



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

1^η Εργασία

Ασφάλεια και οπτικοποίηση μεγάλων δεδομένων

Συγγραφείς:

Αλέξανδρος Χαντζάρας, 2122, dit2122dsc@go.uop.gr

Αλέξανδρος Βίτσας, 2103, dit2103dsc@go.uop.gr

Γεώργιος Σπηλιάκος, 2119, dit2119@go.uop.gr

Νικόλαος Κυριακάκης, 2112, dit2112dsc@go.uop.gr

Διδάσκοντες:

Νικόλαος Κολοκοτρώνης, *Αναπληρωτής Καθηγητής*

Νικόλαος Πλατής, *Επίκουρος Καθηγητής*

Μάιος 2022

Πίνακας περιεχομένων

1	Εισαγωγή.....	3
2	Zeek.....	4
2.1	Αρχιτεκτονική Zeek Cluster.....	4
2.2	Network Traffic Slicing.....	5
3	ELK Stack	5
3.1	Filebeat	5
3.2	Elastic Search	6
3.3	Kibana	7
4	User Manual	9
4.1	Configurations:	9
4.2	Ανίχνευση εισβολών.....	10
4.3	Επίθεση με Pcap	12
5	Βιβλιογραφία	15

1 Εισαγωγή

Η ραγδαία εξέλιξη της χρήσης του διαδικτύου, της επέκτασης του IoT στον τομέα των δεδομένων και η ευρεία χρήση υπολογιστών συστημάτων παντός σκοπού έχουν επιφέρει μεγάλο ενδιαφέρον στο χώρο της Κυβερνοασφάλειας. Μεταξύ άλλων ερευνών σχετικά με τις οικονομικές επιπτώσεις των κυβερνοεπιθέσεων το 2017 πραγματοποιήθηκε το WannaCry ransomware attack με αποδέκτες 200.000 μηχανήματα σε 15 χώρες και οικονομικό κόστος 8 δισεκατομμυρίων δολαρίων.

Στον τομέα των του εντοπισμού ανώμαλων λειτουργιών και προστασίας ενός υπολογιστικού συστήματος η βιβλιογραφία παρουσιάζει τη χρήση Machine Learning τεχνικών τόσο για την προσπάθεια πρόληψης όσο και αντιμετώπισης τους. Στην έρευνα [1] αναφέρεται η χρήση ευρέως διαδομένων αλγορίθμων μηχανικής μάθησης όπως ο SVM αλλά και ensemble classifiers που σκοπεύουν να αναγνωρίσουν την εκάστοτε δικτυακή επικοινωνία ως ομαλή ή ύποπτη.

Σε μια επόμενη εργασία παρουσιάζεται η χρήση επιλεγμένου αρχείου 25.000 επιθέσεων προς εκπαίδευση ενός συστήματος μηχανικής μάθησης, με επιλεγμένα [2] και ταξινομημένα, με βάση την σημασία τους, δεδομένων. Από την άλλη πλευρά δεν είναι τόσο εύκολη η εύρεση και χρήση εκπαιδευτικών αρχείων δεδομένων σε σχέση με επιθέσεις δεδομένου ότι τα δεδομένα που αφορούν διαπιστωμένες επιθέσεις αποτελούν συνήθως το 0,01% του αρχείου δεδομένων και αν συνυπολογίσουμε τον αριθμό των false negatives τότε αντιλαμβανόμαστε ότι δημιουργείται μια μεγάλη πρόκληση για την εξέλιξη των μεθόδων μηχανικής μάθησης [3].

Στόχος της παρούσας εργασίας ήταν η δημιουργία ενός κατανεμημένου δικτύου ανίχνευσης εισβολών (IDS). Η δομή του συστήματος που δημιουργήθηκε απαρτίζεται από μια συλλογή εργαλείων, τα οποία δρουν συνεργατικά ώστε να επιτευχθεί επιτήρηση της δικτυακής κίνησης και μετέπειτα να εφαρμοστούν σε αυτή τεχνικές ανίχνευσης ανωμαλιών. Για το σκοπό αυτό δημιουργήθηκε ένα κατανεμημένο zeek cluster αποτελούμενο από 3 workers που λειτουργούν στο ίδιο μηχάνημα. Οι προηγούμενοι κάνουν sniffing τη δικτυακή κίνηση σε κοινό network-interface. Τα captured logs στέλνονται στη στοίβα λογισμικού ELK, η οποία είναι υπεύθυνη για τη συλλογή, οπτικοποίηση και ανάλυση των δεδομένων αυτών. Στη συνέχεια αξιοποιούνται δύο plugins του ELK stack (ElastAlert, Elastic ML) για την ανίχνευση ανωμαλιών στα παραπάνω. Για να επιβεβαιώσουμε την ορθότητα του συστήματος, διενεργήθηκε επανεκπομπή pcap αρχείων από το IoT Network intrusion dataset [4].

2 Zeek

Το zeek εγκαταστάθηκε στο default directory (*/opt/zeek*) και για να προσαρμόσουμε τη λειτουργία του, τροποποιήσαμε το configuration αρχείο που βρίσκεται στη διαδρομή */opt/zeek/etc/node.cfg*. Στο αρχείο αυτό περιλαμβάνεται η επιλογή μεταξύ δύο τρόπων λειτουργίας του Zeek.

Ο πρώτος είναι σε λειτουργία standalone, δηλαδή το Zeek να τρέχει σε ένα πυρήνα του συστήματος και ο άλλος είναι σε cluster λειτουργία. Η δεύτερη λειτουργία είναι και η απαιτούμενη στα πλαίσια της τρέχουσας εργασίας.

2.1 Αρχιτεκτονική Zeek Cluster

Η αρχιτεκτονική cluster χρησιμοποιεί διαφορετικούς κόμβους για να κάνει επιτήρηση της δικτυακής κίνησης με κατανεμημένο τρόπο. Οι βασικοί κόμβοι που απαρτίζουν ένα Zeek cluster είναι οι manager, logger, proxy και ένας αριθμός από workers.

Ο κόμβος manager είναι υπεύθυνος για τη συλλογή logs από τους υπόλοιπους κόμβους και για τη ενοποίηση τους σε ένα κοινό log-file. Αυτή τη διαδικασία την αναλαμβάνει ο κόμβος logger σε περίπτωση που έχει αποφασισθεί να ενεργοποιηθεί και αυτός. Ο τελευταίος ενδέχεται να δημιουργηθεί ώστε να μειωθεί ο φόρτος που θα αναλάμβανε ο manager. Στη συνέχεια, ο proxy κόμβος χρησιμοποιείται όταν υπάρχει ανάγκη το cluster να ισομοιράσει οποιαδήποτε μορφή φόρτου εργασίας και να απελευθερώσει πόρους των υπολοίπων κόμβων. Τέλος, οι worker κόμβοι είναι αυτοί που επιτελούν τη παρακολούθηση της δικτυακής κίνησης και συνήθως αποτελούν το μεγαλύτερο μέρος του cluster. Παρακάτω δίνεται ένα στιγμιότυπο από τις ρυθμίσεις που επιλέχθηκαν.

```
[logger-1]
type=logger
host=localhost
#
[manager]
type=manager
host=localhost
#
[proxy-1]
type=proxy
host=localhost
#
[worker-1]
type=worker
host=localhost
interface=wlp2s0
lb_method=pf_ring
lb_procs=1
#
[worker-2]
type=worker
host=localhost
interface=wlp2s0
lb_method=pf_ring
lb_procs=1
#
[worker-3]
type=worker
host=localhost
interface=wlp2s0
lb_method=pf_ring
```

Εικόνα 1: Το configuration file του Zeek που καθορίζει το τρόπο λειτουργίας του.

2.2 Network Traffic Slicing

Για την επίτευξη του network slicing χρειάστηκε να εγκατασταθεί το PF_RING, ένα είδος kernel socket το οποίο χρησιμοποιεί Direct NIC Access για να επιταχύνει τη συλλογή και μετάδοση πακέτων. Όπως φαίνεται και στην παραπάνω εικόνα επιλέξαμε όλους τους workers να παρακολουθούν το ίδιο network interface, να χρησιμοποιούν το PF_RING και να γίνεται δέσμευση μιας διεργασίας από κάθε worker. Συνεπώς, έχουμε workers που λειτουργούν ως τρεις ξεχωριστές διεργασίες και ο δικτυακός φόρτος μοιράζεται μεταξύ τους.

3 ELK Stack

Το ELK Stack είναι μια σουίτα λογισμικού η οποία συνδυάζει δυνατότητες για επιτήρηση, αποσφαλμάτωση και προστασία IT περιβαλλόντων μεταξύ άλλων. Αποτελείται από 4 κύρια εργαλεία, το Elastic Search, το Logstash, το Filebeat και το Kibana. Αποκτά όλο και μεγαλύτερη δυναμική καθώς εφαρμόζει τεχνικές συλλογής δεδομένων από διάφορες πηγές και κατανεμημένα υπολογιστικά συστήματα. Στις ακόλουθες υποενότητες γίνεται αναφορά σε αυτά τα εργαλεία και παρατίθενται εικόνες από τα configuration files στα οποία έγιναν αλλαγές για να ρυθμιστεί η λειτουργία των πρώτων.

3.1 Filebeat

Το Filebeat είναι υπεύθυνο για τη συλλογή logs, τα οποία μετατρέπονται σε json μορφή από tsv ώστε να είναι δυνατή η ανάγνωσή και επεξεργασία τους από το Elastic Search. Στο επόμενο screenshot φαίνεται το configuration file του Filebeat (*/etc/filebeat/filebeat.yml*) στο οποίο προσδιορίζουμε αν τα logs θα πάνε πρώτα στο Logstash ή αν θα σταλούν κατευθείαν στο Elastic Search. Στη συγκεκριμένη περίπτωση επέλεξαμε να ισχύει το δεύτερο.

```
# Configure what output to use when sending the data collected by the beat.
# ----- Elasticsearch Output -----
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]
```

Εικόνα 2: Προσδιορισμός διαδρομής εξόδου του Filebeat.

Το Filebeat επίσης ρυθμίστηκε να συγκεντρώνει αρχεία από το σημείο που τα αποθηκεύει το Zeek (*/opt/zeek/logs/**). Για να επιτευχθεί αυτό τροποποιήσαμε το αντίστοιχο πεδίο του configuration file που βρίσκεται στο path */etc/filebeat/modules.d/zeek.yml*.

```
- module: zeek
  capture_loss:
    enabled: false
  connection:
    enabled: false
  # var.paths: ["/opt/zeek/logs/current/conn.log", "/opt/zeek/logs/*.conn.json"]
  dce_rpc:
    enabled: false
  dhcp:
    enabled: false
  dnp3:
    enabled: false
  dns:
    enabled: true
    var.paths: ["/opt/zeek/logs/current/dns.log", "/opt/zeek/logs/*.dns.json"]
  dpd:
    enabled: false
  files:
    enabled: true
    var.paths: ["/opt/zeek/logs/current/files.log", "/opt/zeek/logs/*.files.json"]
  ftp:
    enabled: false
  http:
    enabled: true
    var.paths: ["/opt/zeek/logs/current/http.log", "/opt/zeek/logs/*.http.json"]
```

Εικόνα 3: Προσδιορισμός διαδρομής από την οποία θα διαβάζει αρχεία το Filebeat. Εδώ ρυθμίζεται να αντλεί από τα directories στα οποία αποθηκεύονται τα logs του Zeek.

3.2 Elastic Search

Το Elastic Search είναι ένα NoSQL σύστημα βάσεων δεδομένων το οποίο χρησιμοποιείται κυρίως για αναζήτηση και ανάλυση logs. Αποθηκεύει και πραγματοποιεί το indexing αδόμητων δεδομένων και στον τομέα των analytics χρησιμοποιείται παράλληλα με τα υπόλοιπα μέρη του ELK Stack.

Για να προσαρμόσουμε τη λειτουργία του πρέπει να προσδιορίσουμε σε ποια διεύθυνση θα κάνει bind. Στη συγκεκριμένη περίπτωση προσδιορίζουμε να κάνει bind σε όλα τα διαθέσιμα interfaces για λόγους απλότητας. Επίσης, ρυθμίζουμε το port στο οποίο θα ακούει να είναι το 9200, το οποίο είναι και το default port για το Elastic Search.

```
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", ":::1"]
#
discovery.type: single-node
```

Εικόνα 4: Προσδιορισμός της IP, port του Elastic Search και του τρόπου λειτουργίας του.

Στη συνέχεια θέτουμε το μέγεθος στο heap κομμάτι της μνήμης που θα δεσμεύσει το Java Virtual Machine να είναι 256 MB ώστε να τρέξει ομαλά σε σχεδόν οποιοδήποτε υπολογιστικό σύστημα δοκιμαστεί.

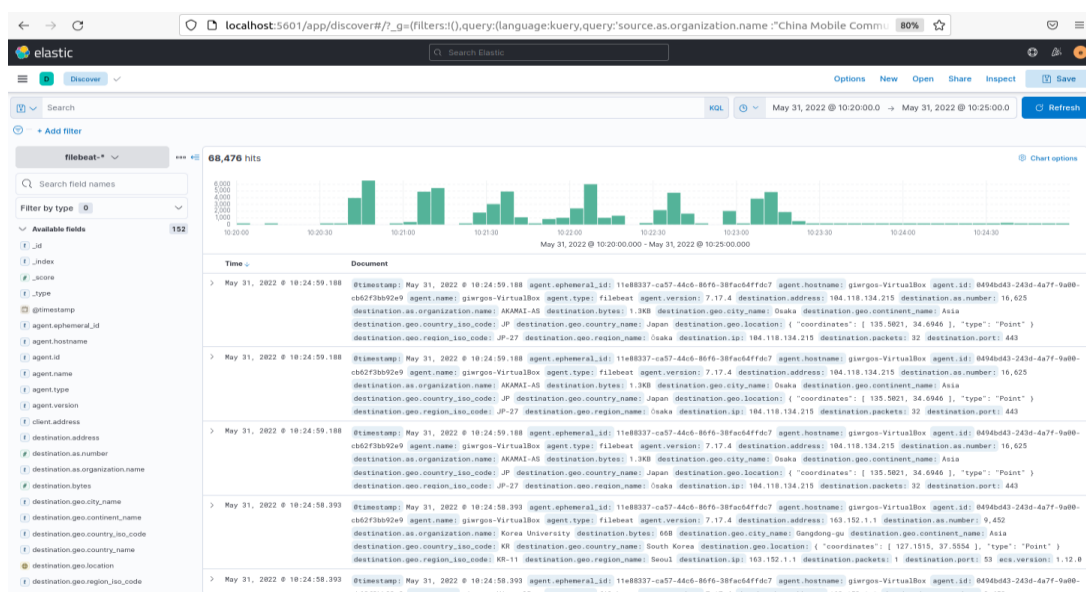
```
#####
## IMPORTANT: JVM heap size
#####
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## and the min and max should be set to the same value. For
## example, to set the heap to 4 GB, create a new file in the
## jvm.options.d directory containing these lines:
##
-Xms256m
-Xmx256m
```

Εικόνα 5: Θέτουμε το μέγιστο χώρο στη μνήμη που θα δεσμεύσει η εικονική μηχανή της Java.

3.3 Kibana

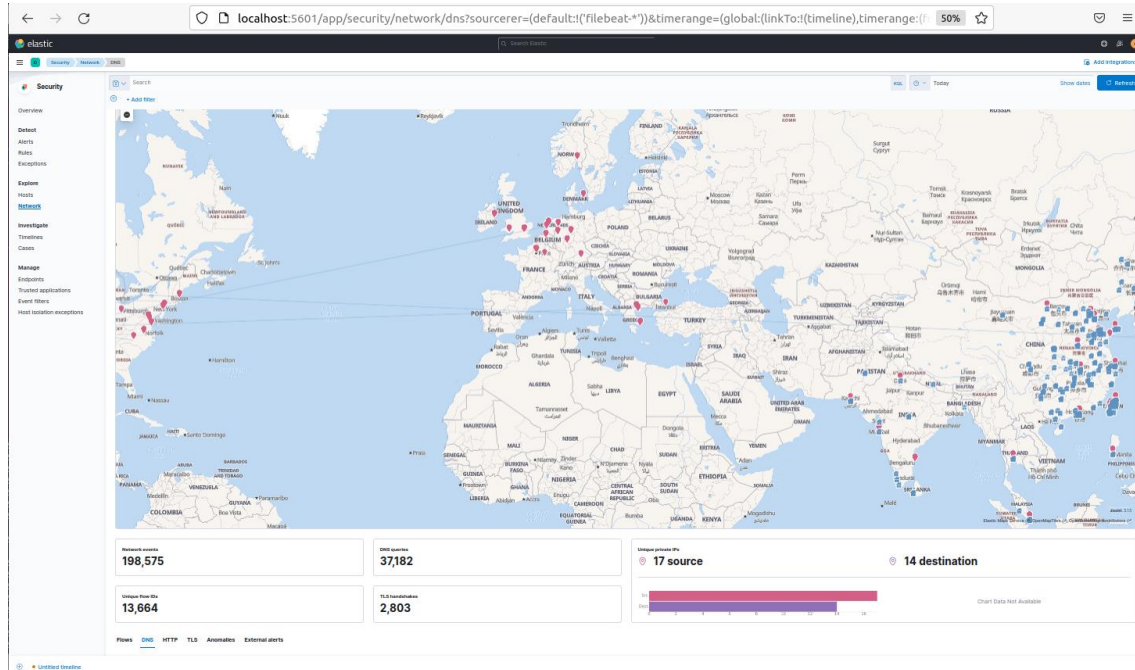
Το Kibana αποτελεί ένα interface το οποίο αντλεί δεδομένα απο το Elastic Search και δημιουργεί οπτικοποιήσεις. Παρέχει πληθώρα διαδραστικών εφαρμογών οι οποίες βοηθούν το χρήστη να εξερευνήσει δεδομένα που έχουν αποθηκευτεί από το Elastic Search με σκοπό τη καλύτερη κατανόηση τους.

Στη περίπτωση που μελετάμε μας είναι χρήσιμο σε πολλούς τομείς. Αρχικά, μέσω του Discover tab μπορούμε να έχουμε μια εποπτική ματιά των events που ανιχνεύθηκαν από το Zeek ώστε να προσδιορίσουμε τις ώρες κατά τις οποίες παρατηρήθηκε η κίνηση, καθώς και το είδος της κίνησης αυτής με ένα πολύ κατανοητό και άμεσο τρόπο. Παρότι τα logs που κρατά το Zeek είναι πλούσια σε πληροφορία, μας είναι αρκετά δύσκολο να τα ερμηνεύσουμε χωρίς μια τέτοια λύση. Παρακάτω παρατίθεται ένα στιγμιότυπο από το Discover.

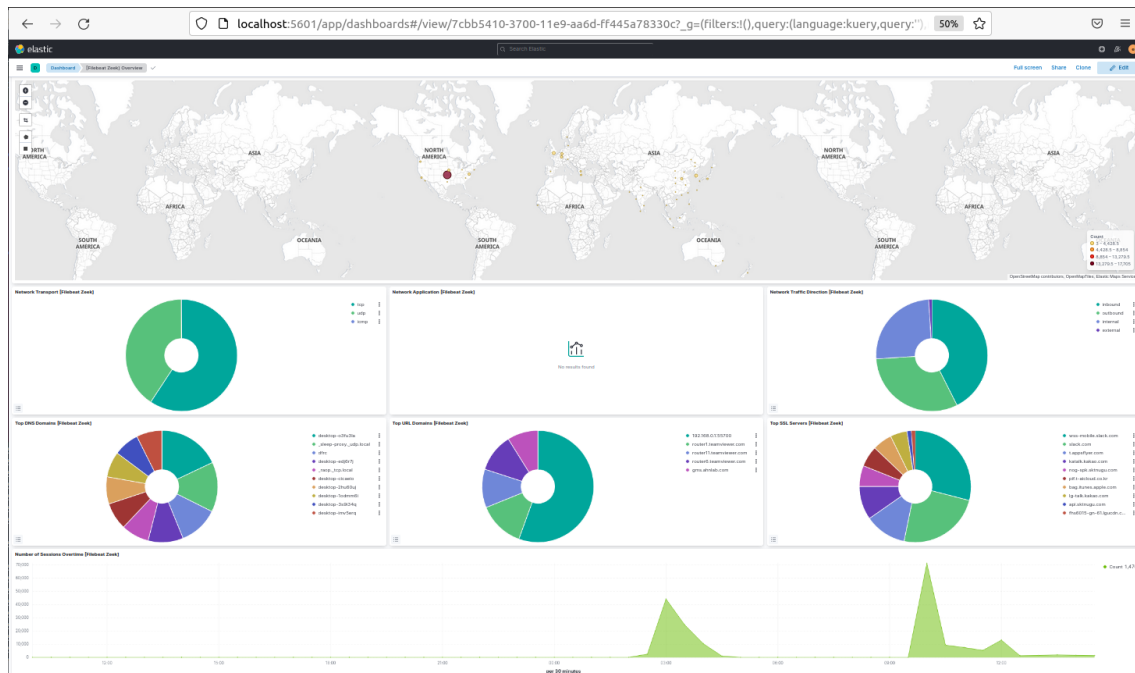


Εικόνα 6: Στιγμιότυπο του Discover, δίνει μια εποπτική ματιά στη δικτυακή κίνηση που έχει συλλεχθεί.

Επιπροσθέτως, θα μπορούσε να μας ενδιαφέρει για παράδειγμα να εντοπίσουμε στο χάρτη κάποια IP. Σε μια τέτοια περίπτωση το Dashboard tab του Kibana είναι ιδιαίτερα χρήσιμο καθώς διενεργεί αυτόματα geolocation tracking. Κατ' αυτό τον τρόπο είναι δυνατό να εντοπισθεί η πηγή κάποιας επίθεσης ή κάποιας ύποπτης σύνδεσης, όπως φαίνεται και στα ακόλουθα στιγμιότυπα.



Εικόνα 7: Στιγμιότυπο του Security Network.



Εικόνα 8: Στιγμιότυπο του Dashboard με στατιστικά από όποια δικτυακή κίνηση έχει συλλεχθεί.

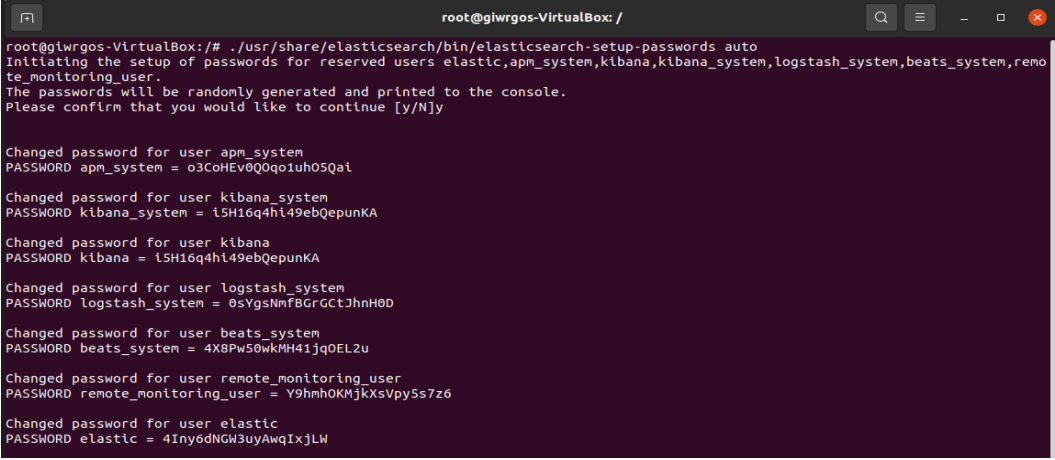
4 User Manual

Σε αυτό το κεφάλαιο θα παρουσιάσουμε αναλυτικά τα βήματα που ακολουθήσαμε κατά την διάρκεια της εκτέλεσης. Το μηχάνημα που χρησιμοποιήσαμε είναι ένα Virtual Machine από το Virtual Box με Ubuntu 16. Έγκαταστήσαμε το Docker έκδοση 20.10.16.

Το directory της εργασίας περιέχει έναν φάκελο “configs” με όλες τις παραμετροποιήσεις που έχουμε για το ELK stack, το Dockerfile από το οποίο θα χτιστεί το image του container μας, τον φάκελο pcap-IoT που περιέχει κάποια από τα pcap αρχεία από το <https://ieee-dataport.org/open-access/iot-network-intrusion-dataset>.

4.1 Configurations:

- Πηγαίνουμε στο directory της εργασίας `$ cd Desktop/SecurityVisualization/`
- Έχοντας εγκατεστημένο το Docker χρησιμοποιούμε την εντολή `“$ sudo docker build -t zeek-base:1.0 .”` Θα χρειαστεί κάποια ώρα ώστε να χτιστεί το image με όλα τα προαπαιτούμενα ~ 2.3GB.
- Στην συνέχεια χρησιμοποιούμε την εντολή `“$ sudo docker run -itd --name zeek -net=host -p 9200:9200 -p 5601:5601 zeek-base:1.0”` για να ξεκινήσουμε το container μας.
- Έπειτα καλούμε την εντολή `“$ sudo docker exec -it zeek bash”` για να ανοίξουμε ένα terminal πάνω στο container μας.
- Αρχικά ενεργοποιούμε το elasticsearch με την εντολή `“service elasticsearch start”` (λόγω προβλήματος στο ξεκίνημα χρειάστηκε να εκτέλεσουμε την εντολή `“sudo sysctl -w vm.max_map_count=262144”` στο host μηχάνημα μας.)
- Μετά πρέπει να φτιάξουμε κωδικούς για τους default χρήστες του elastic καθώς έχουμε την επιλογή `“xpack.security.enabled: true”` στο elasticsearch.yml για το configuration του Elasticsearch. Εκτελούμε την εντολή `“./usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto”` και κρατάμε τους κωδικούς όπως φαίνεται στην παρακάτω φωτογραφία.



```
root@glwrgos-VirtualBox: /
root@glwrgos-VirtualBox:/# ./usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.
The passwords will be randomly generated and printed to the console.
Please confirm that you would like to continue [y/N]y

Changed password for user apm_system
PASSWORD apm_system = o3CoHEv0Q0qoiuh05Qai

Changed password for user kibana_system
PASSWORD kibana_system = i5H16q4h149ebQepunKA

Changed password for user kibana
PASSWORD kibana = i5H16q4h149ebQepunKA

Changed password for user logstash_system
PASSWORD logstash_system = 0sYgsNmFBGrGctJhnH0D

Changed password for user beats_system
PASSWORD beats_system = 4X8Pw50wKMH41jq0EL2u

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = Y9hnh0KMKjKsVpy5s7z6

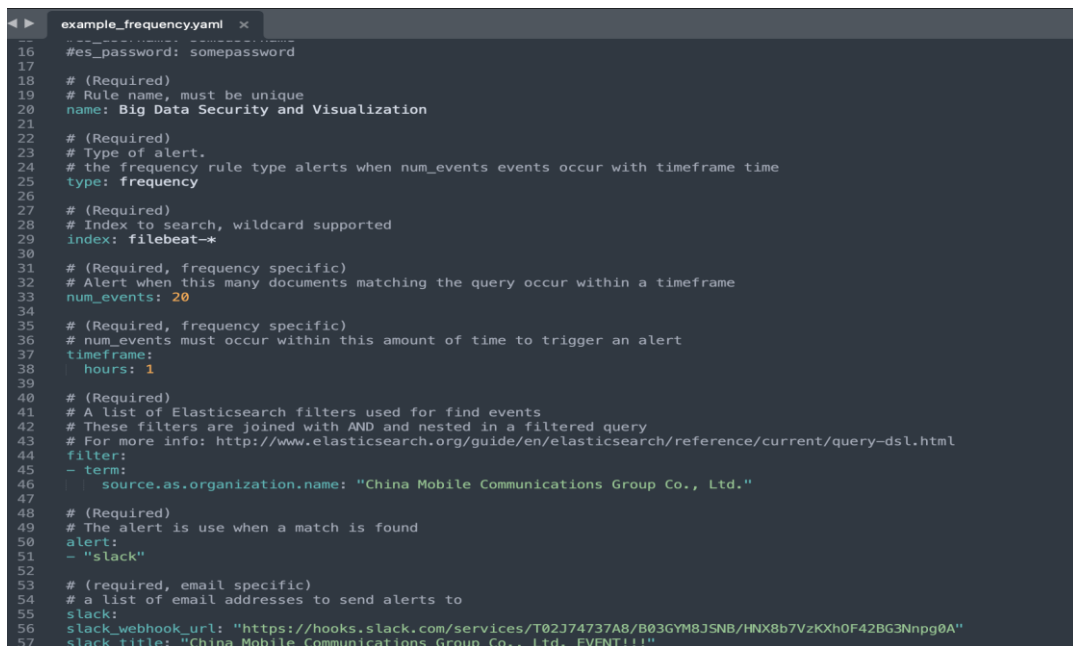
Changed password for user elastic
PASSWORD elastic = 4Iny6dNGW3uyAwqIxjLW
```

Εικόνα 9: Δημιουργία passwords για ενεργοποίηση του xpack.security.

- Ανοίγουμε με έναν editor το kibana.yml configuration και αντικαθιστούμε τα πεδία elasticsearch.username, elasticsearch.password με το key-value του kibana από το προηγούμενο βήμα με την εντολή `"nano etc/kibana/kibana.yml"`
- Ενεργοποιούμε το kibana με την εντολή `"service kibana start"` και βεβαιωνόμαστε μετά από κάποια δευτερόλεπτα ότι μπορούμε να δούμε το elastic στον browser μας `"localhost:5601"` και συνδεόμαστε με το elastic username/password του προηγούμενου βήματος.
- Έπειτα ξεκινάμε το zeek με την εντολή `"zeekctl deploy"`. Ενδεχομένως να χρειάζεται να αλλάξει η επιλογή interface στο `/opt/zeek/etc/node.cfg` για να είναι σύμφωνη με το interface του μηχανήματος στο οποίο γίνεται η δοκιμή.
- Για το filebeat χρησιμοποιούμε την ίδια λογική για το configuration των κωδικών όπως στο kibana με την εντολή `"nano etc/filebeat/filebeat.yml"`
- Εκτελούμε τις εντολές `"filebeat setup"` και `"service filebeat start"`
- Τέλος εκτελούμε τις εντολές `"cd elastalert/"` `"nano config.yaml"` και αντικαθιστούμε τις τιμές es_username, es_password με τα key-value των κωδικών του elastic που παράξαμε σε προηγούμενο βήμα.
- Εκτελούμε την εντολή `"elastalert-create-index"` και ενεργοποιούμε το elastalert με την εντολή `"python3 -m elastalert.elastalert --verbose --rule example_rules/example_frequency.yaml"` για να τρέξει τον κανόνα που έχουμε φτιάξει στο configuration μας

4.2 Ανίχνευση εισβολών

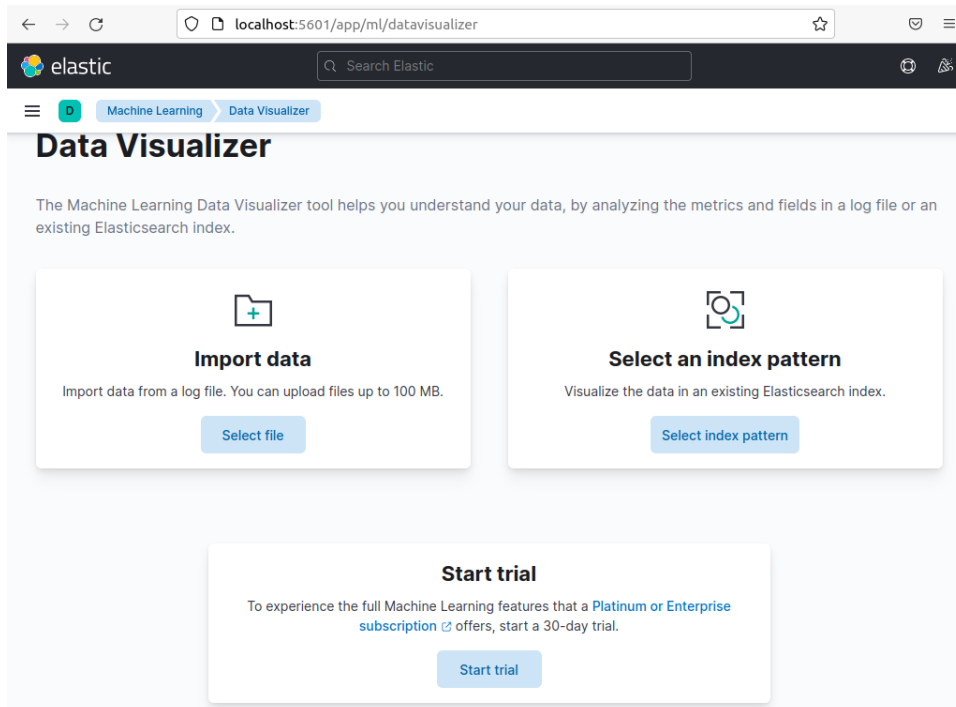
- Έχουμε ενεργοποιήσει το Elastalert για να εφαρμόζει κανόνες. Θεωρούμε ένα συγκεκριμένο source.organization.name ως επικίνδυνο οπότε φτιάξαμε έναν frequency κανόνα που αν εντοπίσει 20 φορές σε διάστημα 1 ώρας κάποιο log με από αυτόν τον οργανισμό μας ειδοποιεί μέσω Slack.



```
example_frequency.yaml
16 #es_password: somepassword
17
18 # (Required)
19 # Rule name, must be unique
20 name: Big Data Security and Visualization
21
22 # (Required)
23 # Type of alert.
24 # the frequency rule type alerts when num_events events occur with timeframe time
25 type: frequency
26
27 # (Required)
28 # Index to search, wildcard supported
29 index: filebeat-*
30
31 # (Required, frequency specific)
32 # Alert when this many documents matching the query occur within a timeframe
33 num_events: 20
34
35 # (Required, frequency specific)
36 # num_events must occur within this amount of time to trigger an alert
37 timeframe:
38   hours: 1
39
40 # (Required)
41 # A list of Elasticsearch filters used for find events
42 # These filters are joined with AND and nested in a filtered query
43 # For more info: http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/query-dsl.html
44 filter:
45   - term:
46     source.as.organization.name: "China Mobile Communications Group Co., Ltd."
47
48 # (Required)
49 # The alert is use when a match is found
50 alert:
51   - "slack"
52
53 # (required, email specific)
54 # a list of email addresses to send alerts to
55 slack:
56   slack_webhook_url: "https://hooks.slack.com/services/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"
57   slack_title: "China Mobile Communications Group Co., Ltd. EVENT!!!"
```

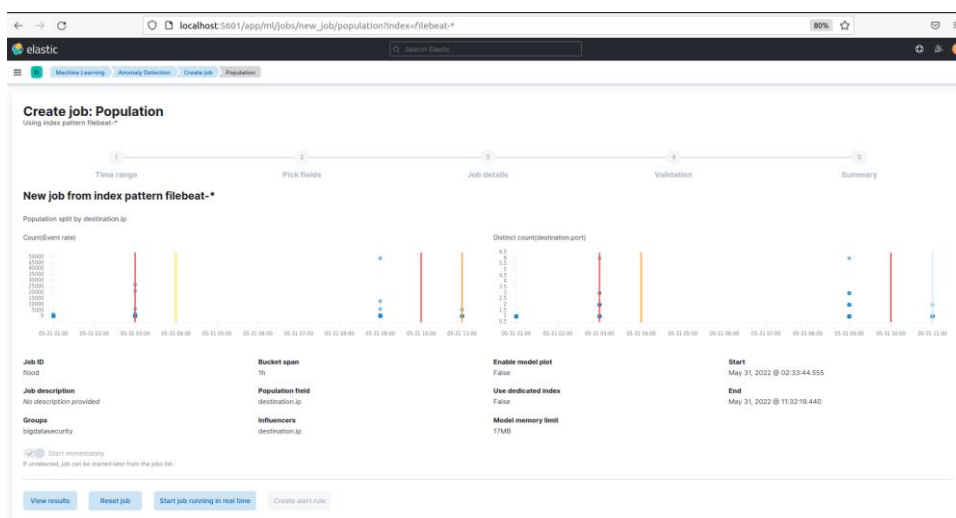
Εικόνα 10: Ορισμός παραμέτρων Elast Alert και ορισμός αποστολής ειδοποιήσεων στο slack.

- Έπειτα θέλουμε να ενεργοποιήσουμε το ElasticML εργαλείο που προσφέρεται από το ELK stack. Πηγαίνουμε στον browser “localhost:5601” στο Analytics/Machine learning tab και ενεργοποιούμε το trial όπως φαίνεται στην παρακάτω εικόνα:

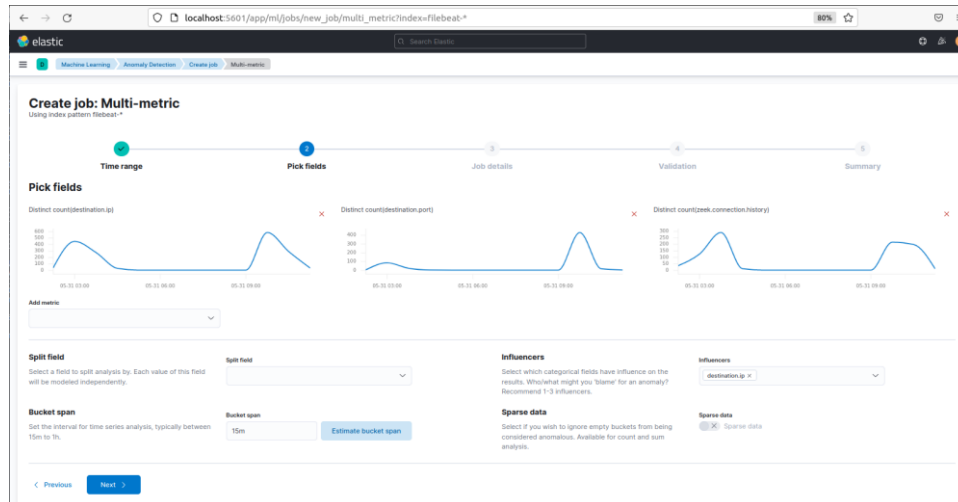


Εικόνα 11: Ενεργοποίηση trial λογαριασμού για πρόσβαση στο Elastic ML

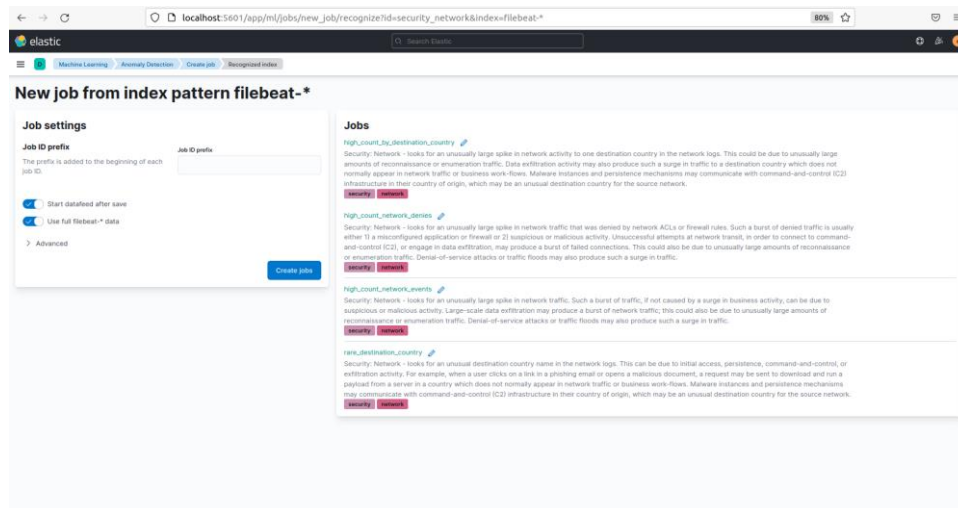
- Στην συνέχεια θα φτιάξουμε ένα Job στο Machine Learning tab για να παρατηρήσουμε τον πληθυσμό των events σε κάποια από τα fields των logs όπως για παράδειγμα το zeek.connection.state που παράγει το zeek, και τα destination.ip και destination.port. Ακόμα ενεργοποιήσαμε το έτοιμο πακέτο Network Security.



Εικόνα 12: Δημιουργία population job για τη ανάλυση αριθμού από events.



Εικόνα 13: Δημιουργία multimeric job για ανίχνευση SYN-DOS attack.



Εικόνα 14: Αξιοποίηση έτοιμου πακέτου για network security που εντοπίζει συχνά εμφανιζόμενες ύποπτες δικτυακές κινήσεις.

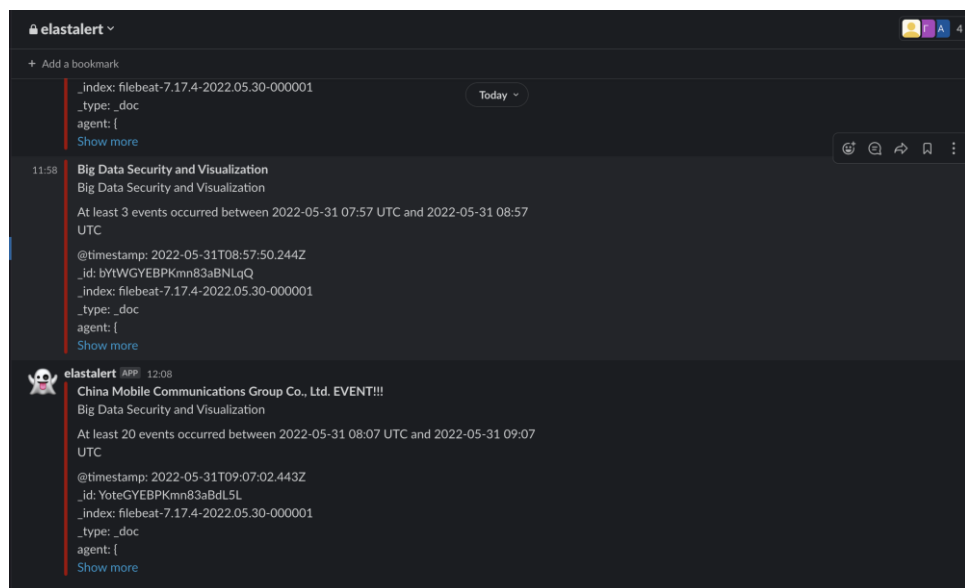
4.3 Επίθεση με Pcap

- Αφού έχουμε στήσει την ανίχνευση εισβολών με κανόνες και με το πακέτο Anomaly Detection δοκιμάζουμε επίθεση με pcap αρχεία.
- Αρχικά κατεβάζουμε το module tcpreplay με την εντολή `apt-get install tcpreplay`
- Έπειτα πηγαίνουμε στον φάκελο pcaps-IoT και εκτελούμε τις ακόλουθες εντολές `tcpreplay --intf1=enp0s3 scan-hostport-6-dec.pcap`, `tcpreplay --intf1=enp0s3 dos-synflooding-3-dec.pcap`, `tcpreplay --intf1=enp0s3 dos-synflooding-1-dec.pcap`, `tcpreplay --intf1=enp0s3 mirai-udpflooding-1-dec.pcap` για να κάνουμε την επανεκπομπή των πακέτων που είχαν σταλεί σε περιπτώσεις παρελθοντικών επιθέσεων.

```
root@giwrgos-VirtualBox: /pcaps-IoT# tcpreplay --intf1=enp0s3 dos-synflooding-1-dec.pcap
Warning in send_packets.c:send_packets() line 644:
Unable to send packet: Error with PF_PACKET send() [637]: Message too long (errno = 90)
Actual: 636 packets (465328 bytes) sent in 7.17 seconds
Rated: 64862.6 Bps, 0.518 Mbps, 88.65 pps
Statistics for network device: enp0s3
  Successful packets: 636
  Failed packets: 1
  Truncated packets: 0
  Retried packets (ENOBUFS): 0
  Retried packets (EAGAIN): 0
root@giwrgos-VirtualBox: /pcaps-IoT# tcpreplay --intf1=enp0s3 dos-synflooding-3-dec.pcap
Warning in send_packets.c:send_packets() line 644:
Unable to send packet: Error with PF_PACKET send() [7707]: Message too long (errno = 90)
Actual: 7706 packets (751579 bytes) sent in 25.67 seconds
Rated: 29273.7 Bps, 0.234 Mbps, 300.14 pps
Statistics for network device: enp0s3
  Successful packets: 7706
  Failed packets: 1
  Truncated packets: 0
  Retried packets (ENOBUFS): 0
  Retried packets (EAGAIN): 0
root@giwrgos-VirtualBox: /pcaps-IoT# tcpreplay --intf1=enp0s3 scan-hostport-6-dec.pcap
Warning in send_packets.c:send_packets() line 644:
Unable to send packet: Error with PF_PACKET send() [998]: Message too long (errno = 90)
Actual: 997 packets (541484 bytes) sent in 8.10 seconds
Rated: 66842.4 Bps, 0.534 Mbps, 123.07 pps
Statistics for network device: enp0s3
  Successful packets: 997
  Failed packets: 1
  Truncated packets: 0
  Retried packets (ENOBUFS): 0
  Retried packets (EAGAIN): 0
root@giwrgos-VirtualBox: /pcaps-IoT#
```

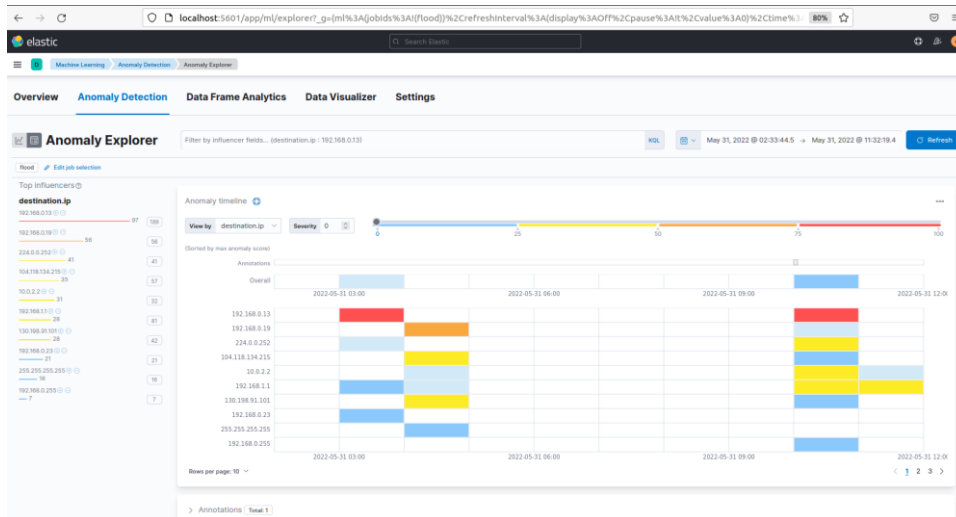
Εικόνα 15: Ενεργοποίηση TCP-replay attack με DOS-SYN flooding.

- Παρατηρούμε ότι τις ώρες των επιθέσεων ο κανόνας του ElastAlert μας ειδοποίησε στο Slack:

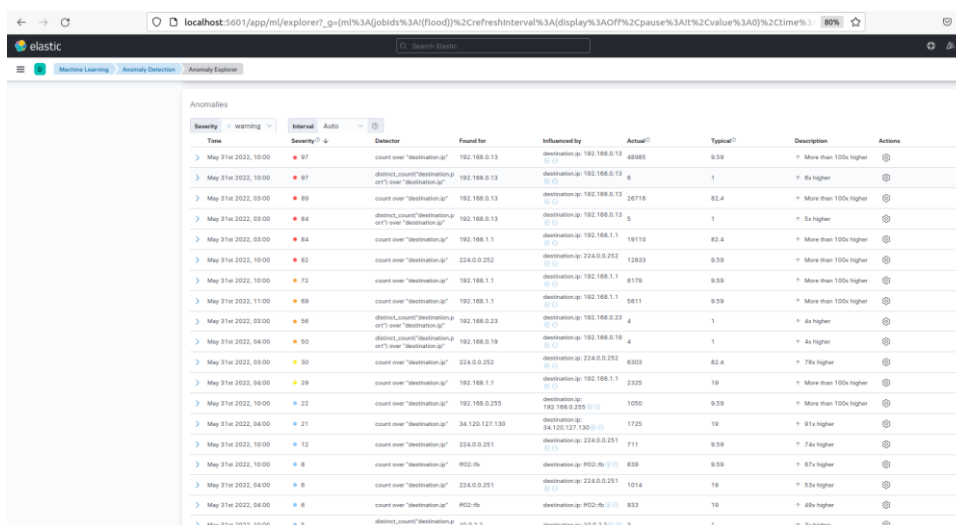


Εικόνα 16: Στιγμιότυπο αποστολής μηνύματος από το Elast Alert στο slack.

- Καθώς και ότι όταν τρέξαμε ανάλυση στα δεδομένα μας μέσω του Anomaly Detection βρήκε αρκετές περιπτώσεις IP οι οποίες είχαν ξεπεράσει κατά πολύ την τυπική συμπεριφορά όπως φαίνεται και στις ακόλουθες εικόνες.



Εικόνα 17: Αποτελέσματα για flood anomaly.



Time	Severity	Detector	Found for	Influenced by	Actual	Typical	Description	Actions
May 31st 2022, 10:00	97	count over "destination_ip"	192.168.0.13	destination_ip: 192.168.0.13	48985	9.59	More than 100x higher	
May 31st 2022, 10:00	97	distinct_count("destination_ip") over "destination_ip"	192.168.0.13	destination_ip: 192.168.0.13	4	1	5x higher	
May 31st 2022, 03:00	89	count over "destination_ip"	192.168.0.13	destination_ip: 192.168.0.13	28716	82.4	More than 100x higher	
May 31st 2022, 03:00	84	distinct_count("destination_ip") over "destination_ip"	192.168.0.13	destination_ip: 192.168.0.13	4	1	5x higher	
May 31st 2022, 03:00	84	count over "destination_ip"	192.168.1.1	destination_ip: 192.168.1.1	19110	82.4	More than 100x higher	
May 31st 2022, 10:00	82	count over "destination_ip"	224.0.0.252	destination_ip: 224.0.0.252	12833	9.59	More than 100x higher	
May 31st 2022, 10:00	72	count over "destination_ip"	192.168.1.1	destination_ip: 192.168.1.1	6179	9.59	More than 100x higher	
May 31st 2022, 11:00	69	count over "destination_ip"	192.168.1.1	destination_ip: 192.168.1.1	5611	9.59	More than 100x higher	
May 31st 2022, 03:00	56	distinct_count("destination_ip") over "destination_ip"	192.168.0.23	destination_ip: 192.168.0.23	4	1	4x higher	
May 31st 2022, 04:00	50	distinct_count("destination_ip") over "destination_ip"	192.168.0.19	destination_ip: 192.168.0.19	4	1	4x higher	
May 31st 2022, 03:00	30	count over "destination_ip"	224.0.0.252	destination_ip: 224.0.0.252	6303	82.4	76x higher	
May 31st 2022, 04:00	29	count over "destination_ip"	192.168.1.1	destination_ip: 192.168.1.1	2325	19	More than 100x higher	
May 31st 2022, 10:00	22	count over "destination_ip"	192.168.0.255	destination_ip: 192.168.0.255	1050	9.59	More than 100x higher	
May 31st 2022, 04:00	21	count over "destination_ip"	34.126.127.130	destination_ip: 34.126.127.130	1725	19	91x higher	
May 31st 2022, 10:00	12	count over "destination_ip"	224.0.0.251	destination_ip: 224.0.0.251	211	9.59	74x higher	
May 31st 2022, 10:00	8	count over "destination_ip"	R02-ib	destination_ip: R02-ib	639	9.59	67x higher	
May 31st 2022, 04:00	6	count over "destination_ip"	224.0.0.251	destination_ip: 224.0.0.251	1014	19	53x higher	
May 31st 2022, 04:00	6	count over "destination_ip"	R02-ib	destination_ip: R02-ib	933	19	49x higher	
May 31st 2022, 10:00	5	distinct_count("destination_ip") over "destination_ip"	10.0.2.2	destination_ip: 10.0.2.2	3	1	3x higher	

Εικόνα 18: Αποτελέσματα για flood anomaly.

5 Επίλογος

Από όσα εκθέσαμε παραπάνω, είναι εμφανές ότι ο συνδυασμός του Zeek ως εργαλείο network monitoring με το ELK stack δίνουν εκτεταμένες δυνατότητες συλλογής, ανάλυσης και οπτικοποίησης δεδομένων. Ιδιαίτερα στον τομέα της ασφάλειας, η πληθώρα επιλογών παραμετροποίησης του ELK stack, καθώς και η ύπαρξη εξειδικευμένων εργαλείων για ανίχνευση απειλών δίνει σαφές πλεονέκτημα σε ομάδες ασφαλείας που έχουν ως στόχο την αποτροπή κακόβουλων ενεργειών εις βάρος κάποιου οργανισμού.

6 Βιβλιογραφία

- [1] Tsai, Chih-Fong & Hsu, Yu-Feng & Lin, Chia-Ying & Lin, Wei-Yang. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*. 36. 11994-12000. [10.1016/j.eswa.2009.05.029](https://doi.org/10.1016/j.eswa.2009.05.029).
- [2] Sarker, Iqbal & Abushark, Yoosef & Alsolami, Fawaz & Khan, Asif. (2020). IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model. *Symmetry*. 12. [10.3390/sym12050754](https://doi.org/10.3390/sym12050754).
- [3] Amit, Idan & Matherly, John & Hewlett, William & Xu, Zhi & Meshi, Yinnon & Weinberger, Yigal. (2019). Machine Learning in Cyber-Security -Problems, Challenges and Data Sets.
- [4] Hyunjae Kang, Dong Hyun Ahn, Gyung Min Lee, Jeong Do Yoo, Kyung Ho Park, Huy Kang Kim, September 27, 2019, "IoT network intrusion dataset", IEEE Dataport, doi: <https://dx.doi.org/10.21227/q70p-q449>.