολιτική παρακολούθησης (monitoring only): **p = none**

Η πολιτική «none», παρέχει μόνο πληροφορίες σχετικά με το ποιος στέλνει μηνύματα ηλεκτρονικού ταχυδρομείου για λογαριασμό ενός τομέα και δεν θα επηρεάσει τη δυνατότητα παράδοσης.

2. Πολιτική καραντίνας: **p = quarantine**

Τα e-mail που περνούν τους ελέγχους DMARc θα παραδοθούν στα κύρια εισερχόμενα του δέκτη. Η πολιτική «καραντίνας» μετριάσει τον αντίκτυπο της πλαστογράφησης παραδίδοντας στον παραλήπτη το μήνυμα ταξινομώντας το στον φάκελο ανεπιθύμητων μηνυμάτων.

3. Πολιτική απόρριψης: **p = reject**

Η τρίτη πολιτική είναι η πολιτική απόρριψης. Αυτή η πολιτική μετριάζει τον αντίκτυπο της πλαστογράφησης δεδομένου ότι η απόρριψη πολιτικής DMARC διασφαλίζει ότι όλα τα spoofing e-mails δεν θα σταλούν στα εισερχόμενα του δέκτη [11].

4.3.1 ARC

Ωστόσο, όλα τα μηνύματα δεν μεταδίδονται απευθείας από τον αποστολέα στον παραλήπτη. Ορισμένες υπηρεσίες, όπως οι λίστες αλληλογραφίας ή η προώθηση λογαριασμού (account forwarding), γνωστές και ως διαμεσολαβητές (intermediaries), λαμβάνουν ένα νόμιμο μήνυμα και ενδέχεται να κάνουν αλλαγές σε αυτό πριν τον στείλουν στον τελικό παραλήπτη. Έτσι, ενδεχομένως να οδηγήσουν σε αποτυχία ‘ευθυγράμμισης’ του SPF, του DKIM ή και του DMARC. Συνεπώς, το μήνυμα, παρά τη νομιμοποίησή του, μπορεί να μην παραδοθεί. [20]

Το παραπάνω πρόβλημα της μη έμμεσης ροής της αλληλογραφίας ήρθε να λύσει το ARC. Το ARC λοιπόν, συμβάλλει στη διατήρηση της αυθεντικοποίησης των αποτελεσμάτων μέσω ηλεκτρονικού ταχυδρομείου και επαληθεύει την ταυτότητα των μεσολαβητών του μηνύματος που πρόκειται να το διαβιβάσουν στον τελικό του προορισμό [12].

Υπάρχουν τρία βασικά στοιχεία του ARC:

- ARC Authentication Results header: κεφαλίδα που περιέχει αποτελέσματα επαλήθευσης του e-mail όπως του SPF, του DKIM και του DMARC
- ARC Signature: υπογραφή που μοιάζει με DKIM, λαμβάνει ένα στιγμιότυπο από τις πληροφορίες της κεφαλίδας του μηνύματος, συμπεριλαμβανομένου του ‘to’, ‘from’, ‘subject’ και του ‘body’ του μηνύματος
- ARC Seal: μια άλλη υπογραφή που μοιάζει με DKIM και περιλαμβάνει τις πληροφορίες της κεφαλίδας ARC Signature και της ARC Authentication Results

5. Αρχιτεκτονική υλοποίησης

Σκοπός της εργασίας είναι η αναπαράσταση και η συσχέτιση δεδομένων που θα υποδεικνύουν κακόβουλη ή μη επιθυμητή/αναμενόμενη συμπεριφορά μιας ηλεκτρονικής αλληλογραφίας. Για να πραγματοποιηθεί το τελευταίο θα πρέπει να προηγηθεί η συλλογή των πληροφοριών ή/και η αποθήκευση τους και η ανάλυση τους, όπου αυτή απαιτείται, πριν την οπτικοποίηση των δεδομένων. Το συναφές με τα παραπάνω όνομα που έχει δοθεί στην εφαρμογή είναι το sace (spam analysis and correlation engine).

Πριν περιγραφεί η αρχιτεκτονική λύσης, σκόπιμο είναι να δοθούν μερικά στοιχεία για το περιβάλλον στο οποίο υλοποιήθηκε η εργασία. Τα σχετικά αρχεία εκτελούνται σε λειτουργικό Ubuntu 18.04.5 LTS σε ένα virtual machine με public IP 195.x.x.x. Ο κύριος κώδικας είναι γραμμένος σε Python 2.7.17 και το web framework που χρησιμοποιήθηκε είναι το Flask 1.1.2. Για κάθε επόμενη υπηρεσία που αξιοποιήθηκε, θα δοθούν στη συνέχεια οι αντίστοιχες πληροφορίες αναφορικά μ' αυτήν.

5.1 Δημιουργία collector

Πρώτο βήμα είναι η συλλογή του mail header. Γι' αυτό το σκοπό δημιουργήθηκε ο collector, με το αντίστοιχο αρχείο να είναι το «collector.py». Ο collector «ακούει» στην πόρτα 5000 και το uri path είναι το /SendMail. Κύριος στόχος του αρχείου είναι να παραλαμβάνει τα mail headers και να τα αποστέλλει στον analyzer. Ο collector τρέχει εκμεταλλευόμενο το ssl έτσι ώστε η επικοινωνία μεταξύ του αποστολέα και του collector να είναι κρυπτογραφημένη. Συνεπώς έχουν δημιουργηθεί ένα public και ένα private key όπως περιγράφεται και στην εικόνα με σκοπό τη δημιουργία ενός self-signed πιστοποιητικού.

```
georgia@[redacted]:~/Desktop/MSc_SACE/sace/keys$ openssl req -x509 -newkey rsa:4096 -nodes -out collector_cert.pem -keyout collector_key.pe
m -days 365
Can't load /home/georgia/.rnd into RNG
139740744872384:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/home/georgia/.rn
d
Generating a RSA private key
.....................................++++
.........................++++
writing new private key to 'collector_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:Greece
string is too long, it needs to be no more than 2 bytes long
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Athens
Locality Name (eg, city) []:Pireaus
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SACE
Organizational Unit Name (eg, section) []:sace
Common Name (e.g. server FQDN or YOUR name) []:GeorMich
Email Address []:
georgia@kavourdistiri:~/Desktop/MSc_SACE/sace/keys$
```

Εικόνα 5.1 Δημιουργία self-signed πιστοποιητικού

Έτσι ένας αποστολέας μπορεί να στείλει ένα υπογεγραμμένο mail header σε json μορφή *(βλ. ενότητα 7)* στον collector, παραδείγματος χάριν με την εντολή:

```
curl -X POST --data-binary "@/mailheader.txt" -H "Content-Type:
application/json" https://195.x.x.x:5000/SendMail
```

όπου στο mail header βρίσκεται το η κεφαλίδα του e-mail που πρόκειται να σταλεί στον collector.

5.2 Δημιουργία analyzer

Καθώς ο “collector” παραλαμβάνει τα mail headers, σ’ ένα άλλο script θα γίνεται η ανάλυση αυτών. Η δημιουργία του «analyzer.py» είναι απαραίτητη αρχικά με σκοπό να δέχεται τα mail headers που στέλνονται από τον collector και δεύτερο βήμα είναι η ανάλυση αυτών. Ο analyzer «ακούει» στην 5001, το path του είναι /Analyzer και τρέχει τοπικά.

5.2.1 Δημιουργία Βάσης Δεδομένων σε MySQL

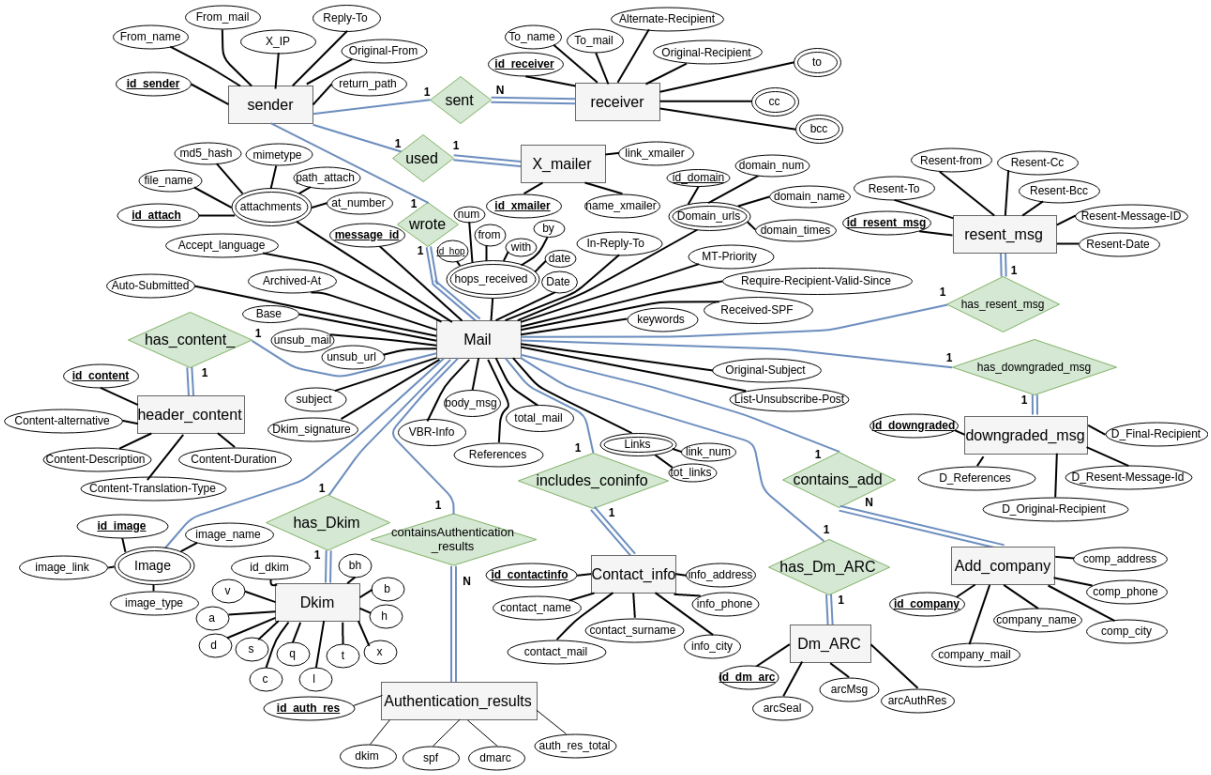
Απαραίτητη διαδικασία για την αποθήκευση των πεδίων του mail header είναι η δημιουργία μίας βάσης δεδομένων. Η βάση ονομάστηκε ‘spam_mail_db’ και είναι γραμμένη σε SQL κώδικα. Σύμφωνα με την περιγραφή των headers, όπως παρουσιάστηκαν παραπάνω, δημιουργήθηκαν 20 πίνακες για την αποθήκευση των πεδίων ενός mail header.

Πιο αναλυτικά, για την αποτελεσματική κατασκευή μιας βάσης δεδομένων (ΒΔ) ακολουθήθηκαν τα ακόλουθα βήματα:

1) Συλλογή και Ανάλυση Απαιτήσεων
2) Εννοιολογικός Σχεδιασμός
3) Σχεσιακό Σχήμα της ΒΔ
4) Υλοποίηση της ΒΔ

Το βήμα 1) περιλαμβάνει την ανάλυση των πεδίων των e-mails σύμφωνα με το IANA στην προηγούμενη ενότητα. Εκεί, καθορίστηκαν ποια είναι τα απαραίτητα πεδία που πλαισιώνουν ένα mail header καθώς και τα επιμέρους χαρακτηριστικά αυτών ώστε να αποθηκευτούν στην βάση.

Στο βήμα 2) χρειάζεται η δημιουργία του εννοιολογικού σχήματος της βάσης δεδομένων. Ένα εννοιολογικό σχήμα περιλαμβάνει το διάγραμμα Οντοτήτων-Συσχετίσεων (Entity-Relationship (E-R) diagram). Ενδεικτικά το εννοιολογικό σχήμα της βάσης περιγράφεται στην ακόλουθη εικόνα.



Εικόνα 5.2 Σχεσιακό σχήμα βάσης δεδομένων

Στο βήμα 3) λοιπόν, μέσω του σχεσιακού σχήματος ορίζονται οι τελικές σχέσεις και κατά συνέπεια οι τελικοί πίνακες της βάσης δεδομένων. Κάθε οντότητα, σύμφωνα με τους Κανονικούς Τύπους Οντοτήτων του Σχεσιακού Μοντέλου Δεδομένων, αποτελεί μία σχέση στο σχεσιακό σχήμα και άρα έναν πίνακα στην βάση. Τα γνωρίσματα της οντότητας αποτελούν τις στήλες ή στοιχεία του πίνακα.

Στο βήμα 4) δημιουργείται η βάση σε **SQL** με την εντολή 'CREATE DATABASE spam_mail_db;'. Για την δημιουργία του πίνακα 'Sender' εκτελείται ο παρακάτω κώδικας:

```
1.  CREATE TABLE Sender(
2.  id_sender BIGINT NOT NULL AUTO_INCREMENT,
3.  from_name VARCHAR(50),
4.  from_mail VARCHAR(200) NOT NULL,
5.  original_from VARCHAR(200),
6.  X_IP VARCHAR(45),
7.  return_path VARCHAR(200),
8.  PRIMARY KEY (id_sender));
```

Το id_sender ορίζεται ως AUTO_INCREMENT καθώς επιτρέπει την αυτόματη δημιουργία ενός μοναδικού αριθμού όταν εισάγεται νέα εγγραφή σε έναν πίνακα. Ύστερα από την δημιουργία του πίνακα πρέπει να οριστεί και μία κωδικοποίηση η οποία να εξυπηρετεί τον σκοπό μας. Συγκεκριμένα, χρησιμοποιήθηκε το utf8mb4, το οποίο υποστηρίζει και ελληνικούς χαρακτήρες. Παρόμοια διαδικασία πραγματοποιήθηκε και για τους υπόλοιπους πίνακες.

5.2.2 Εισαγωγή των headers στην MySQL βάση δεδομένων

Η σύνδεση του αρχείου της python “analyzer.py” με την MySQL βάση δεδομένων υλοποιείται μέσω της βιβλιοθήκης ‘MySQL’. Για την εισαγωγή της βιβλιοθήκης αναγκαία είναι η αντίστοιχη βιβλιοθήκη. Η σύνδεση του analyzer με την βάση γίνεται με τον ακόλουθο τρόπο :

```
1. conn = mysqldb.connect(host= "x.x.x.x", user="...", passwd="...",    db="spam_mail_
   db" ,charset="utf8", use_unicode=True)
```

Στην παράμετρο “user” και “passwd” βρίσκονται τα διαπιστευτήρια του χρήστη, ενώ στην παράμετρο “db”, το όνομα της βάσης. Με τις παραμέτρους “charset” και “use_unicode“ ρυθμίζεται η κωδικοποίηση της βάσης. Επιπρόσθετα, με την μέθοδο “execute” της python εκτελούνται ερωτήματα ή εντολές με κώδικα SQL. Για παράδειγμα, η εισαγωγή των γνωρισμάτων στον πίνακα ‘sender’ μέσω της python εκτελείται με τον ακόλουθο κώδικα:

```
1. x_sender.execute("""INSERT INTO Sender (`from_name`, `from_mail`,  `original_from`,
    `X_IP`, `return_path`) VALUES (%s,%s,%s,%s,%s)""",(from_name,from_mail,orfrom,xip,
   ret_path))
```

Το x_sender αφορά την σύνδεσή της μεταβλητής αυτής με την βάση μέσω της μεθόδου ‘cursor()’. Στην πρώτη παρένθεση βρίσκονται οι στήλες του πίνακα “Sender” σύμφωνα με την βάση “spam_mail_db”. Παράλληλα, η μέθοδος “execute” απαιτεί να γνωρίζει τον τύπο μεταβλητής που θα έχει ως είσοδο. Στην δεύτερη παρένθεση λοιπόν, βρίσκονται πέντε παράμετροι τύπου string, όσες και οι στήλες του πίνακα Sender. Στην τελευταία παρένθεση, ορίζονται οι μεταβλητές της python, οι οποίες περιέχουν το αντίστοιχο πεδίο του mail header μετά το παρσάρισμα. Αντίστοιχη διαδικασία ακολουθείται και για τους υπόλοιπους πίνακες της βάσης.

6. Συλλογή και εξαγωγή χρήσιμων στοιχείων από το Mail Header (Extracted Intelligence)

6.1 Εξαγωγή και αποθήκευση των συνημμένων αρχείων

Βασικό τμήμα του σώματος ενός ηλεκτρονικού μηνύματος είναι ένα συνημμένο αρχείο που μπορεί να υπάρχει σ' αυτό. Η εξαγωγή του συνημμένου είναι πολύ σημαντική καθώς στη συνέχεια θα εξεταστεί ως προς το πιθανό ιομορφικό περιεχόμενο του.

Ένα συνημμένο αρχείο αποτελεί μέρος του payload του e-mail καθώς αφορά τμήμα μεταδιδόμενων δεδομένων αυτού και μπορεί να είναι ένα απλό μήνυμα κειμένου ή μια δομημένη ακολουθία υπό-μηνυμάτων σε κωδικοποιημένη μορφή.

Για την εξαγωγή των συνημμένων χρησιμοποιήθηκε η βιβλιοθήκη 'emaildata' η οποία αφορά πακέτο της python για την εξαγωγή περιεχομένου από μηνύματα ηλεκτρονικού ταχυδρομείου. Έτσι, με την ακόλουθη επαναληπτική διαδικασία:

```
1. for content, filename, mimetype, message in Attachment.extract(message, False)
```

εξάγονται όλα τα συνημμένα αρχεία που επισυνάπτονται στο εκάστοτε e-mail. Παράλληλα, δημιουργούνται δύο νέα αρχεία με επέκταση .html και .ksh, όπου στο πρώτο αποθηκεύεται όλο το μήνυμα του e-mail, ενώ στο δεύτερο αποθηκεύεται το περιεχόμενο από όλα τα συνημμένα αρχεία. Τα αρχεία αποθηκεύονται στο ακόλουθο path: "sace/uploads". Το περιεχόμενο των συνημμένων καθώς και κάποια από τα χαρακτηριστικά τους όπως ο τύπος αρχείου, το hash τους κ.α. αποθηκεύονται στην MySQL βάση.

6.2 Διαφημιζόμενη εταιρεία ως αποστολέας ενός e-mail

Όπως έχει αναφερθεί προηγουμένως, τα περισσότερα spam μηνύματα στέλνονται κυρίως για διαφημιστικούς λόγους με σκοπό την προώθηση προϊόντων ή ιδεών. Σ' αυτήν την ενότητα λοιπόν, θα παρουσιαστεί μία προσπάθεια για την αναγνώριση του αποστολέα σε περίπτωση που το e-mail αφορά διαφημιζόμενη εταιρεία.

Ένα e-mail χωρίζεται σε δύο μέρη, το πρώτο μέρος πριν από το σύμβολο "@" αφορά το όνομα του χρήστη ενώ το δεύτερο μέρος μετά το σύμβολο "@", αφορά το domain. Στην περίπτωση που ένα e-mail ανήκει σε μία εταιρεία, συνήθως στο domain του e-mail αναγράφεται και το όνομα της. Μία εταιρεία εφόσον επιθυμεί να διαφημίσει τα προϊόντα της, τις περισσότερες φορές στέλνει στους παραλήπτες της και τους αντίστοιχους συνδέσμους που οδηγούν στην ιστοσελίδα της εταιρείας.

Το κοινό στοιχείο που έχουν τα περισσότερα urls που υπάρχουν στο body ενός μηνύματος, εάν πρόκειται για διαφημιζόμενη εταιρεία, είναι το domain name που περιέχεται σε κάθε url. Αυτό το domain name αν εμφανίζεται πολλές φορές σε αρκετά links, σε σχέση με τα συνολικά, τότε πιθανόν θα υποδεικνύει και το όνομα της εταιρείας.

Με σκοπό να αυξηθούν οι πιθανότητες της παραπάνω υπόθεσης, θεωρήθηκε επιπλέον ότι, αν το πολύ εμφανιζόμενο domain των url ταυτίζεται με το domain του e-mail του αποστολέα, τότε πρόκειται για διαφημιζόμενη εταιρεία και το domain υποδηλώνει το όνομα της εταιρείας.

Για παράδειγμα, ακολούθως φαίνεται ότι το domain του e-mail του sender το οποίο είναι «igi-global» βρέθηκε να είναι ίδιο με το «κύριο» domain όλων των link όπου είναι επίσης το «igi-global».

```
mysql> SELECT mh_timestamp,from_mail,domain_master,domain_times,tot_times FROM i FROM indexHeader LEFT JOIN Sender ON indexHeader.id_index=Send
er_iheader   LEFT JOIN Mail ON mail_send=Sender.id_sender    LEFT JOIN Detail_mail ON detail_mail=Mail.id_mail  where detail_mail= (SELECT MAX(
id_mail) FROM Mail);
+---------------------+---------------------------+---------------+--------------+-----------+
| mh_timestamp        | from_mail                 | domain_master | domain_times | tot_times |
+---------------------+---------------------------+---------------+--------------+-----------+
| 2020-12-17 22:13:40 | development@igi-global.com | igi-global    |           98 |       230 |
+---------------------+---------------------------+---------------+--------------+-----------+
1 row in set (0.00 sec)
```

Εικόνα 6.1 Εύρεση domain διαφημιζόμενης εταιρείας

Το «igi-global» ως domain name, βρέθηκε σε 98 από τα 230 links του e-mail.

6.3 Συλλογή ελεύθερα διαθέσιμης πληροφορίας (Open Source Intelligence-OSINT)

Το Open-source intelligence (OSINT) αναφέρεται σε δεδομένα που συλλέγονται από πηγές που είναι διαθέσιμες στο κοινό για να χρησιμοποιηθούν σε ένα πλαίσιο πληροφοριών. Ο όρος "open" αναφέρεται σε εμφανείς, διαθέσιμες στο κοινό πηγές _[13]. Σ' αυτήν την ενότητα, θα περιγραφεί το πώς αξιοποιήθηκε η ελεύθερα διαθέσιμη πληροφορία (OSINT) μέσω διαφόρων ιστοσελίδων που παρέχουν χρήσιμες πληροφορίες για τον εντοπισμό / έλεγχο της αξιοπιστίας της IP του αποστολέα, των πιθανόν υπερσυνδέσμων ή και των πιθανόν συνημμένων αρχείων που περιλαμβάνονται σ' ένα e-mail.

6.3.1 Έλεγχος για καταχωρημένη IP του αποστολέα σε μαύρη λίστα

Ο έλεγχος της διεύθυνσης IP σε μαύρη λίστα (blacklist check) υποδεικνύει εάν η IP είναι καταχωρημένη σε κυρίως online συστήματα για κακόβουλες/ανεπιθύμητες ενέργειες.

SANS:

Ο πρώτος έλεγχος διεξάγεται μέσω του API της ιστοσελίδας της **SSANS** (https://isc.SANS.edu/api/#ip). Το τελευταίο επιστρέφει μια σύνοψη των πληροφοριών που περιέχει η βάση δεδομένων της για μια συγκεκριμένη διεύθυνση IP. Χρησιμοποιώντας δηλαδή το ακόλουθο link "https://isc.SANS.edu/api/ip/IP_Sender?json", όπου "IP_Sender", η IP του αποστολέα, παρατηρούνται πόσα counts και πόσα attacks είχε η συγκεκριμένη IP σε json μορφή. Ως count θεωρείται ο συνολικός αριθμός των πακέτων που έχουν αποκλειστεί από αυτήν την IP. Ως attacks θεωρείται ο αριθμός των μοναδικών διευθύνσεων IP προορισμού για αυτά τα πακέτα. Για παράδειγμα για την IP '70.91.145.10' τα αποτελέσματα είναι τα παρακάτω:

{"ip":{"number":"70.91.145.10","count":10,"attacks":1 ...

Το count βρέθηκε '10' ενώ το attacks '1'. Συνεπώς, αυτές τις δύο τελευταίες παράμετροι ανακτώνται κάθε φορά για την IP του αποστολέα ενός Mail Header με σκοπό να εισαχθούν σ' έναν νέο πίνακα στην

βάση. Για τις καταχωρήσεις των δεδομένων που αφορούν reputation των επιμέρους πεδίων του Mail Header χρησιμοποιήθηκε MongoDB βάση δεδομένων όπως θα περιγραφεί στην ενότητα 8.5.2.

AbuseIPDB:

Ένας δεύτερος έλεγχος των blacklisted IP γίνεται μέσω της ιστοσελίδας της **AbuseIPDB** (https://www.AbuseIPDB.com/). Το AbuseIPDB παρέχει ένα δωρεάν API για την αναφορά και τον έλεγχο των διευθύνσεων IP. Κάθε μέρα οι webmasters, οι διαχειριστές συστημάτων και άλλοι επαγγελματίες τεχνολογιών πληροφορικής χρησιμοποιούν το API του για να αναφέρουν χιλιάδες διευθύνσεις IP που απασχολούν spamming, hacking και άλλες κακόβουλες δραστηριότητες σε πραγματικό χρόνο. Αυτό το API επιτρέπει τον έλεγχο διευθύνσεων IP από τη βάση δεδομένων της AbuseIPDB. Το API είναι ελεύθερο για χρήση, εφόσον έχει δημιουργηθεί ο αντίστοιχος λογαριασμός από τον χρήστη. Παράλληλα, υπάρχουν και κάποια όρια χρήσης. Όλοι οι δωρεάν λογαριασμοί έχουν ένα όριο 1.000 αναφορών και ελέγχων ανά ημέρα και οι λογαριασμοί Webmaster έχουν ένα όριο ποσοστού 3.000 αιτήσεων / ημέρα. Τέλος, για να αποφευχθεί η αυτόματη υποβολή διπλών αναφορών μέσω του API περιορίζουν κάθε λογαριασμό να αναφέρει την ίδια διεύθυνση IP μία φορά ανά 15 λεπτά. [14]

Συνεπώς, εφαρμόζοντας τα παραπάνω, σχετικά με το API του AbuseIPDB, αξιοποιούνται τα αποτελέσματα από το ακόλουθο link : https://www.AbuseIPDB.com/check/[IP]/json?key=[API_KEY], όπου API_KEY το μοναδικό κλειδί που ανήκει στον χρήστη έχοντας κάνει εγγραφή στο site του Abuse. Η πληροφορία που συλλέχθηκε απεικονίζεται και στην παρακάτω εικόνα

Εικόνα 6.2 Κακόφημη IP μέσω AbuseIPDB

και αφορά κυρίως την υποβαλλόμενη IP, το domain, κάποιες γεωγραφικές πληροφορίες και επίσης τον συνολικό αριθμό των αναφορών που την σχετίζουν με κακόβουλη δραστηριότητα.

FraudGuard:

Αντίστοιχη διαδικασία ακολουθείται και για το API που παρέχει το **FraudGuard**. Το FraudGuard αποτελεί μια υπηρεσία που έχει σχεδιαστεί για να παρέχει έναν εύκολο τρόπο επικύρωσης δεδομένων συλλέγοντας και αναλύοντας συνεχώς την κίνηση στο Διαδίκτυο σε πραγματικό χρόνο [15]. Όπως και στα παραπάνω η πληροφορία είναι δομημένη σε json. Π.χ.

```
{
    "isocode": "KR",
    "country": "Republic of Korea",
    "state": "Seoul",
    "city": "Seoul",
    "discover_date": "2018-12-11 07:00:45",
    "threat": "honeypot_tracker",
    "risk_level": "5"
}
```

Τα δύο σημαντικότερα πεδία που επιλέγονται να εξαχθούν σύμφωνα με το παραπάνω json είναι το πεδίο “threat” και το “risk level”. Το πρώτο αφορά την απειλή με την οποία σχετίζεται η υποβληθείσα IP και το δεύτερο είναι ένα επίπεδο στο οποίο ως μέγιστη τιμή υψηλού ρίσκου ορίζεται το 5.

AlientVault:

Τελευταία πηγή OSINT που επιλέγεται ως προς τον έλεγχο της IP του αποστολέα σε blacklist είναι το **Alienvault**. Σ’ αυτήν την περίπτωση ο έλεγχος πραγματοποιείται με τη βοήθεια του AlienVault-OTX/OTX-Python-SDK στο αντίστοιχο directory του github. Προϋπόθεση και πάλι ήταν η εγγραφή στη σελίδα του alienvault με σκοπό την απόκτηση ενός apikey.

Η πληροφορία που συλλέγεται είναι σχετική με κακόβουλη δραστηριότητα της συγκεκριμένης IPs καθώς ανακτώνται τα αντίστοιχα “tags” που χαρακτηρίζουν την IP. Επίσης συλλέγεται και ο αριθμός των pulses. Σκοπός των pulses είναι να παρέχουν μια σύνοψη της απειλής και την σχετική ένδειξη παραβίασης (IOC) που μπορεί να χρησιμοποιηθεί για τον εντοπισμό των απειλών. [16]

6.3.2 Έλεγχος συνημμένων αρχείων για ιομορφικό περιεχόμενο

Ένα συνημμένο αρχείο ενός e-mail μπορεί να είναι για παράδειγμα ένα αρχείο zip, word, excel κτλ. Ένα e-mail που περιέχει ένα επιβλαβές συνημμένο αρχείο δεν μπορεί να βλάψει έναν υπολογιστή αν δεν εκτελεστεί/ανοιχτεί το αρχείο. Συχνά, ένα κακόβουλο αρχείο μπορεί να μεταμφιεστεί ως έγγραφο του word ή κάποιου άλλου τύπου αρχείου. Οι επιτιθέμενοι μπορούν εύκολα να αλλάξουν μια επέκταση .exe ενός κακόβουλου αρχείου σε .doc.

Για τον έλεγχο ενός συνημμένου αρχείου χρησιμοποιήθηκε το API που παρέχει η ιστοσελίδα του VirusTotal. Το VirusTotal είναι μια ιστοσελίδα που δημιουργήθηκε από την ισπανική εταιρεία ασφάλειας Hispasec Sistemas. Ξεκίνησε τον Ιούνιο του 2004 και εξαγοράστηκε από την Google Inc. τον Σεπτέμβριο του 2012. Το VirusTotal αναλύει ύποπτα αρχεία, hashes, διευθύνσεις URL κ.α για τον εντοπισμό κακόβουλου δραστηριότητας. Αρχεία έως και 256 MB μπορούν να μεταφορτωθούν στον

ιστότοπο ή να σταλούν μέσω ηλεκτρονικού ταχυδρομείου. Επιπρόσθετα, το VirusTotal παρέχει ως δωρεάν υπηρεσία ένα δημόσιο API που επιτρέπει την αυτοματοποίηση μερικών από τις διαδικτυακές του δυνατότητες, όπως μεταφόρτωση και σάρωση αρχείων, υποβολή και σάρωση διευθύνσεων URL, πρόσβαση σε ολοκληρωμένες αναφορές σάρωσης και δημιουργία αυτόματων σχολίων σε διευθύνσεις URL. Ισχύουν ορισμένοι περιορισμοί για αιτήματα που υποβάλλονται μέσω του δημόσιου API, όπως το ότι μπορούν να υποβληθούν 4 αιτήματα (requests) ανά 1 λεπτό. [17]

Στα πλαίσια της παρούσας εργασίας, εξετάζονται τα md5 hashes των αρχείων που πιθανόν να επισυνάπτονται σ' ένα e-mail. Βρίσκοντας το md5 hash για κάθε αρχείο, γίνεται η υποβολή του στο VirusTotal με την βοήθεια των παρακάτω παραμέτρων :

```
1.  params = {'apikey': 'MY_API_KEY', 'resource': at_md5}
```

όπου 'MY_API_KEY' το API key που παρέχεται μετά την εγγραφή του χρήστη στο VirusTotal και 'at_md5' το hash του συνημμένου αρχείου.

Στη συνέχεια ακολουθείται η διαδικασία της εξαγωγής των πεδίων που θεωρούνται χρήσιμα ώστε να βγουν συμπεράσματα για την συμπεριφορά του αρχείου. Ένα από τα πεδία είναι το 'positives' όπου δείχνει πόσα αντιικά βρήκαν το συγκεκριμένο hash ότι σχετίζεται με κάποιον ιό. Έπειτα επιλέγεται το πεδίο "total" όπου αφορά τον αριθμό των αντιικών που σκάναραν το αρχείο. Τέλος, πραγματοποιείται η ανάκτηση της πληροφορίας που αφοράτην 'οικογένεια ιού' στην οποία ανήκει δηλαδή, ένας ιός. Για να επιτευχθεί αυτό γίνεται επεξεργασία των αποτελεσμάτων από τα αντιικά που κατέγραψαν ως κακόβουλο ένα αρχείο. Ακολουθείται η αφαίρεση των λέξεων που δημιουργούν «θόρυβο», όπως των stopwords (συνήθως αναφέρονται ως τις πιο συνηθισμένες λέξεις σ' ένα κείμενο) καθώς και άλλων λέξεων που εμφανίζονται συχνά στο περιεχόμενο, όπως σχετικές λέξεις με το VirusTotal και λέξεις που αναφέρονται σε αντιικά, παραδείγματος χάριν, όπως το positives, το total, το Antivirus, το AdAware κτλ. Έτσι, με την μέθοδο Counter() βρίσκεται η συχνότερη λέξη που αναφέρεται στην 'οικογένεια' του ιού.

6.3.3 Έλεγχος κακόβουλων υπερσυνδέσμων

Ο αποστολέας ενός επιβλαβούς e-mail μπορεί να εκμεταλλευτεί τους υπερσυνδέσμους ως εργαλείο επίθεσης. Σκοπός του επιτιθέμενου είναι απλά ο παραλήπτης να κάνει κλικ σε έναν κακόβουλο σύνδεσμο που του έχει στείλει μέσω του e-mail.

Αυτή η ενέργεια του χρήστη συνήθως θα παρατηρηθεί ως επακόλουθη συμπεριφορά μέσω ενός Spearphishing Link. Το Spearphishing με σύνδεσμο είναι μια συγκεκριμένη παραλλαγή του Spearphishing. Είναι διαφορετικό από άλλες μορφές Spearphishing διότι ο αποστολέας χρησιμοποιεί την χρήση συνδέσμων για τη λήψη κακόβουλου λογισμικού που περιέχεται σε e-mail. Επιλέγει δηλαδή αντί να επισυνάπτει κακόβουλα αρχεία στο ίδιο το e-mail, να τοποθετεί σ' αυτό κακόβουλα μη ανιχνεύσιμα link ώστε να αποφευχθεί ο εντοπισμός του ιομορφικού περιεχομένου του από τα εκάστοτε αντιικά. Οι επιτιθέμενοι έτσι ενδέχεται να αποκτήσουν πρόσβαση σε συστήματα των θυμάτων. [18]

Ο παραλήπτης σε μεγάλο βαθμό μπορεί να γίνει θύμα επίθεσης με την χρήση social engineering και έτσι κάνοντας κλικ σ' ένα link μπορεί να οδηγηθεί στην εκτέλεση κώδικα στο σύστημα του. Κάνοντας κλικ σε έναν σύνδεσμο μπορεί επίσης να οδηγηθεί σε άλλες τεχνικές εκτέλεσης, όπως την εκμετάλλευση ενός προγράμματος περιήγησης ή κάποια ευπάθεια εφαρμογών. Οι σύνδεσμοι ενδέχεται επίσης να οδηγήσουν τους χρήστες στη λήψη αρχείων που απαιτούν εκτέλεση μέσω κακόβουλου λογισμικού.

Για τον εντοπισμό τέτοιων κακόβουλων υπερσυνδέσμων, χρησιμοποιήθηκε το URLHaus. Το URLHaus είναι ένα έργο του “abuse.ch” με στόχο την κοινή χρήση κακόβουλων διευθύνσεων URL που χρησιμοποιούνται για τη διανομή κακόβουλου λογισμικού. [19]

Αξιοποιώντας το API του URLHaus γίνεται ανάκτηση των αποτελεσμάτων σχετικά με κάθε υποβληθέν url του e-mail. Τα πεδία που επιλέχθηκαν είναι τα tags για το σχετικό url, τα αποτελέσματα που μπορεί να υπάρχουν από το VirusTotal και η σχετική αναφορά του URLHaus μέσω link *(βλ. Ενότητα 10 & 11)*.

6.3.4 Εύρεση συντόμευσης υπερσυνδέσμων

Με παρόμοια λογική, όπως περιγράφηκε παραπάνω, ένας αποστολέας αυξάνοντας το επίπεδο εξαπάτησης, θα μπορούσε να χρησιμοποιήσει υπηρεσία συντόμευσης url (url shortening). Οι συντομεύσεις διευθύνσεων URL χρησιμοποιούνται συχνά από «εγκληματίες» του κυβερνοχώρου με σκοπό να μεταμφιέσουν ένα κακόβουλο link σ' ένα ανυποψίαστο, αποκρύπτοντας έτσι έναν πραγματικά κακόβουλο προορισμό.

Με σκοπό την εύρεση των συντομευμένων υπερσυνδέσμων χρησιμοποιείται η βιβλιοθήκη της python "unshortenit". Υποστηρίζει την «αποσυμπίεση» των παρακάτω συντομεύσεων: Adf.ly και σχετικά subdomain, Sh.st και Adfoc.us. Υποστηρίζει οποιαδήποτε διεύθυνση ανακατεύθυνσης 301 και διευθύνσεις ανακατεύθυνσης Meta Refresh. [20]

7. Json Web Token για την αποστολή του Mail Header

7.1 Ορισμός του JSON web token (JWT)

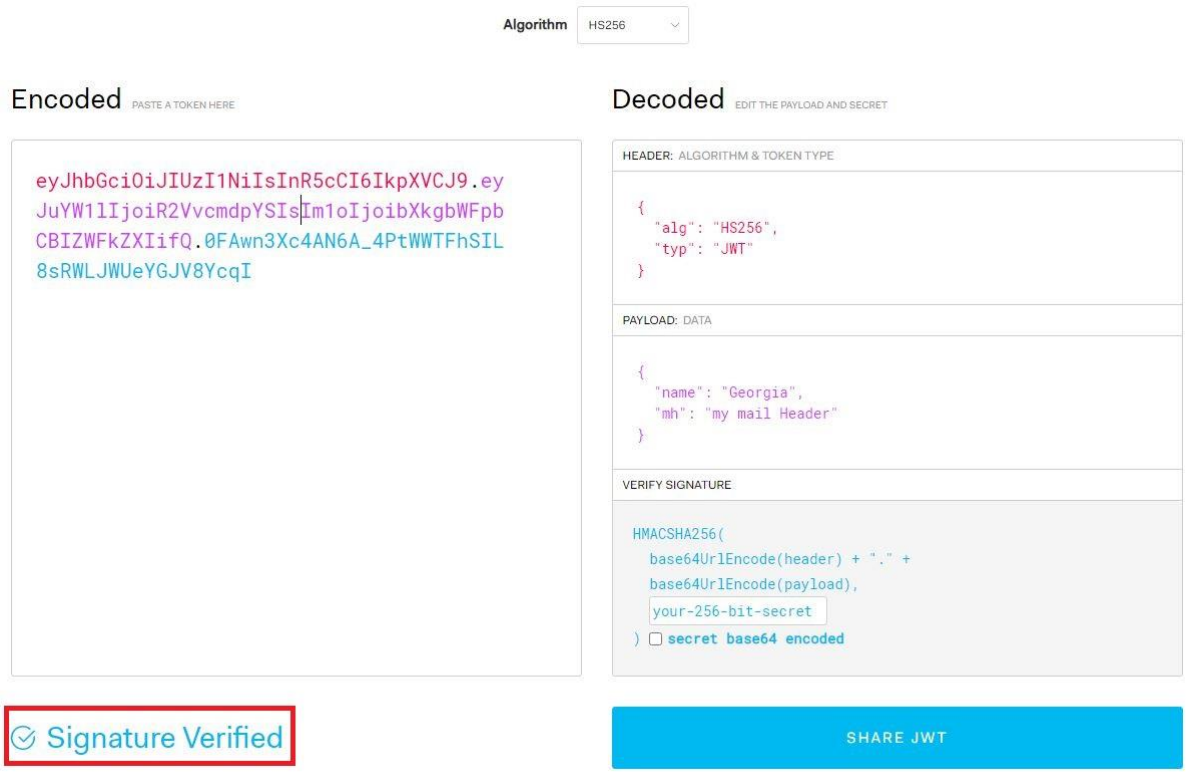
Το JSON web token (JWT) αποτελεί ένα πρότυπο που ορίζεται από το (RFC 7519 [21]) που επιτρέπει την ψηφιακή ασφαλή μετάδοση πληροφοριών μεταξύ δύο ή περισσότερων υπηρεσιών. Το περιεχόμενο των δεδομένων χρησιμοποιεί τη δομή JavaScript Object Notation (JSON).

Ένα JWT μπορεί να σταλεί είτε μέσω μιας διεύθυνσης URL, είτε μέσω μιας παραμέτρου POST είτε μέσα σε μια κεφαλίδα HTTP, λόγω του σχετικά μικρού μεγέθους του.

Το JWT αποτελείται από τρία μέρη:

• η κεφαλίδα(header), περιγράφει συνήθως τον αλγόριθμό που έχει χρησιμοποιηθεί και την δομή του token.
• το σύνολο δεδομένων (payload) ή σώμα, αφορά τα δεδομένα τα οποία είναι δομημένα σε JSON μορφή.
• η υπογραφή (signature) ή ο κωδικός ελέγχου ταυτότητας μηνύματος (the message authentication code) όπου συμπεριλαμβάνει το κωδικοποιημένο mail header και το payload σε base64url [22] [23].

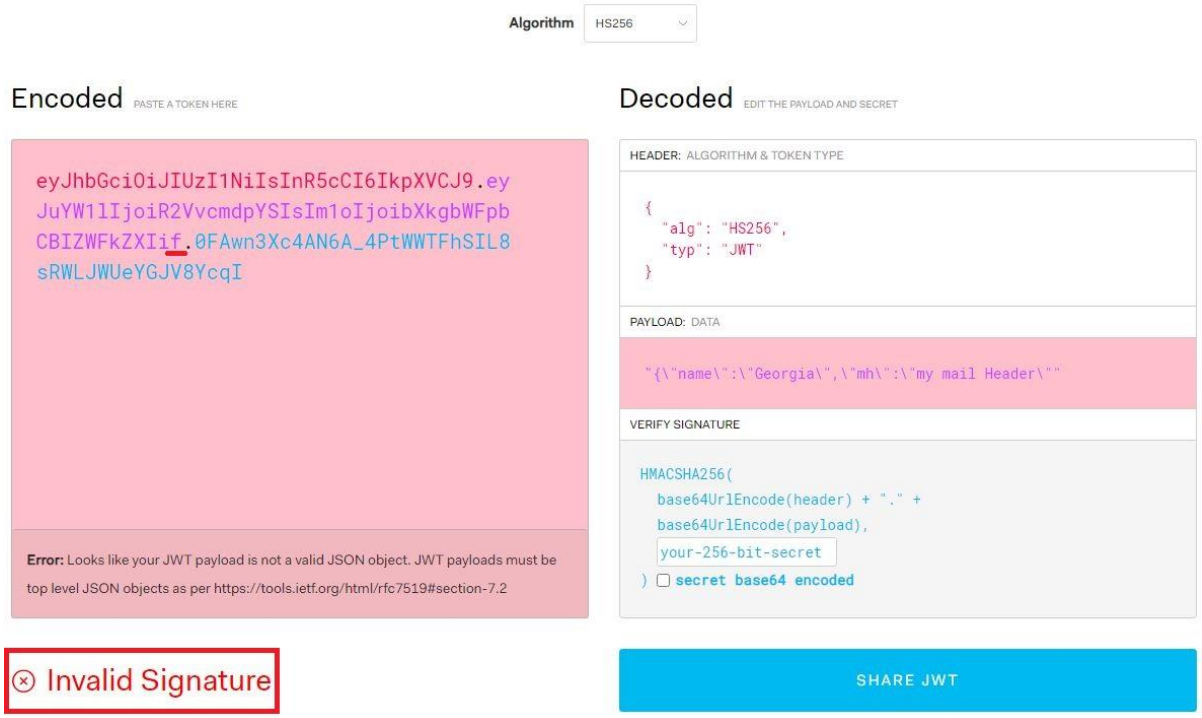
Παράδειγμα χρήσης του JWT για την κωδικοποίηση δεδομένων σύμφωνα με το online εργαλείο του «JWT.io» [24] :



Εικόνα 7.1 JWT – Έγκυρη υπογραφή

Κάθε κωδικοποιημένο πεδίο του JWT σε base64url είναι χωρισμένο από το σύμβολο της τελείας «.». Όπως φαίνεται και στην εικόνα 7.1, οι κόκκινοι κωδικοποιημένοι συμβολοχαρακτήρες στην αριστερή στήλη αντιστοιχίζονται με το header στην δεξιά στήλη. Αντίστοιχα οι μωβ συμβολοχαρακτήρες αφορούν το payload και οι γαλάζιοι το signature.

Αν για παράδειγμα αλλάξει ένα τουλάχιστον σύμβολο ή χαρακτήρας από το token, τότε η υπογραφή δεν θα είναι έγκυρη. Στην εικόνα 1.2 απεικονίζεται το παραπάνω token, αφού έχει αφαιρεθεί ο τελευταίος χαρακτήρας του payload.



Εικόνα 7.2 JWT – Μη έγκυρη υπογραφή

Στο συγκεκριμένο παράδειγμα με το τροποποιημένο token αποδεικνύεται η μη έγκυρη υπογραφή. Αξιοσημείωτο εδώ είναι να αναφερθεί πως με αυτόν τον τρόπο διαπιστώνεται και η ακεραιότητα (**integrity**) του μηνύματος.

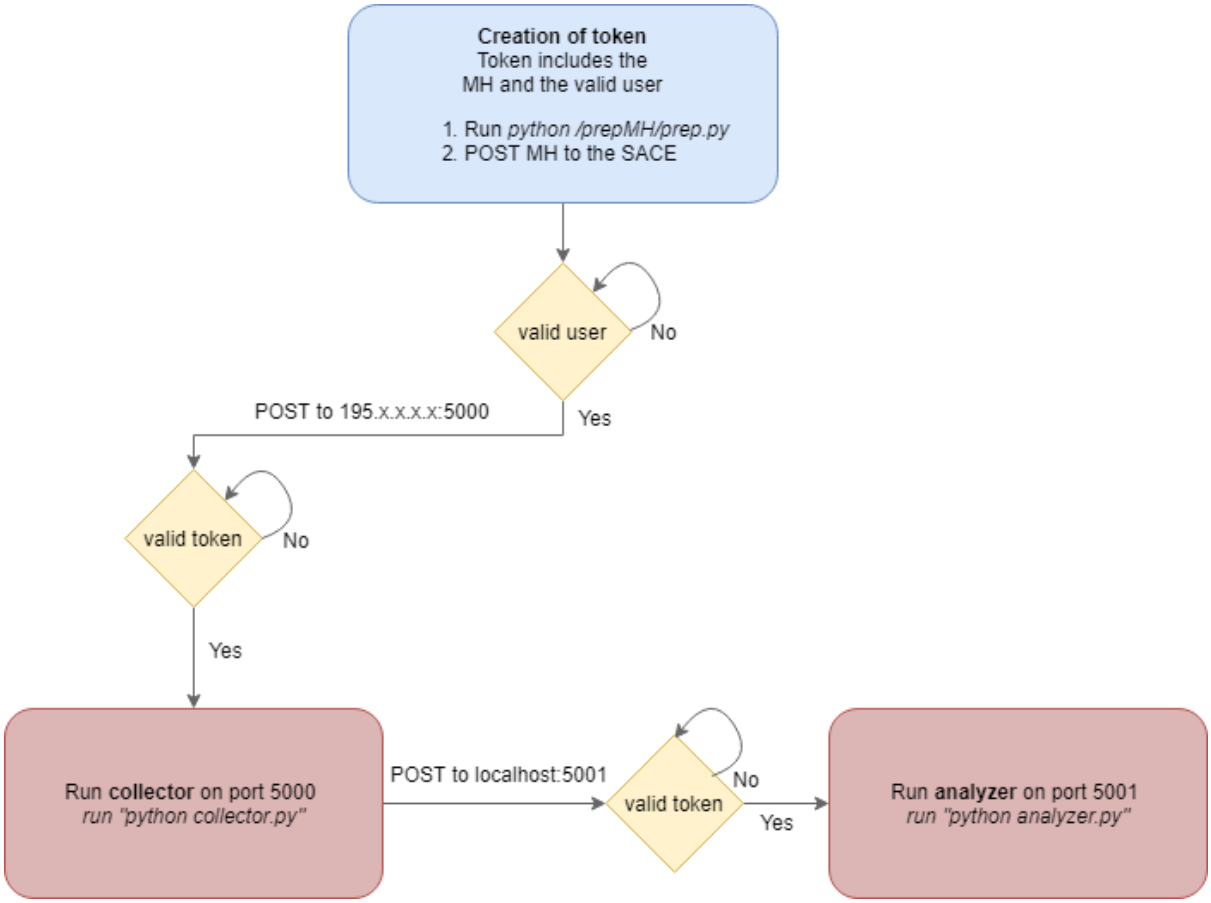
Σύμφωνα με την παραπάνω πρόταση, το JWT δίνει την δυνατότητα της αυθεντικοποίησης (**authentication**) του χρήστη ή μίας υπηρεσίας. Στο token δηλαδή μπορεί να εμπεριέχονται τα διαπιστευτήρια του χρήστη και στη συνέχεια αντί αυτών να αξιοποιείται το αντίστοιχο ID token ελέγχοντας επίσης την ακεραιότητα του μέσω της υπογραφής.

Συνοπτικά, το JWT είναι ένας καλός τρόπος για την **ασφαλή μετάδοση πληροφοριών** μεταξύ διάφορων υπηρεσιών, εφόσον ενεργοποιηθεί μηχανισμός κρυπτογράφησης, επειδή μπορούν να υπογραφούν, πράγμα που σημαίνει ότι διασφαλίζεται η ταυτότητα του αποστολέα. Επιπλέον, η δομή ενός JWT επιτρέπει την επαλήθευση της μη παραβίασης του περιεχομένου όπως προαναφέρθηκε.

7.2 Χρήση του JWT στην εφαρμογή sace

Η τεχνική του JWT όσον αφορά την εφαρμογή συμβάλλει κυρίως στην αυθεντικοποίηση του χρήστη ο οποίος στέλνει τα mail headers στον collector και για την επαλήθευση της ακεραιότητας του mail header με μία διαδικασία εξέτασης της εγκυρότητας της υπογραφής. Παρόμοια διαδικασία ακολουθείται και για την μεταφορά του mail header από τον collector προς τον analyzer όπως θα περιγραφεί στη συνέχεια.

Αρχικά, με σκοπό την κατανόηση της ανάγκης της χρησιμότητας του JWT, ακολούθως αποτυπώνεται η συνολική διαδρομή των δεδομένων που αφορά το e-mail header.



Εικόνα 7.3 **Διαδρομή του Mail header**

Ιεραρχικά παρατηρώντας την διαδρομή των data από πάνω προς τα κάτω, το πρώτο βήμα είναι η δημιουργία του token. Σε αυτό το βήμα θα χρησιμοποιηθεί το αρχείο «prep.py». Το συγκεκριμένο αρχείο θα δημιουργήσει το token εφόσον αυθεντικοποιηθεί αρχικά ο χρήστης. Έπειτα διαβάζει το επιθυμητό mail header απο ένα δεύτερο αρχείο, το 'mail.txt', και γράφει σ' αυτό το αντίστοιχο token σε json μορφή. Το token αποτελείται από τον **αυθεντικοποιημένο χρήστη** και το **mail header**.

Αν ο χρήστης είναι έγκυρος, τότε με ενέργεια αυτού μπορεί να σταλεί το mail header στο VM που φιλοξενεί το collector. Με παρόμοιες ενέργειες ένα νέο token στέλνεται από τον collector προς τον analyzer.

7.3 Έλεγχος εγκυρότητας του token μέσω της python

Αρχική διαδικασία για την υλοποίηση του JWT μηχανισμού με την χρήση της γλώσσας python είναι η δημιουργία υπογραφής δημοσίου κλειδιού **(Public-Key Signature)**.

Η χρήση του υπογεγραμμένου κλειδιού συμβάλλει στην ασφαλή αποθήκευση αυτού δίνοντας την δυνατότητα σε περισσότερες από μία υπηρεσίες να το χρησιμοποιούν. Έτσι οι υποκείμενες υπηρεσίες μπορούν να επαληθεύσουν το token χωρίς να έχουν πρόσβαση στο κλειδί. Το προαναφερθέν μπορεί να επιτευχθεί με την **κρυπτογράφηση δημόσιου κλειδιού**.

Η κρυπτογράφηση δημοσίου κλειδιού βασίζεται σε κλειδιά κρυπτογράφησης που έχουν δύο στοιχεία: ένα δημόσιο κλειδί (public key) και ένα ιδιωτικό κλειδί (private key). Το δημόσιο κλειδί όπως εύκολα συμπεραίνεται και από το όνομα του μπορεί να κοινοποιηθεί σε όλους. Δύο περιπτώσεις που μπορεί να επιτευχθεί η κρυπτογράφηση δημόσιου κλειδιού είναι η κρυπτογράφηση μηνυμάτων και η υπογραφή μηνύματος.

Ως προς το δεύτερο σενάριο, υπάρχουν αλγόριθμοι υπογραφής για την πραγματοποίηση του JWT μηχανισμού. Τα tokens δηλαδή υπογράφονται με το ιδιωτικό κλειδί του διακομιστή και, στη συνέχεια, μπορούν να επαληθευτούν από οποιονδήποτε χρησιμοποιεί το δημόσιο κλειδί του διακομιστή, το οποίο είναι ελεύθερα διαθέσιμο.

Στα πλαίσια της εργασίας ο αλγόριθμος υπογραφής που αξιοποιήθηκε είναι ο RS256, ο οποίος είναι η συντομογραφία του **RSA-SHA256**.

Απαραίτητο πακέτο προς εγκατάσταση είναι το :

```
pip install cryptography
```

Το επόμενο βήμα είναι η δημιουργία του δημόσιου και ιδιωτικού κλειδιού. Η δημιουργία κλειδιών πραγματοποιήθηκε με παρόμοιο τρόπο όπως σ' αυτόν της ενότητας 4.1 για την δημιουργία κλειδιών για την επικοινωνία με τον collector.

Με την περάτωση της εντολής, δημιουργούνται τα αντίστοιχα κλειδιά, το "mykey" και το "mykey.pub". Το πρώτο είναι το ιδιωτικό κλειδί, το οποίο θα χρησιμοποιηθεί για τη δημιουργία υπογραφής του token. Το αρχείο αυτό θα πρέπει να βρίσκεται σ' ένα προστατευμένο και μη προσπελάσιμο μέρος από μη επιτρεπτές υπηρεσίες/διαδικασίες. Το αρχείο .pub θα χρησιμοποιηθεί για την επαλήθευση του token. Δεδομένου ότι αυτό το αρχείο δεν έχει ευαίσθητες πληροφορίες, μπορεί ελεύθερα να βρίσκεται σε οποιονδήποτε κατάλογο.

Παράλληλα, επόμενο πακέτο προς εγκατάσταση είναι το

```
pip install pyjwt
```

Έπειτα, ακολουθεί η δημιουργία του token.

```
1.  import jwt
2.  private_key = open('mykey').read()
3.  token = jwt.encode(MailHeader, private_key, algorithm='RS256').decode('utf-8')
```

Το ιδιωτικό κλειδί RSA δηλώνεται ως όρισμα του private_key. Η τιμή αυτού του κλειδιού είναι ολόκληρο το περιεχόμενο του αρχείου "mykey". Ο αλγόριθμος όπως προαναφέρθηκε είναι ο RS256.

Έχοντας δημιουργήσει το token, στη συνέχεια γίνεται η επαλήθευση αυτού χρησιμοποιώντας το δημόσιο κλειδί "mykey.pub".

```
1.  public_key = open('mykey.pub').read()
2.  data = … # token
3.  payload = JWT.decode(data, public_key, algorithms=['RS256']
4.  email=payload['MailHeader']
5.  user=payload['user']
```

Στη μεταβλητή ‘email’ περιέχεται το mail header που βρέθηκε από το αποκωδικοποιημένο token σε json μορφή, ενώ στην μεταβλητή user βρίσκεται ο χρήστης ο οποίος έστειλε το ΜΗ προς τον collector.

Αντίστοιχη διαδικασία ακολουθείται και για την αποστολή των δεδομένων από τον collector προς τον analyzer με την χρήση του ίδιου ζεύγους κλειδιών.

8. Βασικές έννοιες και χρήση του ELK stack

Το ELK αποτελεί συντομογραφία του ElasticSearch, Logstash και του Kibana. Όπως εύκολα συμπεραίνεται το ELK αποτελείται από τρία εργαλεία με διαφορετική χρήση το καθένα. Το ακόλουθο διάγραμμα συμβάλλει στην κατανόηση του ELK stack.

Εικόνα 8.1 ELK stack

Αυτό το διάγραμμα δείχνει ότι το Logstash χρησιμοποιείται για την προώθηση δεδομένων απευθείας στο ElasticSearch. Αυτά τα δεδομένα δεν περιορίζονται σε δεδομένα καταγραφής, αλλά μπορούν να περιλαμβάνουν οποιοδήποτε τύπο δεδομένων. Το ElasticSearch αποθηκεύει δεδομένα που προέρχονται από το Logstash, και το Kibana χρησιμοποιεί τα δεδομένα που είναι αποθηκευμένα στο ElasticSearch την οπτικοποιήση αυτών.

8.1 Ορισμός και βασικές έννοιες για το ElasticSearch

Το ElasticSearch είναι μια μηχανή αναζήτησης πραγματικού χρόνου (real time search) και ανάλυσης δεδομένων (analytics engine). Βασίζεται στο apache lucene, μια υψηλών επιδόσεων με αρκετά χαρακτηριστικά βιβλιοθήκη για μηχανές αναζήτησης. Επιτρέπει την γρήγορη εξερεύνηση των δεδομένων. Χρησιμοποιείται για την αναζήτηση πλήρους κειμένου, δομημένη αναζήτησης, ανάλυσης δεδομένων και σε συνδυασμό και των τριών.

Κάποιες από τις εφαρμογές του σε μεγάλες εταιρείες είναι οι ακόλουθες:

- Η **Βικιπαίδεια** χρησιμοποιεί το ElasticSearch για να παρέχει αναζήτηση πλήρους κειμένου με επισημασμένα αποσπάσματα αναζήτησης.

- Η **Guardian** χρησιμοποιεί το Elasticscarch για να συνδυάσει αρχεία καταγραφής επισκεπτών με δεδομένα κοινωνικού δικτύου για να παρέχει σχόλια σε πραγματικό χρόνο στους συντάκτες του σχετικά με την ανταπόκριση του κοινού σε νέα άρθρα.

- Το **Stack Overflow** συνδυάζει το πλήρες κείμενο με ερωτήματα γεωγραφικής θέσης και χρησιμοποιεί “More Like This Query” (*βλ.* [25]) για να βρει σχετικές ερωτήσεις και απαντήσεις.

- Το **GitHub** χρησιμοποιεί το ElasticSearch για να κάνει αναζήτηση σε 130 δισεκατομμύρια γραμμές κώδικα.

Το ElasticSearch είναι γραμμένο σε Java χρησιμοποιώντας εσωτερικά το Lucene με σκοπό την ευρετηριοποίηση (indexing) και την αναζήτηση δεδομένων. Η έννοια του indexing είναι από τις σημαντικότερες σε ότι αφορά την λειτουργία του ElasticSearch. [26]

Ευρετήριο (Index):

Το ευρετήριο είναι το μέρος όπου το ElasticSearch αποθηκεύει τα δεδομένα του. Σε σύγκριση με μία σχεσιακή βάση δεδομένων, ένα ευρετήριο μπορεί να αντιστοιχηθεί μ' έναν πίνακα. Αλλά σε αντίθεση με μια σχεσιακή βάση δεδομένων, οι τιμές του πίνακα που είναι αποθηκευμένες σε ένα ευρετήριο προετοιμάζονται για γρήγορη και αποτελεσματική αναζήτηση πλήρους κειμένου. Παραδείγματος χάριν το ευρετήριο ElasticSearch μπορεί να θεωρηθεί σαν μια συλλογή στο MongoDB.

Έγγραφο (Document):

Η κύρια οντότητα που είναι αποθηκευμένη στο ElasticSearch είναι ένα έγγραφο. Σε μια αναλογία με σχεσιακές βάσεις δεδομένων, ένα έγγραφο είναι μια σειρά δεδομένων σε έναν πίνακα βάσεων δεδομένων. Συγκρίνοντας ένα έγγραφο του ElasticSearch με τη MongoDB, και τα δύο μπορεί να έχουν διαφορετικές δομές, αλλά στο ElasticSearch τα δεδομένα πρέπει να έχουν τον ίδιο τύπο σε κοινά πεδία.

Τα έγγραφα αποτελούνται από πεδία (στήλες σειράς), αλλά κάθε πεδίο μπορεί να εμφανιστεί αρκετές φορές και ένα τέτοιο πεδίο ονομάζεται πολλαπλών τιμών. Κάθε πεδίο έχει έναν τύπο (κείμενο, αριθμός, ημερομηνία και ούτω καθεξής). Οι τύποι πεδίων μπορεί επίσης να είναι περίπλοκοι - ένα πεδίο μπορεί να περιέχει άλλα δευτερεύοντα έγγραφα ή πίνακες. Ο τύπος πεδίου είναι σημαντικός για το ElasticSearch - δίνει πληροφορίες στη μηχανή αναζήτησης για το πώς πρέπει να εκτελούνται διάφορες λειτουργίες, όπως σύγκριση ή ταξινόμηση. Ευτυχώς, αυτό μπορεί να προσδιοριστεί αυτόματα. Σε αντίθεση με τις σχεσιακές βάσεις δεδομένων, τα έγγραφα δεν χρειάζεται να έχουν σταθερή δομή. Κάθε έγγραφο μπορεί να έχει διαφορετικό σύνολο πεδίων και επιπλέον, τα πεδία δεν χρειάζεται να είναι γνωστά κατά την ανάπτυξη εφαρμογών.

Τύπος εγγράφου (Document Type):

Στο ElasticSearch, ένα ευρετήριο μπορεί να αποθηκεύσει πολλά αντικείμενα για διαφορετικούς σκοπούς. Για παράδειγμα, μια εφαρμογή ιστολογίου μπορεί να αποθηκεύσει άρθρα και σχόλια. Ο τύπος εγγράφου στο ElasticSearch σ' ένα ευρετήριο μπορεί να αποθηκεύσει πολλά αντικείμενα με διαφορετικούς σκοπούς. Ο τύπος εγγράφου επιτρέπει εύκολα την διαφοροποίηση αυτών των κειμένων. Αξίζει να σημειωθεί ότι σχεδόν για κάθε έγγραφο μπορεί να έχει διαφορετική δομή, αλλά σε πραγματικές λειτουργίες, ο διαχωρισμός των τύπων βοηθά σημαντικά στη διαχείριση δεδομένων. Φυσικά, υπάρχουν και περιορισμοί. Ένας τέτοιος περιορισμός είναι ότι οι διαφορετικοί τύποι εγγράφων δεν μπορούν να ορίσουν διαφορετικούς τύπους για την ίδια ιδιότητα.

Κόμβος (node) και συστάδα (cluster):

Το ElasticSearch μπορεί να λειτουργήσει ως αυτόνομος διακομιστής. Ωστόσο, με σκοπό την επεξεργασία μεγάλων συνόλων δεδομένων και την επίτευξη ανοχής σφαλμάτων, το ElasticSearch μπορεί να εκτελεστεί σε πολλούς συνεργαζόμενους διακομιστές. Αυτοί οι διακομιστές ονομάζονται

συστάδα (cluster) και καθένας από αυτούς ονομάζεται κόμβος (node). Μεγάλες ποσότητες δεδομένων μπορούν να χωριστούν σε πολλούς κόμβους μέσω ενός shard ευρετηρίου (χωρίζοντάς τα σε μικρότερα μεμονωμένα μέρη). Καλύτερη διαθεσιμότητα και απόδοση επιτυγχάνεται μέσω των αντιγράφων (replicas).

Shard:

Όταν έχουμε μεγάλο αριθμό εγγράφων, μπορούμε να φτάσουμε σε ένα σημείο όπου ένας μόνο κόμβος δεν είναι αρκετός λόγω των περιορισμών της μνήμης RAM, της χωρητικότητας του σκληρού δίσκου και ούτω καθεξής. Το άλλο πρόβλημα είναι ότι η επιθυμητή λειτουργικότητα είναι τόσο περίπλοκη που η υπολογιστική ισχύς του διακομιστή δεν επαρκεί. Σε τέτοιες περιπτώσεις, τα δεδομένα μπορούν να χωριστούν σε μικρότερα μέρη που ονομάζονται shards, όπου κάθε shard είναι ένα ξεχωριστό Apache Lucene index. Κάθε shard μπορεί να τοποθετηθεί σε διαφορετικό διακομιστή και έτσι τα δεδομένα μπορούν να εξαπλωθούν μεταξύ των συστάδων. Όταν υποβληθεί ένα ερώτημα σε ένα ευρετήριο που έχει δημιουργηθεί από πολλά shards, το ElasticSearch στέλνει το ερώτημα σε κάθε σχετικό shard και συγχωνεύει το αποτέλεσμα με τρόπο ώστε η εφαρμογή να μην χρειάζεται να γνωρίζει για τα σχετικά shards.

Αντίγραφα (Replica):

Για να αυξηθεί η ταχύτητα διεκπεραίωσης του ερωτήματος ή να επιτευχθεί υψηλή διαθεσιμότητα, μπορούν να χρησιμοποιηθούν τα shards. Ένα αντίγραφο είναι ένα ακριβές αντίγραφο του πρωτεύοντος shard και κάθε shard μπορεί να έχει μηδενικά ή περισσότερα αντίγραφα. Όταν χαθεί το κύριο shard (για παράδειγμα, ο διακομιστής που κρατά τα δεδομένα του shard δεν είναι διαθέσιμος), μία συστάδα μπορεί να προωθήσει ένα αντίγραφο για να είναι το νέο κύριο shard.

Τρόπος επικοινωνίας με το ElasticSearch:

Ένας από τους τρόπους, που επιλέχθηκε στην υλοποίηση της εφαρμογής με σκοπό την δημιουργία επικοινωνίας με το ElasticSearch, είναι το «RESTful API με json μέσω http». Ένα αίτημα προς το ElasticSearch περιέχει τα ακόλουθα μέρη ενός HTTP αιτήματος:

```
curl -X<VERB> '<PROTOCOL>://<HOST>/<PATH>?<QUERY_STRING>' -D '<BODY>'
```

- verb: η κατάλληλη HTTP μέθοδος: GET, POST, PUT, HEAD ή DELETE.
- *protocol*: http/https
- *host:* του hostname του node από το ElasticSearch cluster ή *localhost* αν τρέχει τοπικά.
- *port:* η πόρτα που τρέχει το ES, από προεπιλογή τρέχει στην πόρτα 9200
- *query_string:* οποιαδήποτε προαιρετική παράμετρος (πχ *?pretty,* όπου θα τυπώσει ένα ευανάγνωστο *"pretty-print"* json response)
- *body:* ένα κωδικοποιημένο json αίτημα σώματος (αν χρειάζεται)

Π.χ για την εύρεση των αριθμό των clusters:

```
curl -XGET ‘http://localhost:9200/_count?pretty’ -d ‘
{   “query”: {
    “match_all”: {}
    }
}
```

Προσανατολισμένο κατά έγγραφο (Document Oriented):

Το ElasticSearch είναι προσανατολισμένο κατά έγγραφο το οποίο σημαίνει ότι αποθηκεύει ολόκληρα documents ή objects – τα οποία μπορεί να είναι μία λίστα με keys και values, είτε ακόμα και μία δομή δεδομένων που περιέχει ημερομηνίες, γεωγραφική τοποθεσία ή άλλα objects. Επιπρόσθετα, χρησιμοποιεί το JavaScript Object Notation, ή *JSON* για την μορφοποίηση των documents. Το JSON έχει γίνει το κύριο format για NoSQL.

Mapping:

Το ElasticSearch προκειμένου να είναι σε θέση να αντιμετωπίζει τα πεδία ημερομηνιών ως ημερομηνίες, τα αριθμητικά πεδία ως αριθμούς και τα πεδία συμβολοσειρών ως συμβολοσειρές πλήρους κειμένου ή ακριβούς τιμής, πρέπει να γνωρίζει τι τύπου δεδομένων κάθε πεδίο περιέχει. Αυτές οι πληροφορίες διαμορφώνονται κατά τη διαδικασία του mapping. Όπως έχει περιγραφεί παραπάνω, κάθε έγγραφο σε ένα ευρετήριο έχει έναν τύπο. Γίνεται επομένως αντιστοίχιση των πεδίων αναλόγως με τον τύπο δεδομένων τους. Το ElasticSearch καθορίζει τον τύπο δεδομένων για κάθε πεδίο και τον τρόπο χειρισμού του πεδίου. Το mapping χρησιμοποιείται επίσης για τη διαμόρφωση μεταδομένων που σχετίζονται με τον τύπο δεδομένων.

Το ElasticSearch υποστηρίζει τους ακόλουθους τύπους δεδομένων:

- Συμβολοσειρά (String): *string*
- Αριθμός: byte, short, integer, long
- Δεκαδικός αριθμός: *float, double*
- Boolean: *boolean*
- Ημερομηνία: *date*

Template:

Ένα template ορίζει ρυθμίσεις και αντιστοιχίσεις που μπορούν να εφαρμοστούν αυτόματα κατά τη δημιουργία νέων ευρετηρίων. Το ElasticSearch εφαρμόζει templates σε νέα ευρετήρια με βάση ένα μοτίβο ευρετηρίου που ταιριάζει με το όνομα ευρετηρίου.

Τα templates εφαρμόζονται μόνο κατά τη δημιουργία του ευρετηρίου. Οι αλλαγές στα template ενός ευρετηρίου δεν επηρεάζουν τα υπάρχοντα ευρετήρια. Οι ρυθμίσεις και οι αντιστοιχίσεις που καθορίζονται στο αίτημα δημιουργίας ευρετηρίου API παρακάμπτουν τυχόν ρυθμίσεις ή αντιστοιχίσεις που καθορίζονται σε ένα πρότυπο ευρετηρίου. [27]

8.2 Ορισμός και αρχιτεκτονική του Logstash

Το Logstash είναι ένα ολοκληρωμένο εργαλείο για τη συλλογή, τη συγκέντρωση, την ανάλυση, την αποθήκευση και την αναζήτηση αρχείων καταγραφής δεδομένων. Είναι ένα δωρεάν και ανοιχτού κώδικα λογισμικό (με άδεια του Apache 2.0). Αναπτύχθηκε αρχικά από τον Αμερικανό προγραμματιστή, Jordan Sissel και τώρα διατηρείται από την ομάδα του Elastic. Είναι εύκολο να ρυθμιστεί, να εκτελεστεί και να επεκταθεί. [28]

Η αρχιτεκτονική του Logstash αποτελείται από τρία μέρη:

Είσοδος Καταγραφής (Log Input)

1. Το Logstash διαθέτει μια μεγάλη ποικιλία μηχανισμών εισόδου: μπορεί να λάβει είσοδο από TCP / UDP, αρχεία, Syslog, Microsoft Windows EventLogs, STDN και μια ποικιλία άλλων πηγών. Το ElasticSearch αποστέλλει στο Logstash μέσω μιας συλλογή ανοιχτού κώδικα εισόδου που ονομάζεται Beats, που μπορεί να συμβάλλει στην καταγραφή δεδομένων από διαφορετικές πηγές.

Καταγραφή Φιλτραρίσματος (Log Filtering)

2. Όταν τα συμβάντα φτάσουν στον Logstash server εκεί υπάρχει η δυνατότητα τροποποίησης και μετασχηματισμού αυτών μέσω μίας μεγάλης συλλογής φίλτρων. Τα φίλτρα εφαρμόζονται συχνά υπό όρους ανάλογα με τα χαρακτηριστικά του συμβάντος.

Τέσσερα από τα πιο σημαντικά και δημοφιλή φίλτρα του Logstash είναι τα ακόλουθα:

i. *Grok:* επιτρέπει τη δημιουργία δομής σε μη δομημένα αρχεία καταγραφής.

ii. *Mutate:* επιτρέπει την διαμόρφωση/τροποποίηση των πεδίων. Για παράδειγμα, χρησιμοποιείται για την ένωση ή μετονομασία των πεδίων, κ.α.

iii. *Date:* μπορεί να χρησιμοποιηθεί για την ανάκτηση μίας ώρας και μίας ημερομηνίας από ένα μήνυμα καταγραφής και να οριστεί ως πεδίο χρονικής σήμανσης (@timestamp) για το αρχείο καταγραφής. Μόλις καθοριστεί, αυτό το πεδίο χρονικής σήμανσης θα ταξινομήσει τα αρχεία καταγραφής με τη σωστή χρονολογική σειρά με σκοπό την αποτελεσματική ανάλυση των γεγονότων.

iv. *Json:* Επιτρέπει στους χρήστες να γράφουν δομημένα μηνύματα που μπορούν να διαβαστούν και να αναλυθούν εύκολα. Το Logstash json filter plugin επιτρέπει τη δημιουργία δομής δεδομένων json. [29]

3.

Έξοδος Καταγραφής (Log Output)

Τέλος, κατά την εξαγωγή δεδομένων, το Logstash υποστηρίζει ένα τεράστιο εύρος προορισμών, όπως TCP / UDP, e-mail, αρχεία, HTTP, Nagios και μια μεγάλη ποικιλία δικτύων και διαδικτυακών υπηρεσιών. Υπάρχει η δυνατότητα ενσωμάτωσης του Logstash με μηχανές μετρήσεων, εργαλεία ειδοποίησης, σουίτες γραφημάτων, ή άλλους προορισμούς αποθήκευσης. Τελικά όμως, η συντριπτική πλειονότητα των δεδομένων προορίζεται για το ElasticSearch, όπου γίνεται η αποθήκευση, η υποβολή ερωτημάτων και η διαχείριση αυτών. [28]

8.3 Ορισμός και βασικά χαρακτηριστικά του Kibana

Το Kibana είναι ένα εργαλείο που αποτελεί μέρος του ELK stack. Κατασκευάστηκε και αναπτύχθηκε από την Elastic. Το Kibana είναι μια πλατφόρμα οπτικοποίησης που είναι χτισμένη πάνω από το ElasticSearch και αξιοποιεί τις λειτουργίες αυτού.

Το Kibana λειτουργεί ως ένα ανώτερο επίπεδο του ElasticSearch, παρέχοντας εξαιρετικές απεικονίσεις των δεδομένων (δομημένων ή μη δομημένων). Είναι ένα προϊόν ανάλυσης ανοιχτού κώδικα που χρησιμοποιείται για αναζήτηση, προβολή και ανάλυση δεδομένων. Παρέχει διάφορους τύπους οπτικοποιήσεων, για την απεικόνιση δεδομένων με τη μορφή πινάκων, διαγραμμάτων, χαρτών, ιστογραμμάτων και ούτω καθεξής. Παρέχει επίσης μια διαδικτυακή διεπαφή που μπορεί εύκολα να χειριστεί μια μεγάλη ποσότητα δεδομένων.

Είναι ένα προϊόν με άδεια Apache που στοχεύει στην παροχή μιας ευέλικτης διεπαφής σε συνδυασμό με τις ισχυρές δυνατότητες αναζήτησης της ElasticSearch. Απαιτεί έναν διακομιστή ιστού και οποιοδήποτε σύγχρονο πρόγραμμα περιήγησης ιστού. Συνδέεται με το ElasticSearch χρησιμοποιώντας το REST API. [30]

Dashboard:

Το Kibana επιτρέπει τη δημιουργία ταμπλό (dashboards) που συμβάλλουν στην αναζήτηση δεδομένων σε πραγματικό χρόνο. Τα **dashboards** δεν είναι παρά μια διεπαφή για τα υποκείμενα JSON έγγραφα. Χρησιμοποιούνται για την αναπαράσταση, αποθήκευση και εξαγωγή δεδομένων. Είναι εύκολο να ρυθμιστούν και να χρησιμοποιηθούν, κάτι που κάνει εύκολη την αναζήτηση στο ElasticSearch σε λίγα δευτερόλεπτα.

Discover page:

Το **Discover** είναι μία από τις σελίδες που υπάρχουν στο Kibana που συμβάλλει κυρίως στην απεικόνιση των δεδομένων. Η σελίδα Discover παίζει σημαντικό ρόλο στην κατανόηση των δεδομένων για διαφορετικά είδη οπτικοποίησης. Αυτή η σελίδα παρέχει μια πλήρη επισκόπηση των δεδομένων, συμπεριλαμβανομένων λιστών ευρετηρίων, λιστών πεδίων και εμφάνισης κειμένου που περιέχεται σε πεδία. Σε αυτήν τη σελίδα, υπάρχει η δυνατότητα εύρεσης δεδομένων που είναι αποθηκευμένα σε διαφορετικά ευρετήρια αλλάζοντας το μοτίβο ευρετηρίου.

Το discover περιέχει τα ακόλουθα στοιχεία που απεικονίζονται και στην εικόνα 8.2

- Time filter: περιέχει δεδομένα ενός συγκεκριμένου χρονικού διαστήματος
- Toolbar: περιέχει τη μπάρα αναζήτησης με την επιλογή νέας αναζήτησης, αποθήκευση ή φόρτωση αποθηκευμένης αναζήτησης και άλλων ρυθμίσεων
- Index name: το όνομα του επιλεγμένου ευρετηρίου
- Fields list: περιέχει τα πεδία του επιλεγμένου ευρετηρίου
- Hits: απεικονίζει το σύνολο των εγγράφων που βρέθηκαν σε επιλεγμένο χρονικό πλαίσιο.
- Document data: περιέχει όλα τα έγγραφα μαζί με τα δεδομένα σε όλα επιλεγμένα πεδία

- Histogram: Απεικονίζει τα έγγραφα που βρέθηκαν σε επιλεγμένο χρονικό πλαίσιο.

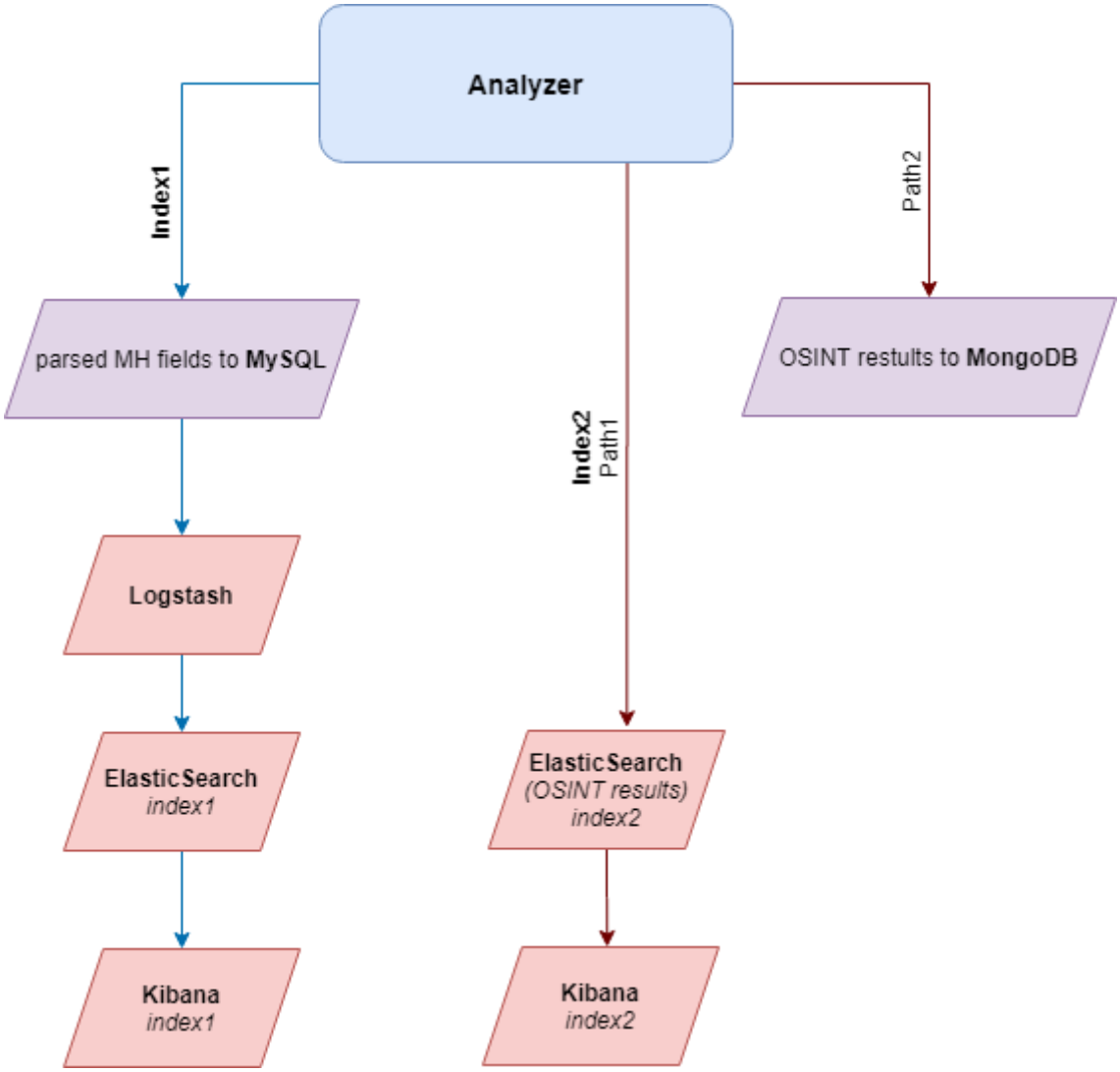
Εικόνα 8.2 Kibana - Discover

Μέσω του toolbar υπάρχει η δυνατότητα υποβολής ερωτημάτων με την γλώσσα KQL. Η **Kibana Query Language (KQL)** διευκολύνει την εύρεση των πεδίων στο ElasticSearch. Κάθε αποτέλεσμα ερωτήματος αναζήτησης εμφανίζει τα αντίστοιχα έγγραφα. Το ιστόγραμμα εμφανίζεται σε αυτήν τη σελίδα, το οποίο βοηθά στην προβολή των δεδομένων σε χρονική σειρά. Η χρονική σήμανση (timestampp) πρέπει να έχει καθοριστεί σε κάθε πεδίο ενός εγγράφου με τύπο “date” για κάθε ευρετήριο.

8.4 Υλοποίηση του ELK στο sace

Εφόσον έχει προηγηθεί η εγκατάσταση του ELK σύμφωνα με το Παράρτημα ΙΙ, σ’ αυτήν την ενότητα θα περιγραφεί η χρήση του στα πλαίσια της υλοποίησης του στην εφαρμογή “sace”. Σκοπός είναι η αποθήκευση, επεξεργασία και η αναπαράσταση των δεδομένων που αφορούν την πληροφορία που βρίσκεται σ’ ένα ή περισσότερα mail headers.

Με σκοπό την κατανόηση της διαδρομής των δεδομένων ακολουθεί ένα συνολικό διάγραμμα της πορείας των δεδομένων από την στιγμή που θα εισέρθουν στον analyzer μέχρι να αποθηκευτούν στο ES.



Εικόνα 8.3 Αποθήκευση των δεδομένων μέσω διαφορετικών διαδρομών στον analyzer

Το διάγραμμα θα αναλυθεί με βάση τα ευρετήρια, δηλαδή το index1 και το index2.

8.4.1 Πρώτο ευρετήριο: Αποστολή πεδίων της MySQL στο ES

- Πρώτο στάδιο που πρέπει να εκπληρωθεί μέσω του analyzer είναι η αποθήκευση των δεδομένων σε στη **MySQL βάση δεδομένων** όπως έχει περιγραφεί στην ενότητα 5.2.1.
- Δεύτερο στάδιο είναι τα δεδομένα των πινάκων από την MySQL να σταλούν στο **Logstash**. Άρα δημιουργείται η ανάγκη το Logstash να μπορεί να επικοινωνήσει με την βάση δεδομένων, σ' αυτό συμβάλλει το **jdbc plugin**.

Το plugin αυτό δημιουργήθηκε με στόχο την ανάγνωση δεδομένων από SQL βάση δεδομένων με διασύνδεση JDBC (Java Database Connectivity) στο Logstash.

Η επιθυμητή βιβλιοθήκη προγραμμάτων οδήγησης jdbc πρέπει να μεταφερθεί στον κατάλληλο κατάλογο έτσι ώστε το Logstash να μπορεί να το διαβάσει. Το Logstash χρησιμοποιεί την επιλογή διαμόρφωσης jdbc_driver_library για αυτόν τον σκοπό.

Υπάρχει η δυνατότητα επίσης προγραμματισμού των ερωτημάτων σε περιοδικό χρόνο, χρησιμοποιώντας σύνταξη cron ή εκτέλεση μία φοράς ενός ερωτήματος ως προς το Logstash [31]. Παραδείγματος χάριν

`* 5 * 1-3 *`	Θα εκτελεστεί κάθε λεπτό από τις 5am κάθε μέρα από τον Ιανουάριο μέχρι τον Μάρτιο.

Το Logstash χρησιμοποιεί την **παράμετρο schedule** για αυτόν τον σκοπό.

Η λήψη του σχετικού jar αρχείο έγινε από τον ακόλουθο σύνδεσμο: https://downloads.mysql.com/archives/c-j/. Στην συνέχεια το επιθυμητό jar αρχείο τοποθετήθηκε στον εξής κατάλογο:

```
/usr/share/logstash/logstash-core/lib/jars/
```

Σ' αυτό το στάδιο μπορεί να γίνει η διαμόρφωση των **input, filter και output** αρχείων του Logstash.

- Στο input αρχείο με όνομα: "002-spamMH.conf" θα δοθούν όλοι οι είσοδοι από τη MySQL μέσω του jdbc με την ακόλουθη δομή *input {jdbc {} }.*

Παράδειγμα εισόδου:

```
1.  input {
2.    jdbc {
3.      jdbc_driver_library => "/usr/share/Logstash/Logstash-core/lib/jars/mysql-
    connector-java-8.0.19.jar"
4.      jdbc_driver_class => "com.mysql.cj.jdbc.Driver"
5.      jdbc_connection_string => "jdbc:mysql://localhost:3306/spam_mail_db"
6.      jdbc_user => "sace"
7.      jdbc_password => "pass"
8.      schedule => "*/1 * * * *"
9.  statement => "SELECT * FROM indexHeader LEFT JOIN Sender ON indexHeader.id_index=Se
    nder.sender_iheader
10.                       LEFT JOIN Receiver ON Receiver.rec_send=Sender.id_sender
11.                       LEFT JOIN Mail ON mail_send=Sender.id_sender
12.                       where
13.                       mh_timestamp > :sql_last_value;"
14.   type => mheader
15.     }
16. }
```

Στο **jdbc_driver_library** ορίζεται η jar βιβλιοθήκη όπως αναφέρθηκε παραπάνω και αντίστοιχα στην παράμετρο **jdbc_driver_class** επιλέγεται η κλάση του driver, η MySQL δηλαδή. Μέσω του **jdbc_connection_string** πραγματοποιείται η σύνδεση με την βάση 'spam_mail_db' και με τα **jdbc_user και jdbc_password** αυθεντικοποιείται ο χρήστης της βάσης. Το **schedule** έχει οριστεί στο

1 λεπτό, όπου αυτό σημαίνει ότι το **statement** που περιέχει το επιθυμητό query θα εκτελείται κάθε 1 λεπτό. Στο **type** δίνεται ένα σχετικό όνομα αναφορικά με την είσοδο των δεδομένων με στόχο την διευκόλυνση της παραμετροποίησης των configuration αρχείων του Logstash (".conf") σε επόμενο βήμα.

- Ακολουθεί η διαδικασία του **φιλτραρίσματος** είτε μέσω δημιουργίας νέου αρχείου είτε μέσα στο input αρχείο με την ακόλουθη δομή *filter {}*.

Παράδειγμα φιλτραρίσματος:

```
filter {
 if [geo_ip_latitude] and [geo_ip_longitude] {
  mutate {
           convert => {"[location][geo_ip_latitude]" => "float"}
           convert => {"[location][geo_ip_longitude]" => "float"}
           }
  mutate {
    rename => {
      "geo_ip_longitude" => "[location][lon]"
      "geo_ip_latitude" => "[location][lat]"
      }
    }
```

Το συγκεκριμένο παράδειγμα στοχεύει στην αλλαγή του τύπου των δεδομένων που προέρχονται από τα πεδία "geo_ip_latitude" και "geo_ip_longitude" και την αλλαγή των ονομάτων τους. Στο template του index1 έχει οριστεί το πεδίο "location" με type "geo_point". Όμως το "geo-point" εκφράζεται ως αντικείμενο, με κλειδιά το lat και το lon. Έτσι αφού γίνει αλλαγή των τύπων των "geo_ip_latitude" και "geo_ip_longitude" μέσω του mutate, μετά γίνεται η μετονομασία τους σε "lat" και "lon" αντίστοιχα μέσω ίδιου φίλτρου και πάλι. Μ' αυτόν τον τρόπο θα γίνει το σωστό mapping ώστε τελικά να γίνει η απεικόνιση των γεωγραφικών σημείων στον χάρτη.

- Τέλος, δημιουργείται το "30-ESoutput.conf" για το **output** αρχείο, ώστε τα δεδομένα να σταλούν στο ElasticSearch. Η αντίστοιχη δομή είναι *output {}*.

Παράδειγμα εξόδου:

```
output {
  if [type] == "mheader" {
          ElasticSearch {
                hosts => ["localhost:9200"]
                index => "spamdb_index1"
                document_id => "%{id_index}"
                codec=> "json"
            }
          }
….
}
```

Πρώτα ελέγχεται το type των δεδομένων όπως τους έχει δοθεί από το input αρχείο. Εάν ισχύει το παραπάνω, μέσα στην δομή *ElasticSearch {}* γίνεται η ρύθμιση επόμενων παραμέτρων. Όπως, το hosts όπου ορίζεται ο host που φιλοξενεί το ElasticSearch, έπειτα το όνομα του ευρετηρίου όπου θα αποθηκευτούν τα δεδομένα. Σημαντικό είναι επίσης με την παράμετρο "document_id" να καθοριστεί ένα μοναδικό id του κάθε εγγράφου όπου θα διαχωρίζει τα έγγραφα μεταξύ τους. Τέλος με το codec γίνεται η αναπαράσταση των δεδομένων σε json μορφή.

Επόμενο βήμα είναι η δημιουργία templates με στόχο το σωστό **mapping** των δεδομένων από τη MySQL στο ElasticSearch.

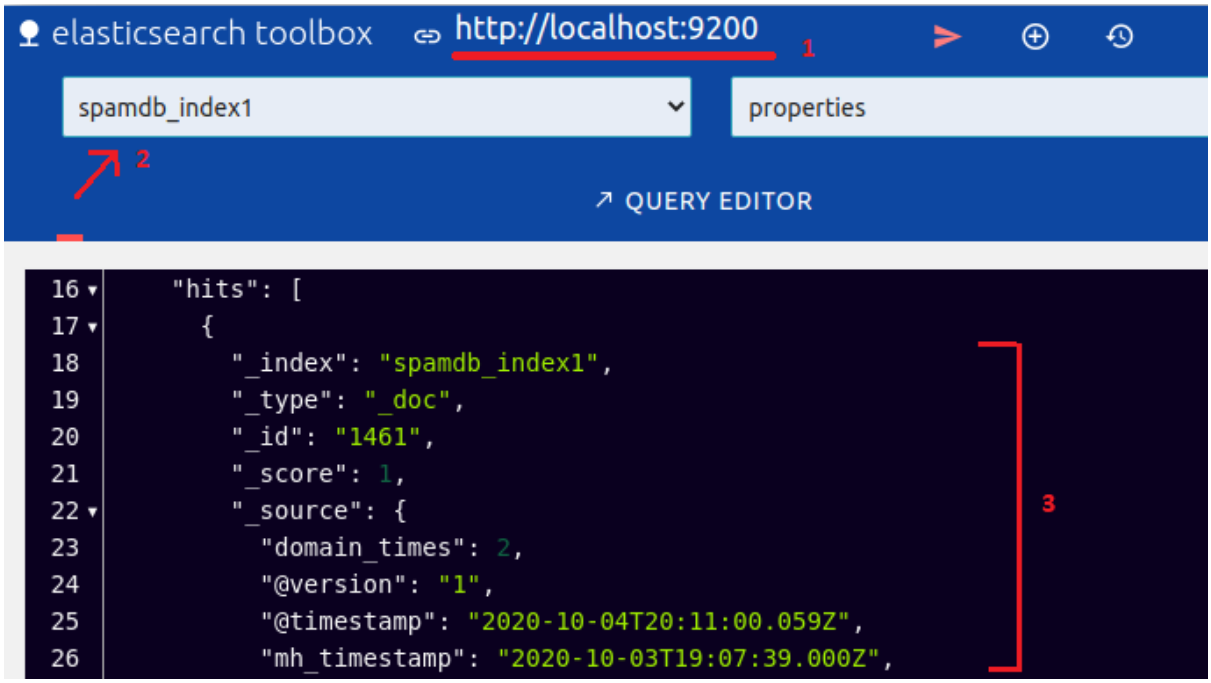
Παράδειγμα template:

```
curl -X PUT "localhost:9200/_template/template_spamdb_index1?pretty" -H 'Content-Type: application/json' -d'
{
    "order": 1000,
    "index_patterns" : ["spamdb_index1"],
    "mappings": {
       "dynamic": "true",
        "properties": {

            "id_index": {"type" : "keyword","index" : false},
            "input_text": {"type": "text"},
            "sace_user": {"type": "keyword"},
            "mh_timestamp": {"type":   "date"},
            "id_geo_ip2": {"type" : "keyword","index" : false},
            "geo_ip_address": {"type": "ip"},
            "location": {"type": "geo_point"},
            "geo_ip_country": {"type": "keyword"},
            "geo_ip_state": {"type": "keyword"},
            "geo_ip_city": {"type": "keyword"},
            "geo_ip_latitude": {"type": "long"},
            "geo_ip_longitude": {"type": "long"},
            "geo_ip_num": {"type": "integer"},
….
}
```

Αξιοποιώντας το REST API δημιουργείται ένα νέο template με όνομα “template_spamdb_index1”. Σημαντικό είναι να οριστεί το όνομα του ευρετηρίου όπου το συγκεκριμένο template θα εφαρμοστεί γι’ αυτό, "index_patterns" : ["spamdb_index1"]. Έπειτα ορίζονται μέσα στο properties τα πεδία της βάσης με τον αντίστοιχο επιθυμητό τύπο. Αξιοσημείωτο είναι το πεδίο "mh_timestamp" όπου αφορά την ακριβή ημερομηνία και ώρα όπου το e-mail header εισήλθε στην βάση, και πρέπει να έχει τύπο “date” για την σωστή χρονολόγηση των γεγονότων.

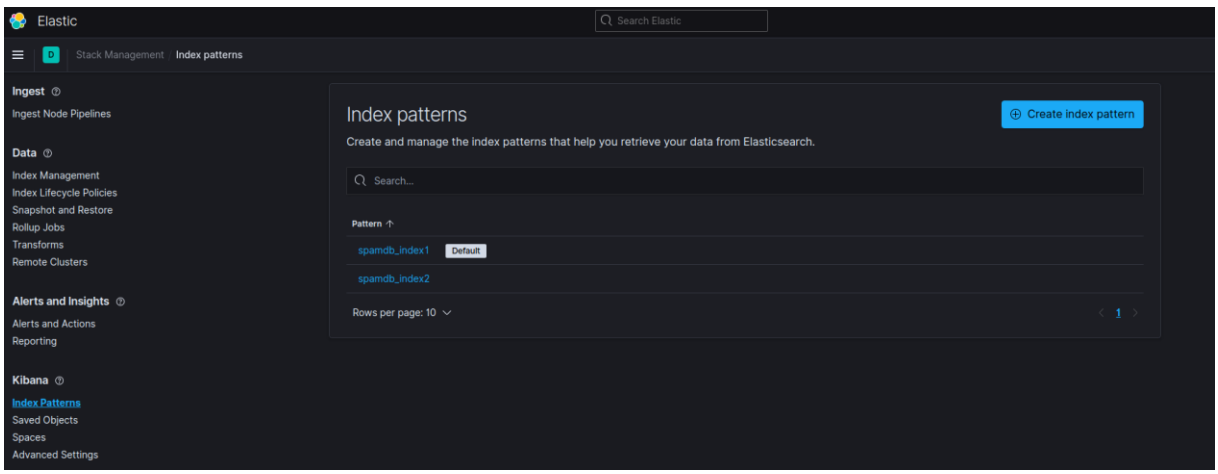
Εφόσον έχουν υλοποιηθεί τα προαναφερθέντα και μετά την εισαγωγή των πρώτων δεδομένων στο ElasticSearch, με το εργαλείο “ElasticSearchToolbox” μπορεί να γίνει επιβεβαίωση ότι τα δεδομένα έχουν αποθηκευτεί σ’ αυτό. Το “ElasticSearchToolbox” είναι ένα εργαλείο που κύριο στόχο έχει την γρήγορη αναζήτηση πεδίων στο ElasticSearch.



Εικόνα 8.4 ElasticSearch Toolbox

Στην εικόνα 8.4 παρατηρείται η επιτυχημένη σύνδεση στο ES *(αρ.1)* μέσω του ElasticSearchToolbox. Έχοντας επιλέξει το επιθυμητό index *(αρ.2)* παρατηρούνται επίσης τα σχετικά hits των εγγράφων *(αρ.3)*.

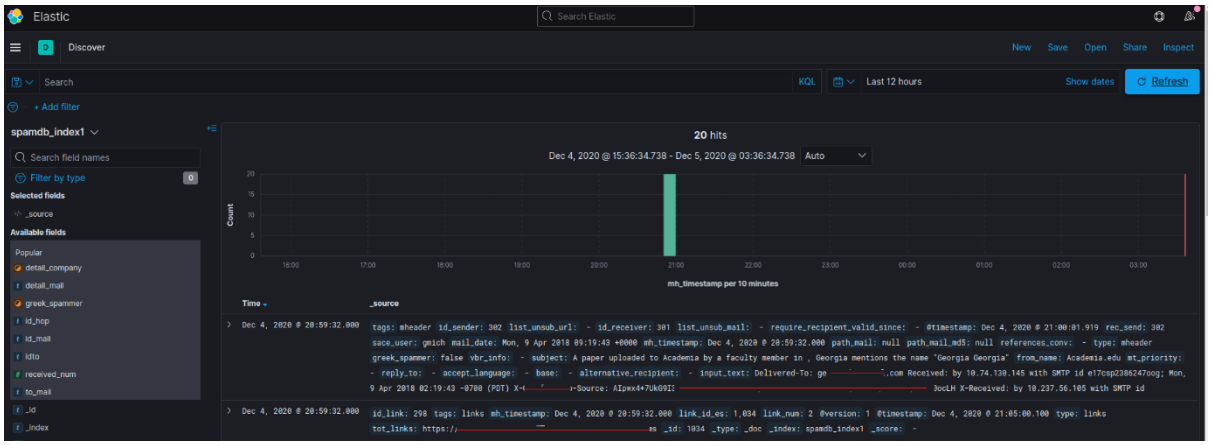
Με την σωστή αποθήκευση των δεδομένων στο ElasticSearch ακολουθεί η δημιουργία του ευρετηρίου στο Kibana. Στην αρχική σελίδα του Kibana, γίνεται η επιλογή του *"Stack Management"*.



Εικόνα 8.5 Δημιουργία ευρετηρίων

Έπειτα από εκεί πατώντας στο 'Create a new index' δηλώνεται στη συνέχεια το index του ES και τέλος σημαντικό είναι να δοθεί το timestamp των γεγονότων.

Τα hits με το σωστό timestampp για το index1 παρατηρούνται στην ακόλουθη εικόνα.



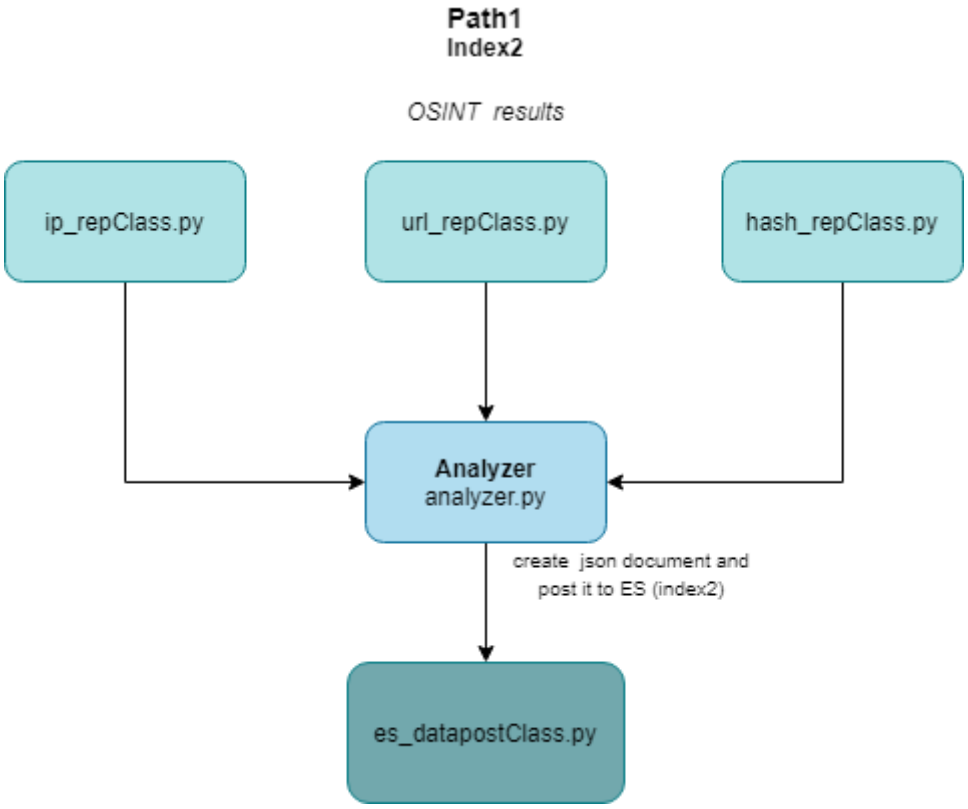
Εικόνα 8.6 Kibana – γεγονότα στο discover

8.4.2 Δεύτερο ευρετήριο: Αποστολή OSINT αποτελεσμάτων στο ES και αποθήκευση σε MongoDB

Αναφορικά με το δεύτερο ευρετήριο υπάρχουν δύο διαδρομές με σκοπό την αποθήκευση των δεδομένων.

Path 1:

Στην διαδρομή αυτήν εφόσον έχει προηγηθεί η επιλογή των δεδομένων, τα οποία θα αποσταλούν μέσω API σε OSINT πηγές, στη συνέχεια ακολουθεί η αποθήκευση αυτών στο ElasticSearch. Στην παρακάτω εικόνα απεικονίζεται ένα διάγραμμα που αφορά την αποστολή των αποτελεσμάτων από κάθε OSINT πηγή σε json μορφή τους το ElasticSearch.



Εικόνα 8.7 Δημιουργία json document και αποστολή στο ES

Η αποθήκευση τους σύμφωνα και με την εικόνα 8.7 γίνεται μέσω του αρχείου “es_datapostClass.py”

```
1. clientES = Elasticsearch(«localhost:9200»)
2. # build the Elasticsearch document from a dict
3. build_doc = {}
4. build_doc[“_index”] = “spamdb_index2”
5. build_doc[“doc_type”] = “_doc”
6. build_doc[«_source»] = doc_data
```

Στην πρώτη γραμμή γίνεται η σύνδεση με το ElasticSearch και τους επόμενες γραμμές δίνονται τα ορίσματα που αφορούν πληροφορίες για το index το οποίο δημιουργείται.

```
1. json_source = json.dumps(build_doc[“_source”])
```

Η παραπάνω γραμμή αφορά τη δημιουργία του JSON string του doc_source που εμπεριέχει το σύνολο των δεδομένων τους αποθήκευση.

```
1. response = clientES.index(
2. index = build_doc[«_index»],
3. doc_type = ‘_doc’,
4. body = json_source)
```

Τέλος υλοποιείται ένα API call στο ElasticSearch cluster ώστε τα δεδομένα να αποσταλούν επιτυχώς στο ElasticSearch.

Ακολουθεί η δημιουργία του ευρετηρίου στο Kibana. Η διαδικασία είναι αντίστοιχη με αυτήν του index1, τους περιεγράφηκε παραπάνω.

Path2:

Στην δεύτερη διαδρομή πραγματοποιείται η αποθήκευση των αποτελεσμάτων σύμφωνα με τις OSINT πηγές σε **MongoDB βάση δεδομένων**.

Αντίστοιχα στην MongoDB αποθηκεύεται η πληροφορία από τις OSINT πηγές. Η σύνδεση με την MongoDB γίνεται με τον ακόλουθο κώδικα:

```
1.  client = MongoClient("localhost:27017";)
2.  db=client.admin
3.  db = client.spam_mail_db;
```

Για παράδειγμα η εισαγωγή του reputation σύμφωνα με τη SANS για μία IP έγινε σύμφωνα με το παρακάτω:

```
1.  mongoSANS = db.SANSIP.insert_one(data_SANS["ip"])
```

9. Δημιουργία ουράς και προγραμματισμός των αιτημάτων

Καθώς τα αιτήματα τους την εφαρμογή μπορεί να είναι πολλαπλά, μπορεί να καταστήσουν αυτήν είτε μη διαθέσιμη, είτε μπορεί να επηρεάσουν την αποτελεσματικότητα τους. Το αποτέλεσμα είναι να μην μπορεί να ανταποκριθεί, ακυρώνοντας νέα mail headers τους επεξεργασία. Έτσι δημιουργείται η ανάγκη αξιοποίησης τους ουράς για την αναμονή των αιτημάτων κατά την αποστολή των mail header προς τον collector καθώς και ο προγραμματισμός αυτών με σκοπό την επεξεργασία τους.

9.1 Ορισμός του Celery

Το Celery είναι ένα ευέλικτο και αξιόπιστο κατανεμημένο σύστημα με σκοπό την επεξεργασία μεγάλης ποσότητας μηνυμάτων. Ουσιαστικά συμβάλλει στην δημιουργία ουράς των επικείμενων εργασιών και η επεξεργασία αυτών γίνεται σε πραγματικό χρόνο, ενώ παράλληλα υποστηρίζει την δυνατότητα του προγραμματισμού των εργασιών αυτών.

Με τον όρο ουρά εργασιών (task queue) θεωρείται η υπηρεσία στην οποία γίνεται ανάθεση εργασιών που προκύπτουν από τον ανάλογο προγραμματισμό τους.

Το Celery θεωρείται μια ασύγχρονη ουρά εργασιών (asynchronous task queue) και αποτελείται από έναν client, έναν broker και πολλούς workers.

Οι **workers** είναι υπεύθυνοι για την εκτέλεση των εργασιών που τοποθετούνται στην ουρά. Παράλληλα με το celery, υπάρχει η δυνατότητα των τοπικών και των απομακρυσμένων worker, το οποίο σημαίνει ότι μία εργασία μπορεί να ανατεθεί σε διαφορετικά μηχανήματα, συνήθως μεγαλύτερων δυνατοτήτων, και τα αποτελέσματα να σταλούν πίσω στον client. Με αυτόν τον τρόπο, μετριάζεται ο φόρτος εργασιών στο κύριο μηχάνημα που τρέχει μία εφαρμογή, αφήνοντας περισσότερους διαθέσιμους πόρους σ' αυτήν.

Ο client είναι υπεύθυνος για την ανάθεση εργασιών στους workers και παράλληλα για την επικοινωνία αυτών μέσω ενός μεσίτη μηνυμάτων (message broker). Ο **broker** διευκολύνει την επικοινωνία μεταξύ του client και του worker μέσω μιας ουράς μηνυμάτων, όπου ένα μήνυμα προστίθεται στην ουρά και ο broker το παραδίδει στον client. Κάποιοι από τους πιο γνωστούς brokers είναι Redis και το RabbitMQ.

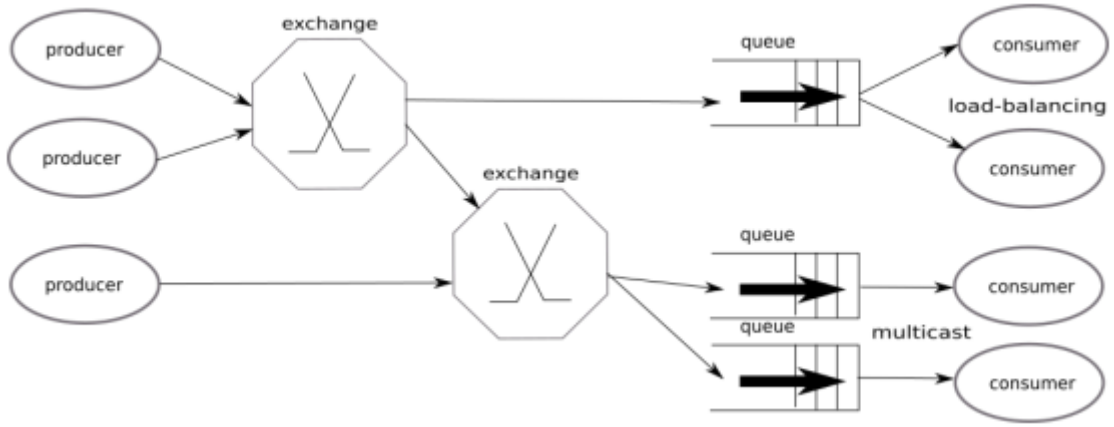
9.1.1 Επιλογή του RabbitMQ ως broker

Το RabbitMQ χρησιμοποιείται ως αποτελεσματική και επεκτάσιμη εφαρμογή του Προηγμένου Πρωτοκόλλου μηνυμάτων (Advanced Message Queuing Protocol ή **AMQP**). Ως εκ τούτου, παρακάτω θα δοθεί μια σύντομη εισαγωγή του AMQP και, στη συνέχεια του RabbitMQ.

Το AMQP δημιουργεί μία πλήρη δια λειτουργικότητα μεταξύ των συμβατών του client και του server μέσω των brokers. Το μοντέλο αυτό διαιρεί μία εργασία ενός μηνύματος του "μεσίτη" (message brokering task) μεταξύ των "ανταλλαγών" (exchanges**)** και της ουράς μηνυμάτων (message queue) [32]:

- Με τον όρο **exchanges** θεωρείται ουσιαστικά ένας δρομολογητής που δέχεται εισερχόμενα μηνύματα από εφαρμογές και, βάσει ενός συνόλου κανόνων ή κριτηρίων, αποφασίζει σε ποιες ουρές θα δρομολογούν τα μηνύματα.

- Η ουρά μηνυμάτων - **message queue,** αποθηκεύει μηνύματα και τα στέλνει στους καταναλωτές μηνυμάτων. Η ανθεκτικότητα του μέσου αποθήκευσης εξαρτάται εξ' ολοκλήρου από την εφαρμογή ουράς μηνυμάτων. Οι ουρές μηνυμάτων συνήθως αποθηκεύουν μηνύματα στο δίσκο έως ότου μπορούν να παραδοθούν, αλλά είναι επίσης δυνατόν οι ουρές να αποθηκεύουν μηνύματα και μόνο στη μνήμη.



Εικόνα 9.1 Αρχιτεκτονική του RabbitMQ

Όπως ειπώθηκε το RabbitMQ αποτελεί μία επεκτάσιμη εφαρμογή του AMQP, το υποστηρίζεται δηλαδή είτε μέσω κάποιας επέκτασης (plugin), είτε βρίσκεται σ' αυτό από προεπιλογή. Το RabbitMQ υπερβαίνει τις ικανότητες του AMQP από πολλές πτυχές. Ορισμένες επεκτάσεις εισάγουν νέες μεθόδους πρωτοκόλλου ενώ άλλες βασίζονται σε υπάρχουσες (βλ _[33]). Ενδεικτικά στην εικόνα 9.1 παρουσιάζεται η αρχιτεκτονική του RabbitMQ περιλαμβάνοντας τους exchanges και την ουρά των μηνυμάτων όπως περιεγράφηκε παραπάνω.

9.2 Χρήση του Celery στο sace

Ως προς την εγκατάσταση της βιβλιοθήκης του celery η εντολή που εκτελέστηκε στο μηχάνημα είναι :

```
pip install celery
```

και για το RabbitMQ, η εντολή είναι η ακόλουθη:

```
apt-get install rabbitmq-server
```

Ως προς την υλοποίηση της εφαρμογής, δημιουργήθηκε ένα αρχείο "task_collector.py" στο οποίο εφόσον γίνει η εισαγωγή των βιβλιοθηκών, στη συνέχεια ορίζεται το Celery. Παράλληλα, εφόσον ο RabbitMQ server τρέχει, επόμενο βήμα είναι να το ενσωματώσουμε στην εφαρμογή ως εξής:

```
1. from celery import Celery
2. app = Celery(__name__, backend='amqp', broker='amqp://guest@localhost//')
```

Η δημιουργία της εκάστοτε εργασίας γίνεται εύκολα με την κλήση της μεθόδου task() αξιοποιώντας το όρισμα app όπου σ' αυτό έχει υλοποιηθεί το Celery με broker το RabbitMQ.

Στη συνέχεια, η συνάρτηση canal() θα εκτελείται κάθε φορά που έρχεται ένα νέο αίτημα. Ουσιαστικά μ' αυτήν την υλοποίηση κάθε νέο αίτημα θα εισέρχεται στην ουρά.

```
1.  @app.task(bind=True)
2.  def canal(self,data):
3.      #Mail Header checks
```

Όταν το πρώτο request στην ουρά περνάει από όλους τους ελέγχους επιτυχώς, στέλνεται στον analyzer με σκοπό την περαιτέρω ανάλυση του.

Στο αρχείο "collector.py" γίνεται η παραλαβή του ΜΗ το οποίο στέλνεται απευθείας στο task_colllector.py.

```
1. canal.delay(data)
```

Η μέθοδος *delay()* χρησιμοποιείται για την αποστολή ενός task message, με σκοπό την εισαγωγή του message στην ουρά.

Για όποια νέα αιτήματα, θα εκτελεστούν μετά τον τερματισμό του τρέχοντος ΜΗ. Έως τότε παραμένουν στην ουρά.

9.3 Παρακολούθηση των tasks μέσω του Flower

Το Flower είναι ένα εργαλείο παρακολούθησης σε πραγματικό χρόνο όσο αναφορά τα tasks που έχουν ανατεθεί στο Celery.

Στο εργαλείο αυτό μπορεί να παρατηρηθεί η πρόοδος της εκτέλεσης καθώς και το ιστορικό των tasks. Δίνεται επίσης η δυνατότητα της εμφάνισης χρήσιμων λεπτομερειών ενός task όπως χρόνος εκκίνησης, χρόνος εκτέλεσης κα. Τέλος, μερικά από τα παραπάνω μπορούν να απεικονιστούν μέσω γραφημάτων και στατιστικών συμπεριλαμβανομένων παραδείγματος χάριν των ποσοστών επιτυχίας ή αποτυχίας των tasks.

Ως προς την εφαρμογή του Flower στο sace, πρώτο βήμα είναι η εγκατάσταση του εργαλείου. Αυτό επιτυγχάνεται με την ακόλουθη εντολή στο terminal:

```
apt-get install flower
```

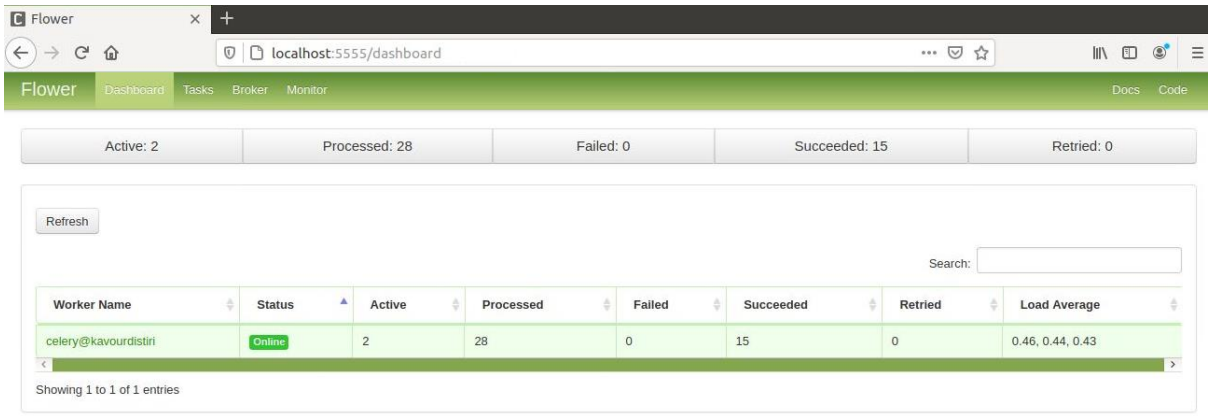
Έπειτα πρέπει να γίνει εισαγωγή του client της εφαρμογής στο Flower. Για να επιτευχθεί αυτό, χρειάζεται να εκκινηθεί το εργαλείο του Flower:

```
flower -A app.client --port=5555
```

Με την παράμετρο *"-A"* ορίζεται ο client και με την παράμετρο *"-port"* ορίζεται η πόρτα που θα "ακούει" το Flower.

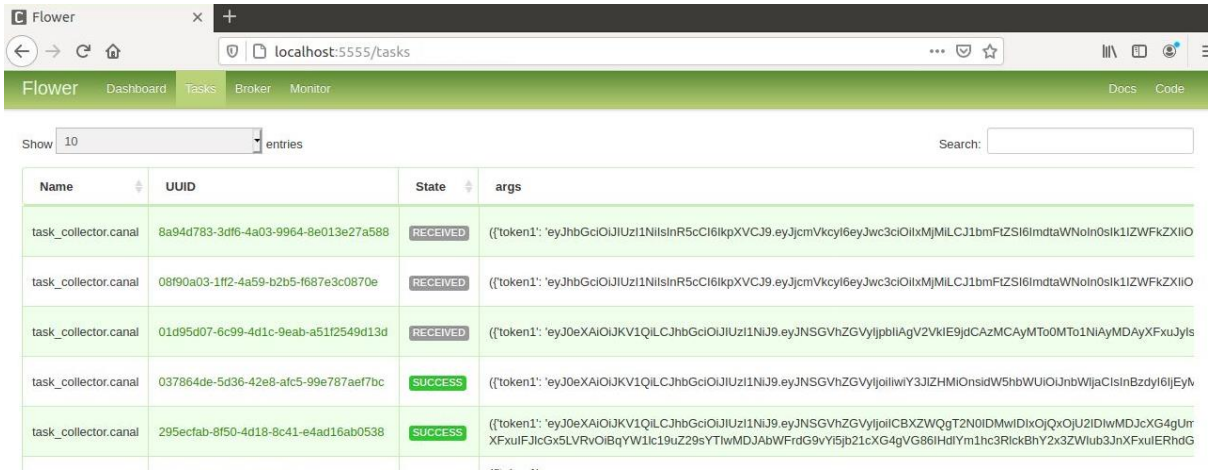
Παραδείγματα στο sace:

Με την αποστολή ενός τουλάχιστον mail header στον collector, το request εισάγεται στην ουρά μέσω του Celery. Στην εικόνα 9.2 πατώντας στο "Dashboard" από την μπάρα πλοήγησης παρατηρείται η κατάσταση των εργασιών.



Εικόνα 9.2 Flower - Dashboard

Επίσης στην εικόνα 9.3 απεικονίζεται μία λίστα από αιτήματα που περιμένουν να εκτελεστούν τα οποία έχουν σημανθεί με State “Received” και άλλα δύο παρατηρείται ότι έχουν εκτελεστεί και έτσι το state έχει αλλάξει σε “Success”.

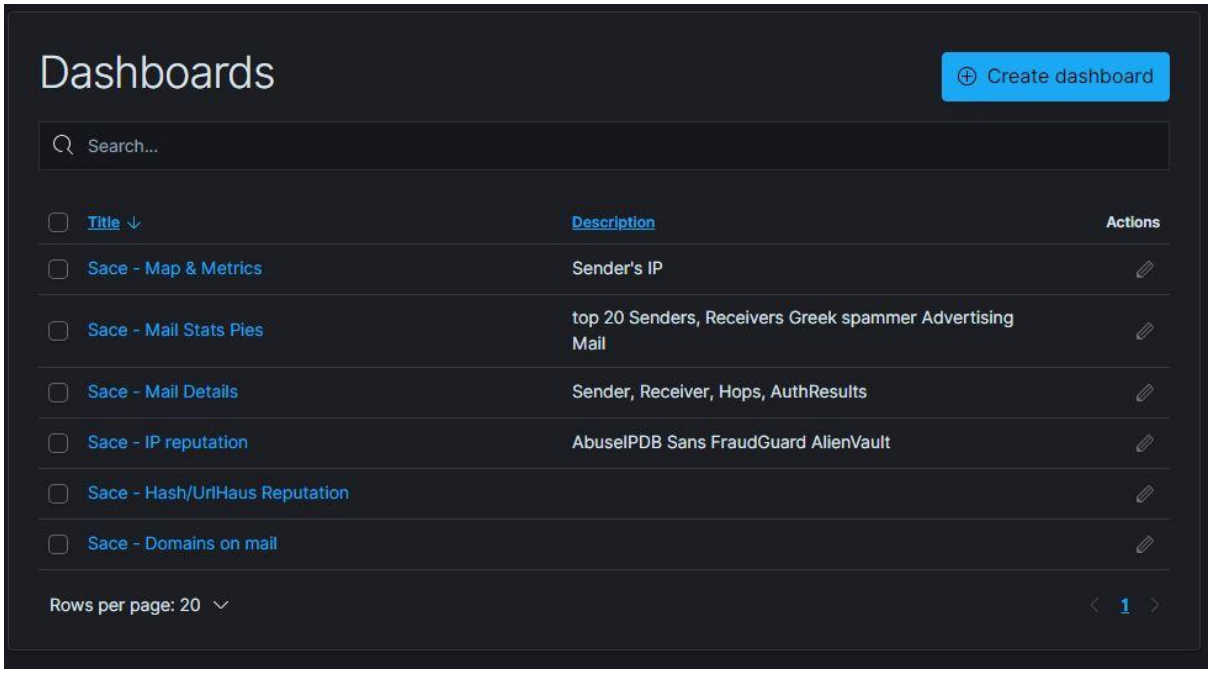


Εικόνα 9.3 Flower – Κατάσταση εργασιών

10. Περιγραφή των αποτελεσμάτων με την χρήση του Kibana

Τα δεδομένα που έχουν εισαχθεί στην εφαρμογή προέρχονται από δύο datasets. Το πρώτο δεν είναι διαθέσιμο διαδικτυακά, αλλά αφορά ένα data set με spam mails του 2018, η πηγή του είναι το Πανεπιστήμιο Θεσσαλίας, ενώ το δεύτερο είναι το “Fraudulent E-mail Corpus CLAIR collection of "Nigerian" fraud emails” _[34]. Να τονιστεί ότι δεν έχει εισαχθεί όλο το σύνολο των μηνυμάτων αλλά μέρος αυτών λόγω των περιορισμών των APIs που χρησιμοποιεί η εφαρμογή. Επίσης κάποια από αυτά είναι πραγματικά spam mails που όμως έχουν ανωνυμοποιηθεί και κάποια είναι «τεχνητά» με σκοπό την επαλήθευση της σωστής λειτουργίας των dashboards.

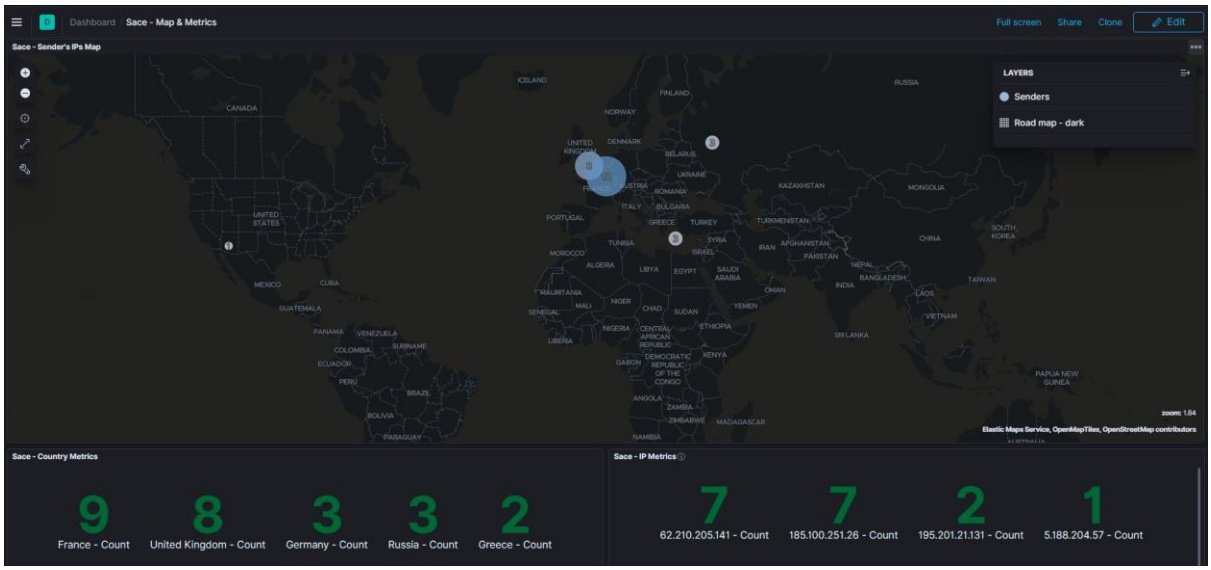
Έξι dashboards έχουν δημιουργηθεί για την αποτύπωση και την ανάλυση των αποτελεσμάτων στο Kibana και πάνω από 30 visualations.



Εικόνα 10.1 Dashboards στο Kibana

SACE – Map & Metrics:

Το συγκεκριμένο dashboard αφορά τον παγκόσμιο χάρτη ώστε από ένα σύνολο δεδομένων να αναπαρασταθεί η τοποθεσία των αποστολέων σε clusters.

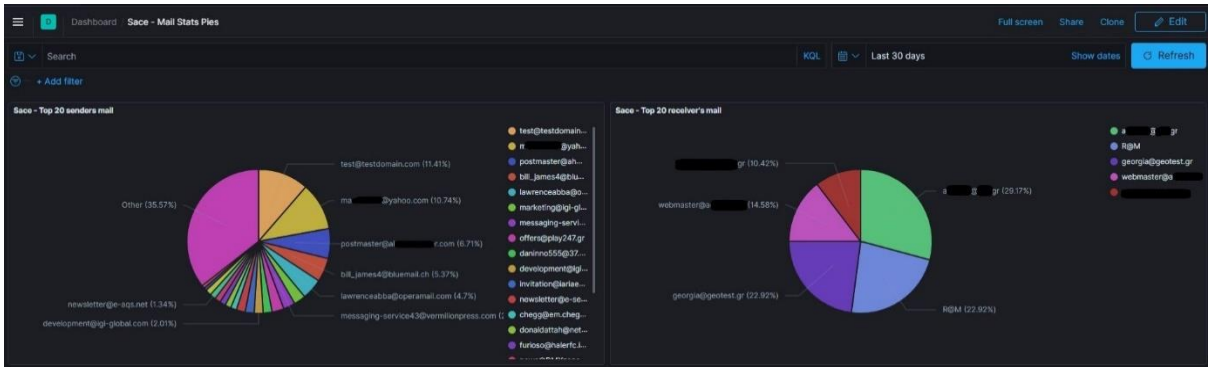


Εικόνα 10.2 – Δημιουργία χάρτη για την απεικόνιση των αποστολέων

Στο dashboard απεικονίζεται ότι τα περισσότερα e-mails που έχει λάβει η εφαρμογή προέρχονται από την κεντρική Ευρώπη. Στο αριστερά κάτω monitor ή visualization αποτυπώνονται οι χώρες από τις οποίες προέρχεται ένα e-mail ενώ αριστερά αποτυπώνονται οι IPs του αποστολέα.

SACE – Mails Stats Pies:

Σ' αυτό το dashboard αποτυπώνονται πληροφορίες και στατιστικά για τα e-mails.



Εικόνα 10.3 – Dashboard "SACE – Mails Stats Pies", 1ο μέρος

Αριστερά φαίνονται οι αποστολείς των μηνυμάτων και δεξιά οι παραλήπτες. Για παράδειγμα ο αποστολέας "maxxxxx@yahoo.com (10.74%)" κατέχει ένα μεγάλο ποσοστό της πίτας, φαίνεται δηλαδή να είναι συχνός αποστολέας σύμφωνα με τα δεδομένα που έχει λάβει η εφαρμογή κάτι το οποίο θα μπορούσε να κινήσει την περιέργεια για την δραστηριότητα του σχετικά με την αλληλογραφία που αποστέλλει.

Εικόνα 10.4 – Dashboard "SACE – Mails Stats Pies", 2ο μέρος

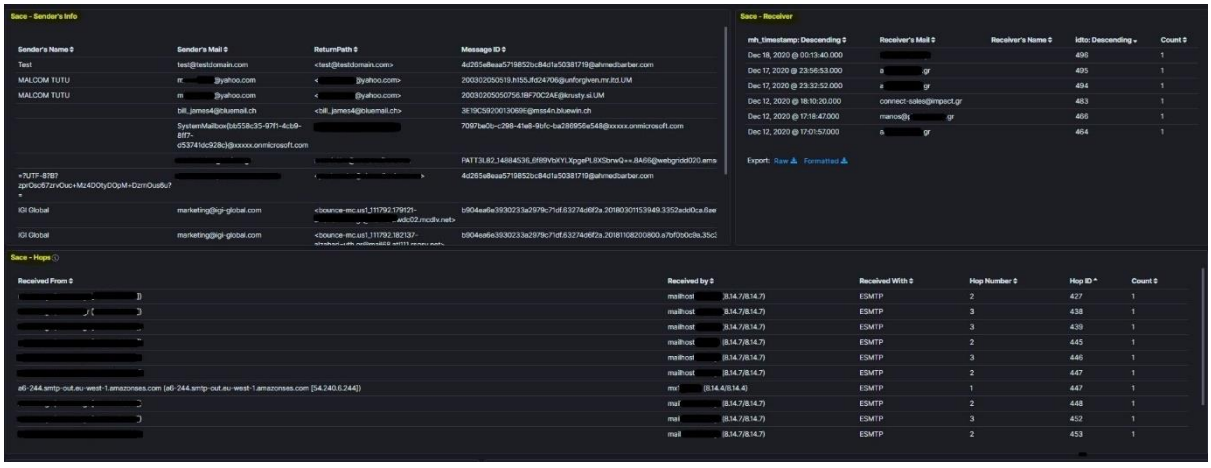
Ταυτόχρονα στο ίδιο dashboard περιέχονται άλλα 4 monitors. Το πάνω αριστερά δείχνει αν από το σύνολο των αποστολέων έχει βρεθεί κάποιος «έλληνας» spammer ψάχνοντας σε μία σχετική λίστα.

Κάτω αριστερά απεικονίζονται τα domains των αποστολέων σύμφωνα με την IP τους, ανακτώντας την πληροφορία αυτήν από το AbuseIPDB.

Δεξιά πάνω αποτυπώνεται το ποσοστό των e-mails απ' των οποίων ο αποστολέας έχει σημανθεί ως μία διαφημιζόμενη εταιρεία σύμφωνα με την αντίστοιχη τεχνική που περιγράφηκε στην ενότητα 6.2. Στο κάτω ακριβώς monitor απεικονίζονται τα σχετικά domains των διαφημιζόμενων εταιρειών.

SACE – Mail Details

Επιλέχθηκε έπειτα να μην αναπαρασταθούν κάποια από τα πεδία του mail header σε γραφήματα αλλά να αναπαρασταθούν σε πίνακες.

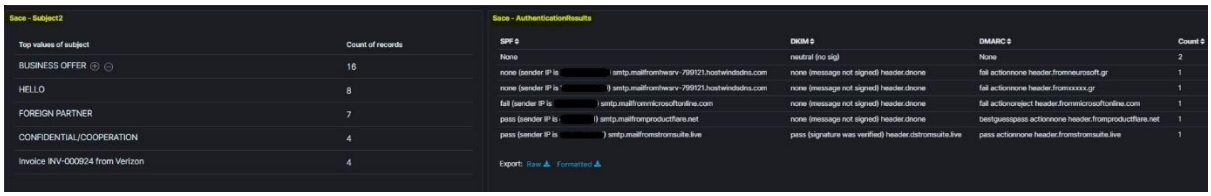


Εικόνα 10.5 – Dashboard "SACE – Mails Details", 1ο μέρος

Σκοπός είναι να μπορεί να αναλυθεί κάποιο e-mail ξεχωριστά φιλτράροντας εύκολα τον «θόρυβο» των πολλών δεδομένων.

Ξεκινώντας από πάνω αριστερά, ο πίνακας αυτός ανακτά πληροφορία αναφορικά με τον αποστολέα, δεξιά αντίστοιχη λογική για πληροφορίες του παραλήπτη, ενώ ο κάτω πίνακας περιλαμβάνει την διαδρομή μέσω διακομιστών που μπορεί να πέρασε ένα e-mail.

Επιπρόσθετα, στο ίδιο dashboard περιλαμβάνεται το πεδίο “Subject” με το οποίο ο αποστολέας πολλές φορές προσπαθεί να δελεάσει τον παραλήπτη ώστε να ασχοληθεί με το e-mail *(βλ. monitor “Sace - Subject”)*. Για παράδειγμα πολλά από τα e-mails του 2ου dataset περιείχαν ως θέμα μηνύματος το “BUSSINESS OFFER” κάτι το οποίο θα μπορούσε να απασχολεί πολλούς χρήστες.

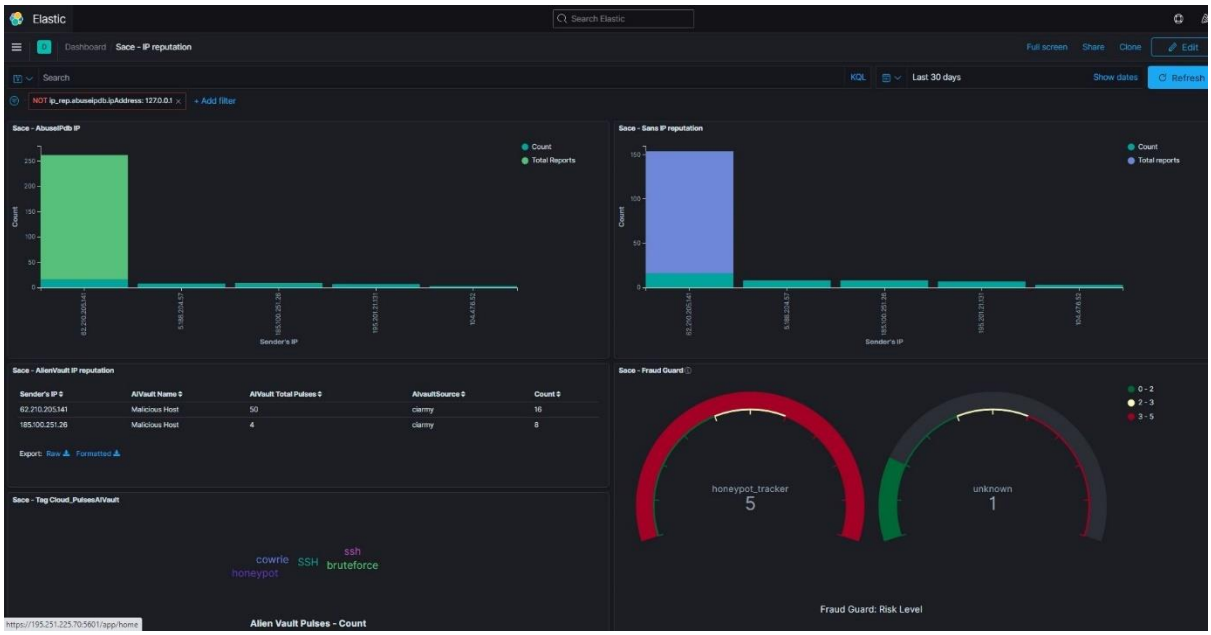


Εικόνα 10.6 – Dashboard “SACE – Mails Details” , 2ο μέρος

Ένα τελευταίο monitor αυτού του dashboard είναι τα αποτελέσματα από το spf, το dkim και το dmarc που απεικονίζεται κάτω δεξιά, θα δοθεί στη συνέχεια ένα παράδειγμα χρήσης του συγκεκριμένου dashboard.

Sace – IP Reputation:

Σκοπός του dashboard είναι η απεικόνιση του reputation που έχει βρεθεί για την IP του αποστολέα σύμφωνα με OSINT πηγές.



Εικόνα 10.7 – Dashboard “SACE – IP reputation” , 1ο μέρος

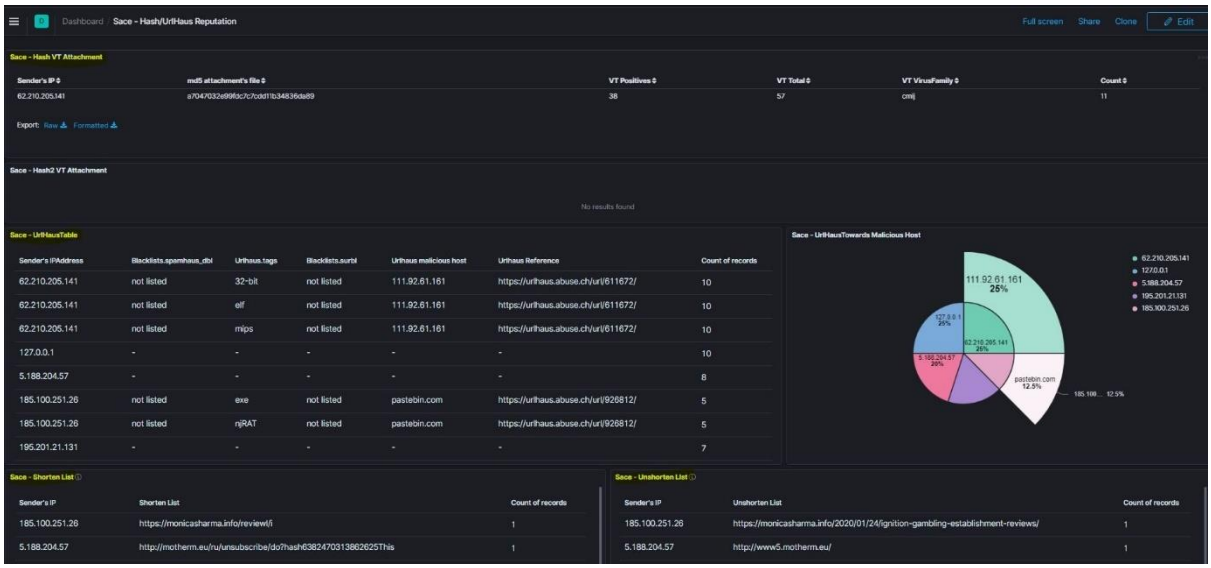
Στο πάνω αριστερά monitor απεικονίζεται η πληροφορία αναφορικά με το reputation της IP σύμφωνα με το AbuseIPDB. Στον άξονα y ορίζεται το σύνολο των φορών που έχει γίνει αναφορά κακόβουλης δραστηριότητας για την εκάστοτε IP. Αντίστοιχη διαδικασία εφαρμόζεται και για το δεξιά monitor που αφορά πληροφορία που έχει ανακτηθεί μέσω του API της SANS.

Κάτω αριστερά αποτυπώνεται πληροφορία που έχει ανακτηθεί από το AlienVault και έχει δημιουργηθεί ένας σχετικός πίνακας και ακριβώς από κάτω έχει προστεθεί ένα επιπλέον monitor με σχετικά tags που υποδηλώνουν τη δραστηριότητα της IP.

Τέλος, το κάτω δεξιά monitor αναπαριστά πληροφορία σύμφωνα με το FraudGuard και υποδηλώνει το επίπεδο υψηλού κινδύνου που έχει οριστεί για κάθε υποβληθείσα IP.

Sace - Hash/URLHaus Reputation:

Το dashboard αυτό αποσκοπεί στην εύρεση κακόβουλης δραστηριότητας, στην οποία μπορεί να υποπέσει ο παραλήπτης.



Εικόνα 10.8 – Dashboard "SACE – IP reputation" , 2ο μέρος

Στο monitor "Sace - Hash VT Attachment" αποτυπώνεται πληροφορία σχετικά με τα αποτελέσματα του VirusTotal.

Στο monitor "Sace - UrlHausTable" αποτυπώνεται πληροφορία σχετικά με τα αποτελέσματα του URLHaus και ταυτόχρονα στο ακριβώς δεξιά monitor, απεικονίζονται στο εξωτερικό τόξο οι κακόβουλοι hosts που αφορά το link που βρίσκεται στο e-mail και στην εσωτερική πίτα απεικονίζεται ο αποστολέας του μηνύματος.

Στο monitor "Sace – Shorten List" φαίνονται οι συντομεύσεις υπερσυνδέσμων που υπάρχουν σ' ένα e-mail και αριστερά οι πραγματικοί σύνδεσμοι που αντιστοιχούν στη shorten list.

Sace – Domains on Mail:

Στο συγκεκριμένο dashboard καταγράφεται η πληροφορία που αφορά τους υπερσυνδέσμους που βρίσκονται σ' ένα e-mail.

Εικόνα 10.9 – Dashboard "SACE – Domain on Mail"

Το πρώτο monitor δείχνει σε χρονικό διάγραμμα τα links που έχουν βρεθεί. Αντίστοιχη πληροφορία δίνεται και στο κάτω αριστερά monitor καταγράφοντας ταυτόχρονα και το σύνολο των φορών που έχει βρεθεί το κάθε link. Τέλος, στο δεξιά monitor αποτυπώνεται η πληροφορία σχετικά το domain που βρέθηκε τις περισσότερες φορές σ' ένα e-mail.

11. Use cases

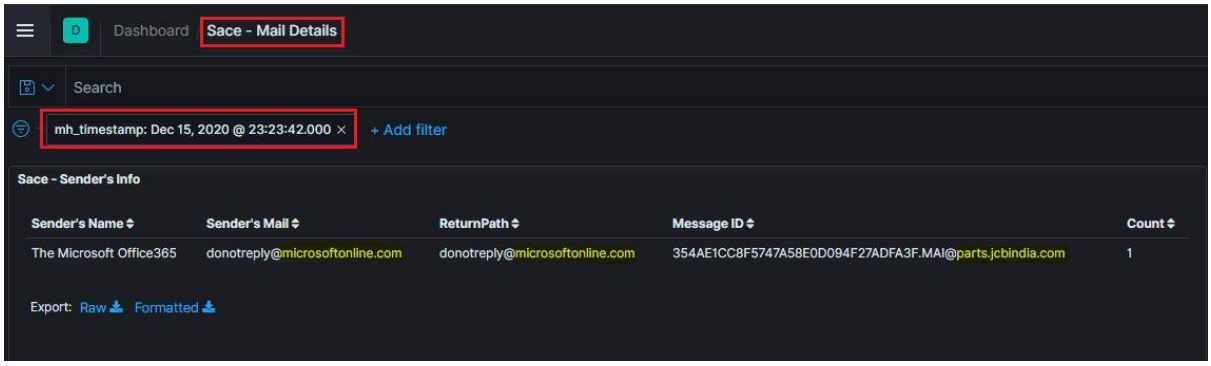
Εύρεση spoofing/phishing e-mail:

Πεδία που υποδεικνύουν την εγκυρότητα του αποστολέα ενός e-mail είναι κυρίως το **spf**, το **dkim** και το **dmarc**. Παράλληλα με τον συνδυασμό και άλλων πεδίων ενός e-mail header μπορούν να βγουν χρήσιμα συμπεράσματα για το αν ένα e-mail είναι spoofing.

Για παράδειγμα, το **Return-Path** καθορίζει από πού προήλθε πραγματικά το e-mail (από ποιον διακομιστή). Στις περισσότερες περιπτώσεις, η κεφαλίδα FROM περιέχει τις ίδιες πληροφορίες με το Return-Path. Όμως ένας επιτιθέμενος μπορεί να αλλάξει αυτό το πεδίο και έτσι το SPF να δώσει ως αποτέλεσμα «pass», καθώς ο έλεγχος θα γίνει μόνο από το πεδίο FROM ψάχνοντας για την αντίστοιχη καταχώρηση στον DNS server.

Ακριβώς όπως είναι δυνατή η πλαστογράφηση άλλων πεδίων μίας κεφαλίδας ενός e-mail, είναι επίσης δυνατή η πλαστογράφηση του πεδίου **message-id.** Τα φίλτρα ανεπιθύμητης αλληλογραφίας ελέγχουν για πιθανόν «άδειο» αναγνωριστικό μηνύματος ή παράνομο μοτίβο αυτού. Παρ' όλ' αυτά, το message-id δεν μπορεί να είναι μια αξιόπιστη ένδειξη ανεπιθύμητων μηνυμάτων. Ένα ύποπτο e-mail από θεωρητικά γνωστή πηγή μπορεί να επαληθευτεί συγκρίνοντας το αναγνωριστικό μηνύματος του e-mail με το αναγνωριστικό του e-mail από την ίδια πραγματική πηγή. Ωστόσο, ο έλεγχος του message-id δεν είναι μια συνεπής μέθοδος ελέγχου ανεπιθύμητων μηνυμάτων, επειδή ένας «καλός» spammer μπορεί να δημιουργήσει το ίδιο μοτίβο του αναγνωριστικού μηνύματος. [35]

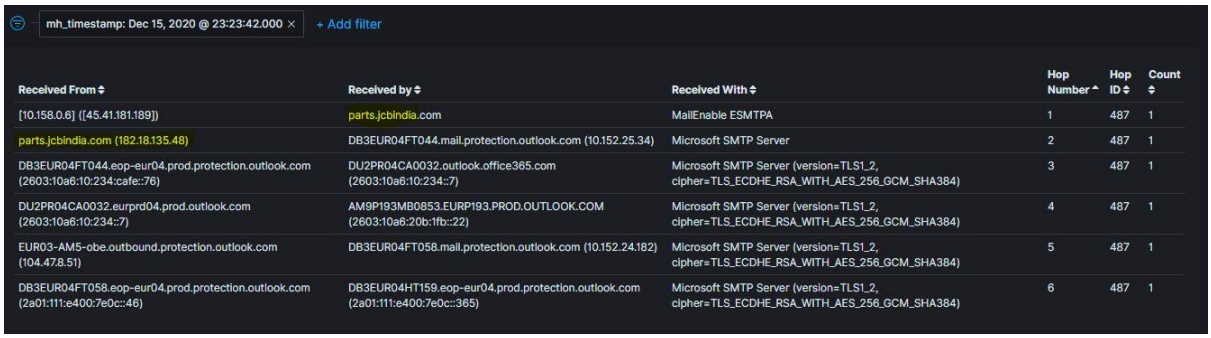
Με το dashboard "Sace - Mail Details" μπορεί να γίνει η ανάκτηση των σχετικών πληροφοριών ώστε να βγουν συμπεράσματα αναφορικά με την αληθινή προέλευση του μηνύματος.



Εικόνα 11.1 – UC: spoofing email, πεδία του αποστολέα

Στην εικόνα 11.1 παρατηρούνται κάποια πεδία ενός e-mail header που παραλήφθηκε στην εφαρμογή στις 15 Δεκεμβρίου στις 23:23. Επίσης παρατηρείται ότι το domain του αποστολέα είναι το microsoftonline.com το οποίο είναι ίδιο με το domain του return-path ενώ είναι διαφορετικό από το πεδίο message-id.

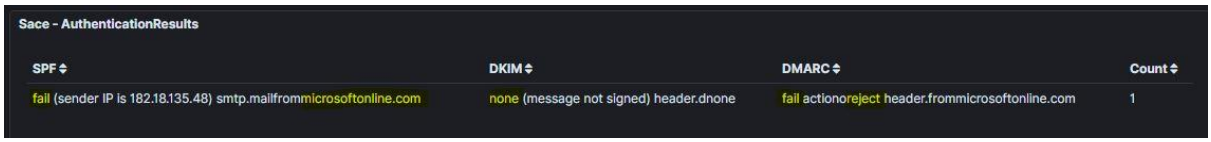
Αντίστοιχα ακολούθως παρατηρούνται τα "hops" δηλαδή οι διακομιστές από τους οποίους πέρασε το e-mail.



Εικόνα 11.2 – UC: spoofing email, hops του email

Τα περισσότερα hops υποδεικνύουν ότι οι διακομιστές σχετίζονται με την Microsoft, όμως από το 1ο hop φαίνεται ότι ο διακομιστής του αποστολέα είναι ο parts.jcbindia.com .

Σκόπιμο και απαραίτητο επίσης είναι να εντοπιστούν τα πεδία spf, dkim και dmarc.



Εικόνα 11.3 – UC: spoofing email, spf, dkim, dmarc αποτελέσματα

Σύμφωνα με την παραπάνω εικόνα, επαληθεύεται πως πρόκειται για ένα spoofed e-mail. Το αποτέλεσμα του spf είναι “fail” καθώς η IP “182.181.135.48” δεν βρίσκεται σε καταγραφή του SPF. Το dkim αντίστοιχα υποδεικνύει ότι το e-mail δεν είναι υπογεγραμμένο και το dmarc είναι επίσης fail.

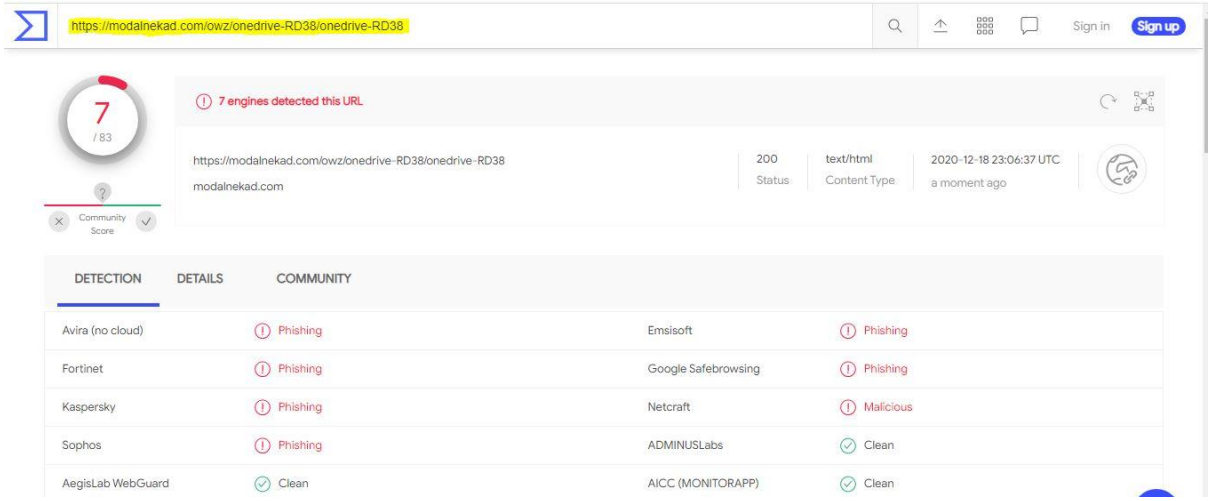
Παράλληλα, στο mail του συγκεκριμένου αποστολέα βρέθηκαν 3 links.

Εικόνα 11.4 – UC: spoofing email, domain των link

Τα δύο από αυτά τα domains περιέχουν το outlook και το google χωρίς αυτό να σημαίνει ότι πρόκειται για μη επιβλαβή links. Το δεύτερο “modalnekad.com” κινεί την περιέργεια καθώς δεν είναι σχετικό με τα άλλα δύο πιο δημοφιλή “domain”. Βρίσκοντας τον υπερσύνδεσμο μέσω του discover

Εικόνα 11.5 – UC: spoofing email, ύποπτο link

και ψάχνοντας για αυτόν στο VirusTotal:



Εικόνα 11.6 – UC: spoofing email, αποτελέσματα στο VT του ύποπτου link

Παρατηρείται ότι 4 αντιϊκά το χαρακτηρίζουν ως phishing.

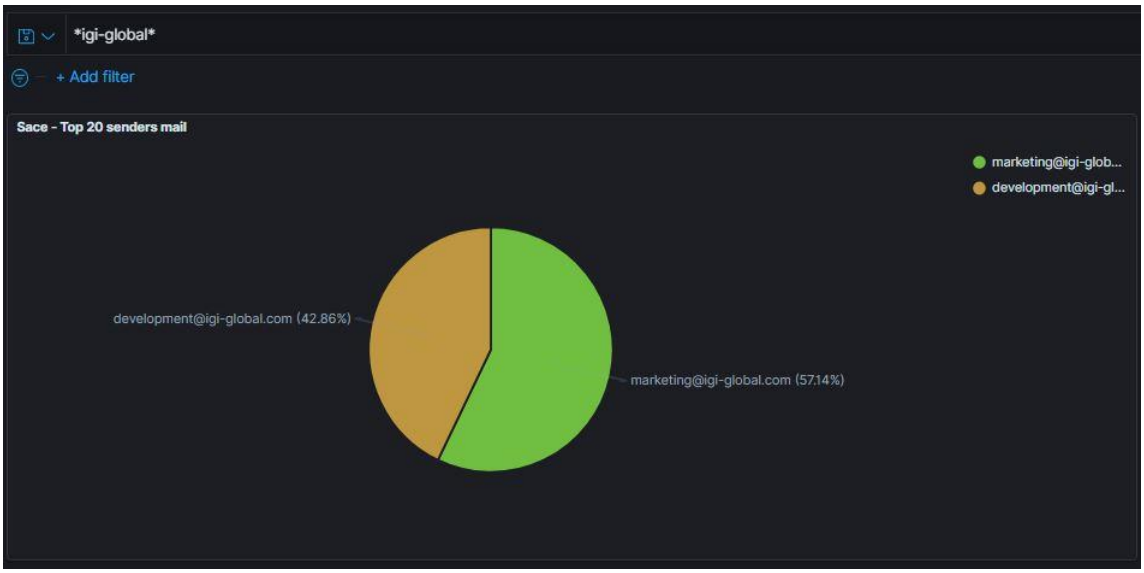
Εύρεση διαφημιζόμενης εταιρείας:

Σύμφωνα με κάποια από τα e-mails του πρώτου dataset που εκχωρήθηκαν στην εφαρμογή στις 12 Δεκεμβρίου, παρατηρήθηκαν μέσω του dashboard “Sace – Domain on Mail” πολλά domains με το όνομα *”igi-global”*.

Εικόνα 11.7 – UC: Εύρεση διαφημιζόμενης εταιρείας μέσω των domain

Παρατηρώντας το monitor “SACE - domainURL3” φαίνεται ότι έχει βρεθεί το αντίστοιχο domain ως το πιο συχνό domain σε σχέση με τα links που περιέχει ένα e-mail.

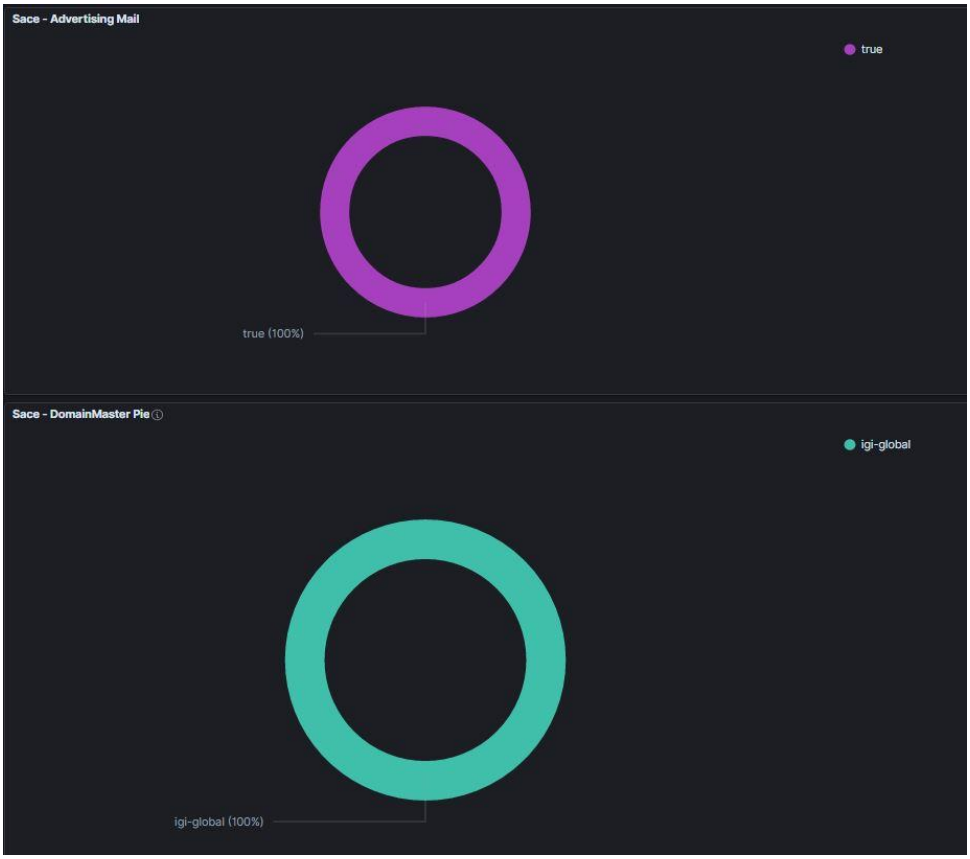
Ανατρέχοντας στο dashboard “Sace – Mail Stats Pies” γίνεται αναζήτηση για το συγκεκριμένο domain.



Εικόνα 11.8 – UC: Εύρεση διαφημιζόμενης εταιρείας αναζήτηση αποστολέων

Έχουν βρεθεί δύο αποστολείς με το ίδιο domain, είναι ο “development@igi-global.com” και ο “marketing@igi-global.com”. Από το όνομα και των δύο συμπεραίνεται εύκολα πως ο αποστολέας δεν είναι ένας απλός χρήστης.

Ως επιβεβαίωση των παραπάνω, δηλαδή ότι τα αποτελέσματα συγκλίνουν στο ότι ο αποστολέας αφορά μία διαφημιζόμενη εταιρεία, παρατίθενται και τα δύο ακόλουθα εικονιζόμενα monitors.



Εικόνα 11.9 – UC: Εύρεση διαφημιζόμενης εταιρείας, True Advertising Mail

Το “igi-global” βρέθηκε ως “true” στο monitor “Advertising Mail”.

Αναζητώντας την IGI Global στο διαδίκτυο συμπεραίνεται ότι η συγκεκριμένη εταιρεία αφορά έναν Διεθνή Ακαδημαϊκό Εκδότη. Επίσης παρέχει την επιλογή για πρόσβαση σε βιβλία και επιστημονικά περιοδικά της IGI Global καθώς και την αναζήτηση σε βάσεις δεδομένων για συντάκτες, συγγραφείς, δημοσιεύσεις κα.

Παράλληλα σε μία τελευταία έρευνα για το συγκεκριμένο domain, παρατηρήθηκε πως πολλά από τα link περιέχουν tracking συμπεριφορά.

Sace - Unshorten List

Sender's IP	Unshorten List	Count of records
127.0.0.1	https://igi-global.us1.list-manage.com/profile	3
127.0.0.1	https://igi-global.us1.list-manage.com/track/click	3
127.0.0.1	https://igi-global.us1.list-manage.com/track/open.php	3

Εικόνα 11.10 – UC: Εύρεση διαφημιζόμενης εταιρείας, tracking link

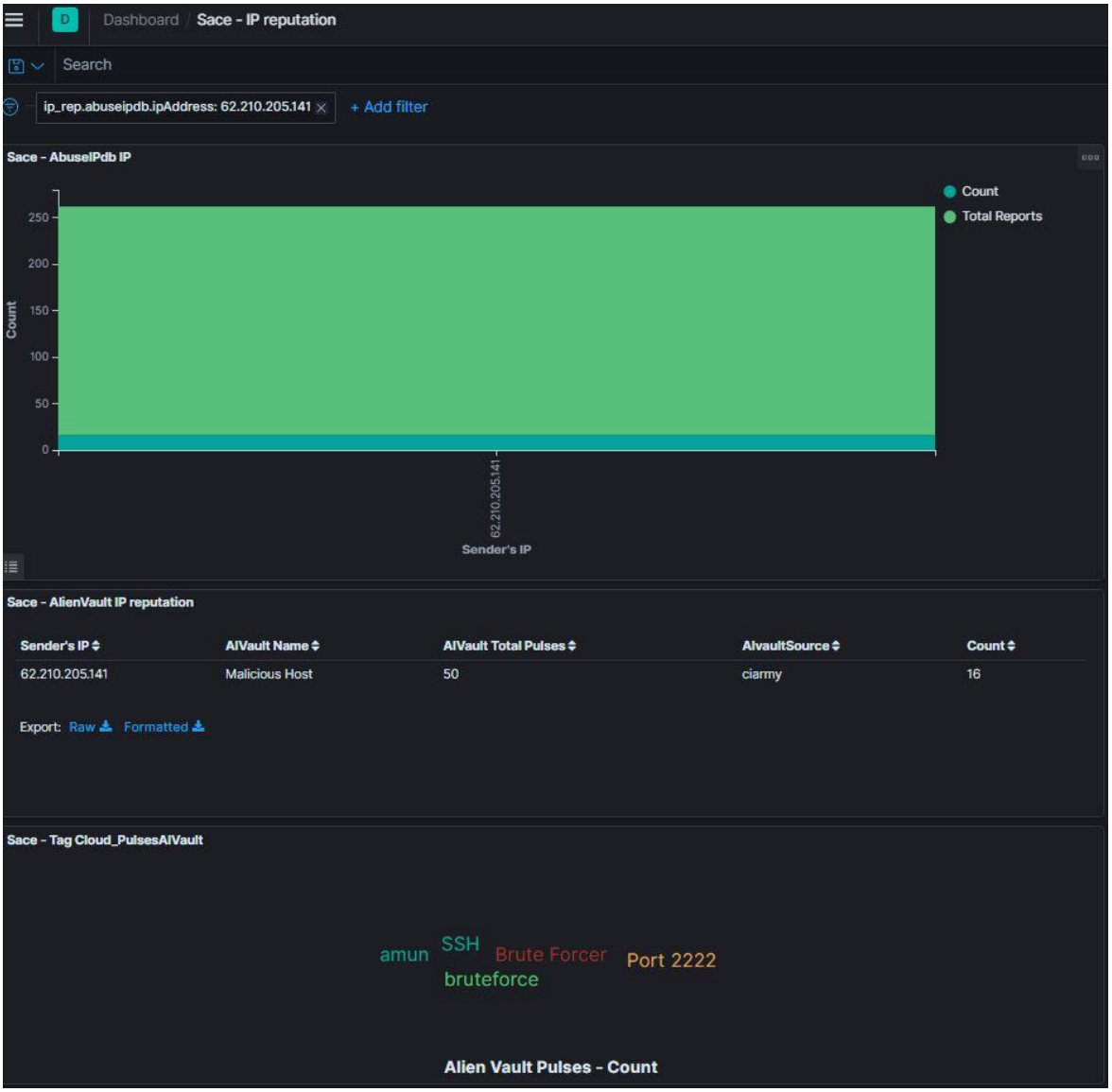
Είναι μία τεχνική που αξιοποιούν πολλές δημοφιλείς υπηρεσίες μάρκετινγκ ηλεκτρονικού ταχυδρομείου ώστε να διαθέτουν δυνατότητες παρακολούθησης των κλικ με σκοπό την παρακολούθηση των χρηστών που ανοίγουν και άρα κάνουν ανάγνωση των μηνυμάτων καθώς και των συνδέσμων που επισκέφθηκαν.

Ανατρέχοντας και πάλι στο discover, αποτυπώνεται ακολούθως η tracking συμπεριφορά του e-mail σύμφωνα με την δομή του link καθώς εκτός από την φράση «track/click» περιέχει και το αντίστοιχο tracking code.

Εικόνα 11.11 – UC: Εύρεση διαφημιζόμενης εταιρείας, εύρεση tracking link στο discover

Εύρεση κακόφημης IP του αποστολέα:

Παρατηρώντας τα δεδομένα της εφαρμογής έγινε εμφανές ότι μία IP ξεχώριζε στο dashboard *“Sace – IP reputation”*.

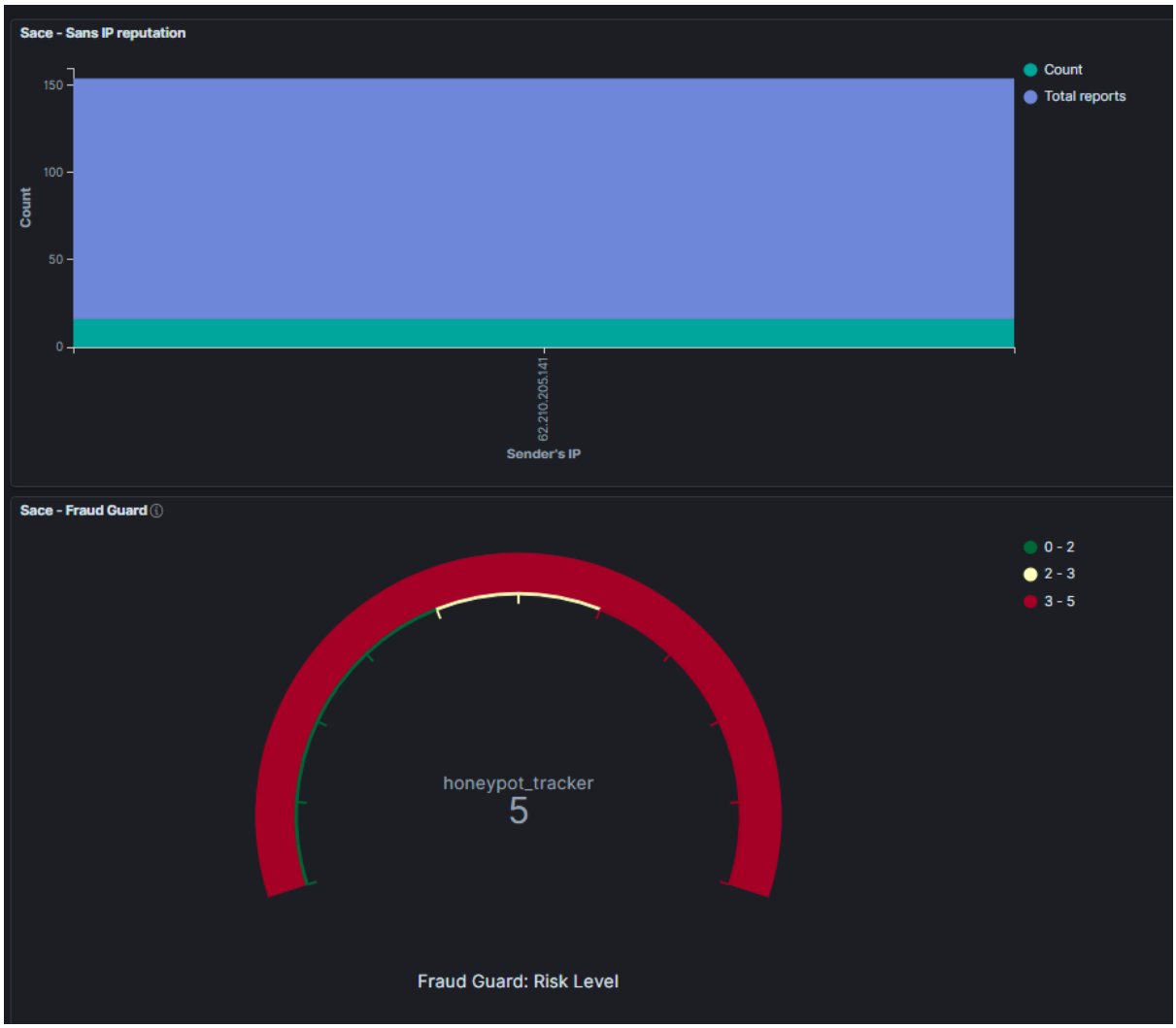


Εικόνα 11.12 – UC: Εύρεση κακόφημης IP του αποστολέα, AbuseIPDB & AlienVault

Η συγκεκριμένη IP της εικόνας 11.12 έχει καταγραφεί και στις τέσσερις πηγές που χρησιμοποιεί η εφαρμογή. Σύμφωνα με το AbuseIPDB έχει αναφερθεί για κακόβουλη δραστηριότητα 246 φορές.

Το AlienVault την έχει σημάνει ως “Malicious Host” και υπάρχουν 50 pulses σχετικά μ’ αυτήν. Παράλληλα τα tags που έχουν δοθεί στην IP αυτήν αποτυπώνονται στο monitor *“Sace - Tag Cloud_PulsesAlVault”*.

Επιπρόσθετα, η IP έχει καταγραφεί στη SANS με 138 αναφορές.

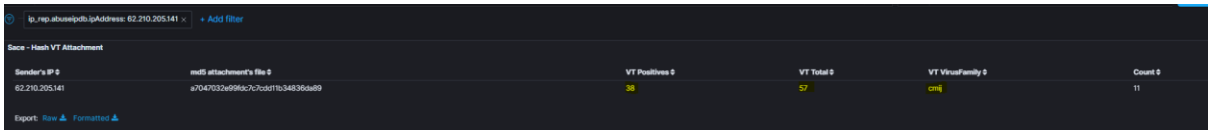


Εικόνα 11.13 – UC: Εύρεση κακόφημης IP του αποστολέα, SANS & FraudGuard

Τέλος, το Fraud Guard την θεωρεί υψηλού κινδύνου IP καθώς το επίπεδο κινδύνου είναι 5 στα 5. Το όνομα απειλής που της έχει απονείμει είναι “honeypot_tracker”.

Εύρεση επισυναπτόμενου αρχείου με ιομορφικό περιεχόμενο:

Στο monitor *“Sace - Hash VT Attachment”* παρατηρείται ότι ο αποστολέας της παραπάνω IP έχει επισυνάψει ένα αρχείο. Από το Virus Total φαίνεται ότι πρόκειται για ένα αρχείο με ιομορφικό περιεχόμενο σύμφωνα με 38 από 57 αντιϊκά. Η οικογένεια ιού η οποία έχει βρεθεί είναι το “cmij”.

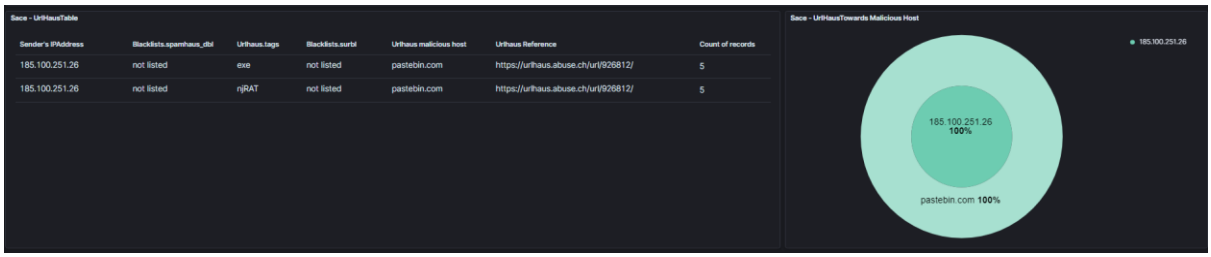


Εικόνα 11.14 – UC: Εύρεση επισυναπτόμενου αρχείου για ιομορφικό περιεχόμενο

Ο ιός αυτός κατηγοριοποιείται στην κατηγορία trojan. Τέτοιου είδους ιοί έχουν ως στόχο να εκτελούν εργασίες χωρίς τη γνώση του χρήστη. Αυτές οι εργασίες περιλαμβάνουν συνήθως τη δημιουργία συνδέσεων απομακρυσμένης πρόσβασης, τη συλλογή πληροφοριών του συστήματος, τη λήψη / μεταφόρτωση αρχείων ή και την απόθεση άλλων κακόβουλων προγραμμάτων στο μολυσμένο σύστημα.

Εύρεση κακόβουλου υπερσυνδέσμου:

Σε μια άλλη αναλυθείσα περίπτωση καταγράφεται η εύρεση κακόβουλου υπερσυνδέσμου σύμφωνα με το URLHaus.



Εικόνα 11.15 – UC: Εύρεση κακόβουλου υπερσυνδέσμου

Ο malicous host είναι το ‘pastebin.com’ και ακολουθώντας το σχετικό reference του URLHaus μέσω της στήλης “UrlHaus Reference” επιβεβαιώνεται ότι ο συγκεκριμένος host σχετίζεται με το nJRAT trojan.

Database Entry

ID:	926812
URL:	https://pastebin.com/raw/dq1J3cjv
URL Status:	Offline
Host:	pastebin.com
Date added:	2020-12-18 01:18:03 UTC
Threat:	Malware download
Google Safe Browsing:	Clean
Spamhaus DBL:	Not listed
SURBL:	Not listed
Quad9:	Not blocked
AdGuard:	Not blocked
Reporter:	@pmelson
Abuse complaint sent (?):	No
Tags:	exe njRAT

Εικόνα 11.16 – UC: κακόβουλος host σ’ εγγραφή της βάσης του URLHaus

12. Συμπεράσματα

Η εφαρμογή “sace” αποτελεί ένα δυναμικό τρόπο παρουσίασης, εύρεσης και ανάλυσης ανεπιθύμητων μηνυμάτων ηλεκτρονικής επικοινωνίας. Η αξιοποίηση σύγχρονων εργαλείων, η πραγματοποίηση API αιτημάτων στις πιο δημοφιλής ιστοσελίδες με σκοπό την εύρεση αναφορών με ενδείξεις κακόβουλης συμπεριφοράς καθώς επίσης και η χρήση του ELK δημιουργούν έναν καινοτόμο τρόπο συλλογής και οπτικοποίησης των δεδομένων ενός ή πολλών κεφαλίδων ενός e-mail.

Συμπερασματικά, δίνεται η δυνατότητα σ’ έναν αναλυτή να αποφανθεί για τον σκοπό και την συμπεριφορά των e-mail, μελετώντας τα πεδία που έχουν εισαχθεί στα ευρετήρια του ElasticSearch και στη συνέχεια με τη χρήση των dashboards του Kibana.

ΠΑΡΑΡΤΗΜΑ Ι

Εγκατάσταση βιβλιοθηκών για τον collector και τον analyzer:

Pip installation:

```
1. sudo apt-get install python-pip
```

Libraries Installation:

```
2. sudo pip install flask
3. sudo pip install Flask-MySQLdb
4. sudo pip install flask-restful
5. sudo pip install Flask-Session
6. sudo pip install headerparser
7. sudo pip install argparse
8. sudo pip install mail-parser
9. sudo pip install python-string-utils
10. sudo pip install numpy
11. sudo pip install pandas
12. sudo pip install -U nltk
13. sudo pip install requests
14. sudo pip install AST
15. sudo pip install uritools
16. sudo pip install Counter
17. sudo pip install mime
18. sudo pip install uuid
19. sudo pip install pytest-shutil
20. sudo pip install pathlib
21. sudo pip install BeautifulSoup
22. sudo pip install MySQL-python
23. sudo pip install --no-deps pyzipcode
24. sudo pip install emaildata
25. sudo pip install netaddr
26. sudo pip install urlunshort
27. sudo pip install tldextract
28. sudo pip install bs4
29. sudo pip install extract-msg
30. sudo pip install ElasticSearch
31. sudo pip install OTXv2
32. sudo pip install lxml
33. sudo pip install termcolor
34. sudo python -m pip install pymongo
35. sudo pip install PyJWT
36. sudo pip install python-JWT
37. sudo pip install cryptography
38. sudo pip install celery
39. sudo pip install redis
40. sudo apt-get install redis-server
41. sudo apt-get install rabbitmq-server
42. sudo pip install flower
```

ΠΑΡΑΡΤΗΜΑ ΙΙ

Εγκατάσταση του ElasticSearch *[36]*

Εισαγωγή του δημοσίου κλειδιού GPG του ElasticSearch στο APT. Το όρισμα *«-fsSL»* αφορά τη σίγαση πιθανών σφαλμάτων.

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –
```

Προσθήκη της λίστας πηγών του Elastic στον κατάλογο *«sources.list.d»*, όπου το APT θα αναζητήσει νέες πηγές:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -
a /etc/apt/sources.list.d/elastic-7.x.list
```

Ενημέρωση των πακέτων, ώστε το APT να διαβάσει το νέο ElasticSearch source:

```
sudo apt update
```

Στη συνέχεια, η εγκατάσταση του ElasticSearch υλοποιείται με την εντολή:

```
sudo apt install elasticsearch
```

Το κύριο configuration file του ElasticSearch είναι το ακόλουθο:

/etc/elasticsearch/elasticsearch.yml

Εκκίνηση του ElasticSearch:

```
sudo systemctl start elasticsearch
```

Η ακόλουθη εντολή επιτρέπει στο ElasticSearch να ξεκινά κάθε φορά που ξεκινά ο διακομιστής:

```
sudo systemctl enable elasticsearch
```

Έλεγχος εγκατάστασης:

Το ES τρέχει από προεπιλογή στην πόρτα 9200. Για τον έλεγχο της σωστής εγκατάστασης τρέχουμε το ακόλουθο με GET αίτημα:

```
curl -X GET 'http://localhost:9200'
```

Το output θα πρέπει να είναι ένα json αποτέλεσμα με στοιχεία σχετικά με το cluster.

Εγκατάσταση του Kibana *[36]*

Λήψη και εγκατάσταση της υπογραφής δημόσιου κλειδιού:

```
1. wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-
   key add -
```

Δημιουργία του Kibana source list:

```
1. echo "deb http://packages.elastic.co/kibana/4.5/debian stable main" | sudo tee -
   a /etc/apt/sources.list.d/kibana-4.5.x.list
```

Η εγκατάσταση του ElasticSearch υλοποιείται με την κάτωθι εντολή αφού πρώτα γίνει update του συστήματος:

```
1. sudo apt-get update && sudo apt-get install kibana
```

Το κύριο configuration file του Kibana είναι το ακόλουθο:

```
/etc/Kibana/kbana.yml
```

Εκκίνηση του Kibana:

```
sudo systemctl start kibana
```

Η ακόλουθη εντολή επιτρέπει στο Kibana να ξεκινά κάθε φορά που ξεκινά ο διακομιστής:

```
sudo systemctl enable kibana
```

Βιβλιογραφία

[1] "A brief history of email: dedicated to Ray Tomlinson," phrasee, 10 March 2016. [Online]. Available: https://phrasee.co/a-brief-history-of-email/. [Accessed 13 11 2020].

[2] "iana," IANA, [Online]. Available: https://www.iana.org/. [Accessed 11 12 2020].

[3] "Message Headers," 01 09 2020. [Online]. Available: https://www.iana.org/assignments/message-headers/message-headers.xhtml. [Accessed 11 12 2020].

[4] D. H. Crocker, "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES," 13 August 1982. [Online]. Available: https://tools.ietf.org/html/rfc822. [Accessed 11 11 2020].

[5] P. Resnick, "Internet Message Format," QUALCOMM Incorporated, April 2001. [Online]. Available: https://tools.ietf.org/html/rfc2822 . [Accessed 11 11 2020].

[6] "Θεματική ενότητα: Ανεπιθύμητες ηλεκτρονικές επικοινωνίες – SPAM," ΑΡΧΗ ΠΡΟΣΤΑΣΙΑΣ ΔΕΔΟΜΕΝΩΝ ΠΡΟΣΩΠΙΚΟΥ ΧΑΡΑΚΤΗΡΑ, [Online]. Available: https://www.dpa.gr/portal/page?_pageid=33,127453&_dad=portal&_schema=PORTAL. [Accessed 10 11 2020].

[7] J. Mehnle, "Sender Policy Framework," OpenSPF.org, 17 04 2010. [Online]. Available: http://www.open-spf.org/Introduction/. [Accessed 13 12 2020].

[8] "Email Header - How to Read and Analyze the Email Header Fields," ARCLAB, [Online]. Available: https://www.arclab.com/en/kb/email/how-to-read-and-analyze-the-email-header-fields-spf-dkim.html. [Accessed 10 11 2020].

[9] D. C. P. H.-B. Tony Hansen, "DomainKeys Identified Mail (DKIM) Service Overview, RFC5585," July 2009. [Online]. Available: https://tools.ietf.org/html/rfc5585. [Accessed 09 11 2020].

[10] "DMARC and the Email Authentication Process," DMARC.org, [Online]. Available: https://dmarc.org/overview/. [Accessed 09 11 2020].

[11] "Everything you need to know about DMARC," [Online]. Available: https://www.dmarcanalyzer.com/dmarc/. [Accessed 09 11 2020].

[12] K. Andersen, "Recommended Usage of the Authenticated Received Chain (ARC)," 22 October 2019. [Online]. Available: https://tools.ietf.org/id/draft-ietf-dmarc-arc-usage-08.html#rfc.section.2.1. [Accessed 08 11 2020].

[13] M. Hoffman, "SEC487: Open-Source Intelligence (OSINT) Gathering and Analysis," SANS, [Online]. Available: https://www.SANS.org/course/open-source-intelligence-gathering. [Accessed 29 11 2020].

[14] "API Documentation - AbuseIPDB," AbuseIPDB , [Online]. Available: https://www.AbuseIPDB.com/api. [Accessed 01 12 2020].

[15] "WHAT IS FRAUDGUARD?," fraudguard, [Online]. Available: https://fraudguard.io/. [Accessed 03 12 2020].

[16] "AT&T Alien Labs Open Threat Exchange," [Online]. Available: https://cybersecurity.att.com/open-threat-exchange. [Accessed 05 12 2020].

[17] "VirusTotal Intelligence API endpoints," VirusTotal , [Online]. Available: https://support.VirusTotal.com/hc/en-us/articles/360002130478-VirusTotal-Intelligence-API-endpoints. [Accessed 02 12 2020].

[18] M. I. D. P. S. (. S. Jeff Sakowicz, M. Wee, M. T. I. C. (. Saisha Agrawal and S. T. (. Army), "Phishing: Spearphishing Link," MITRE ATT&CK, 18 10 2020. [Online]. Available: https://attack.mitre.org/techniques/T1566/002/. [Accessed 09 12 2020].

[19] "URLHaus," URLHaus by Abuse.ch, [Online]. Available: https://URLHaus.abuse.ch/. [Accessed 08 12 2020].

[20] Python Package Index (PyPI), 01 04 2018. [Online]. Available: https://pypi.org/project/unshortenit/. [Accessed 08 12 2020].

[21] M. Jones, J. Bradley, "JSON Web Token (JWT)," May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7519. [Accessed 23 11 2020].

[22] "JSON Web Tokens," Auth0 Docs, [Online]. Available: https://auth0.com/docs/tokens/json-web-tokens. [Accessed 20 11 2020].

[23] S. Krishna, "JSON Web Token (JWT) based client authentication in Message Queuing Telemetry Transport (MQTT)," *Originally written for submission for a PhD course at NTNU,* 7 Mar 2019.

[24] "JWT," Auth0, [Online]. Available: https://jwt.io/. [Accessed 20 11 2020].

[25] "More like this query," elastic, [Online]. Available: https://www.elastic.co/guide/en/ElasticSearch/reference/current/query-dsl-mlt-query.html#query-dsl-mlt-query. [Accessed 25 11 2020].

[26] C. a. Z. T. Gormley, ElasticSearch: the definitive guide: a distributed real-time search and analytics engine, O'Reilly Media, Inc., 2015.

[27] "Put index template API," elastic, [Online]. Available: https://www.elastic.co/guide/en/ElasticSearch/reference/current/indices-templates-v1.html. [Accessed 28 11 2020].

[28] J. Turnbull, The Logstash Book, Version: v5.0.0a (1216eaa), 2016.

[29] "5 Logstash Filter Plugins You Need to Know About," logz.io, 17 Aug 2017. [Online]. Available: https://logz.io/blog/5-Logstash-filter-plugins/ . [Accessed 26 11 2020].

[30] Y. Gupta, Kibana essentials, Packt Publishing Ltd, 2015.

[31] "Jdbc input plugin," elastic, [Online]. Available: https://www.elastic.co/guide/en/Logstash/current/plugins-inputs-jdbc.html. [Accessed 13 11 2020].

[32] P. Dobbelaere and K. S. Esmaili, "Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper," in *Proceedings of the 11th ACM international conference on distributed and event-based systems*, 2017.

[33] "Protocol Extensions," RabbitMQ, [Online]. Available: https://www.rabbitmq.com/extensions.html. [Accessed 19 11 2020].

[34] R. Tatman, "Fraudulent E-mail Corpus CLAIR collection of "Nigerian" fraud emails," Kaggle, [Online]. Available: https://www.kaggle.com/rtatman/fraudulent-email-corpus. [Accessed 10 12 2020].

[35] S. Pasupatheeswaran, "Email 'Message-IDs' helpful for forensic analysis?," School of Computer and Information Science, Edith Cowan University, Perth, Western Australia , 3 12 2008 . [Online]. Available: https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1048&context=adf. [Accessed 22 11 2020].

[36] V. K. Justin Ellingwood, "How To Install ElasticSearch, Logstash, and Kibana (Elastic Stack) on Ubuntu 18.04," digitalocean, 6 11 2018. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-ElasticSearch-Logstash-and-Kibana-elastic-stack-on-ubuntu-18-04. [Accessed 23 11 2020].

[37] C. McLeod, "A framework for distributed deep learning layer design in python," *arXiv preprint arXiv:1510.07303,* 2015.

[38] "Email Header How to Read and Analyze the Email Header Fields," ARCLAB, [Online]. Available: https://www.arclab.com/en/kb/email/how-to-read-and-analyze-the-email-header-fields-spf-dkim.html. [Accessed 10 12 2020].