ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ευφυές σύστημα διαχείρισης πληροφοριών και συμβάντων ασφαλείας (SIEM) με τη χρήση της στοίβας υπηρεσιών ELK

Του φοιτητή
Μουζενίδη Γιάννη
Αρ. Μητρώου: 144267

Επιβλέπων
Ηλιούδης Χρήστος
Καθηγητής

Ημερομηνία 12/01/2022

Τίτλος Δ.Ε. Ευφυές σύστημα διαχείρισης πληροφοριών και συμβάντων ασφαλείας (SIEM) με τη χρήση της στοίβας υπηρεσιών ELK

Κωδικός Δ.Ε. 21209

Ονοματεπώνυμο φοιτητή Μουζενίδης Γιάννης

Ονοματεπώνυμο εισηγητή Ηλιούδης Χρήστος

Ημερομηνία ανάληψης Δ.Ε. 21/03/2021

Ημερομηνία περάτωσης Δ.Ε. 16/01/2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Ο βασικότερος λόγος επιλογής της συγκεκριμένης εργασία είναι το προσωπικό ενδιαφέρον για έρευνα και μελέτη του επιστημονικού πεδίου της ασφάλειας υπολογιστών. Εκτός από αυτό βέβαια, η συγκεκριμένη εργασία καλύπτει σε μεγάλο βαθμό ένα μεγάλο σύνολο τεχνολογιών και προτύπων στο χώρο της ασφάλειας υπολογιστών που χρησιμοποιούνται εκτενώς στην σύγχρονη αγορά εργασίας από οργανισμούς πληροφορικής και όχι μόνο. Κατά τη διαδικασία της εκπόνησης υπήρξε μελέτη σε βάθος πολλών από αυτές τις τεχνολογίες κάτι που επέφερε επιπλέον γνώσεις πάνω στο αντικείμενο.

Περίληψη

Πλέον στη σύγχρονη κοινωνία, όλο και περισσότερες καθημερινές υπηρεσίες, αξιοποιούν σε μεγάλο βαθμό ψηφιακά δεδομένα τα οποία διαχειρίζονται σύνθετα υπολογιστικά συστήματα. Καθώς πολλές από αυτές τις υπηρεσίες έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας, είναι σημαντική η διασφάλιση της σωστής και ομαλής λειτουργίας τους. Η συγκεκριμένη Π.Ε αναλύει τους κινδύνους και κάποιους από τους μηχανισμούς τις τεχνολογίες και τα πρότυπα που χρησιμοποιούνται ευρέως για να συμβάλουν στην ασφάλεια των συστημάτων και των υπηρεσιών αυτών. Μεγαλύτερη έμφαση δίνεται στα συστήματα SIEM, οπού αναλύεται η αρχιτεκτονική και η λειτουργία τους ενώ αξιολογούνται κάποια από τα πιο γνωστά συστήματα SIEM στον τομέα της ασφάλειας. Τέλος αναλύεται σε βάθος η στοίβα υπηρεσιών ELK και η διαδικασία με την οποία η στοίβα αυτή μπορεί να χρησιμοποιηθεί ως βάση για την υλοποίηση ενός ολοκληρωμένου συστήματος SIEM, το οποίο αξιολογείται με βάση συγκεκριμένες περιπτώσεις χρήσης.

Intelligent Security Information and Event Management by using ELK Stack

Giannis Mouzenidis

Abstract

In the modern society, most of everyday services depend on digital data and the computer systems that are used to handle it. Since most of these services became irreplaceable part of the everyday life, it is essential to ensure their functionality. The current paper, analyzes some of the dangers that threatening these systems, and some of the mechanisms and standards that are used in order to secure them. The security mechanisms that are mostly analyzed are the SIEM tools and their functionality and architecture. Also, there is comparison of the most known SIEM tools that are used in the industry. Finally, this paper examines in depth ELK stack and how it can be used as a strong base for a complete SIEM tool, which is later on tested against specific use cases.

Ευχαριστίες

Ευχαριστώ την οικογένεια μου και τους κοντινούς μου ανθρώπους για τη στήριξη τους καθ’ όλη τη διάρκεια της εκπόνησης της Π.Ε αλλά και σε πολλές άλλες περιστάσεις της ζωής μου. Επίσης, ευχαριστώ τον καθηγητή και επιβλέποντα της εργασίας Ηλιούδη Χρήστο για την καθοδήγηση που μου πρόσφερε σχετικά με ολόκληρη τη διαδικασία της εκπόνησης της Π.Ε, καθώς και για τις γνώσεις που μου πρόσφερε ως καθηγητής στα μαθήματα του, μέσω των οποίων ανακάλυψα το ενδιαφέρον μου για τον τομέα της ασφάλειας υπολογιστών, ο οποίος εξελίχθηκε μετέπειτα στον επαγγελματικό μου προσανατολισμό.

Περιεχόμενα

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
SIEM	Security Information Event Management
ELK	Elasticsearch Logstash Kibana

Κεφάλαιο 1ο: Εισαγωγή

1.1 Επιστημονικό πεδίο ασφάλειας υπολογιστών

Με την ραγδαία ανάπτυξη στην επιστήμη των υπολογιστών, τις τελευταίες 3 δεκαετίες, δημιουργήθηκαν νέες απαιτήσεις και ανάγκες. Αυξήθηκε σημαντικά η πολυπλοκότητα των πληροφοριακών και των υπολογιστικών συστημάτων ανά το κόσμο. Υπηρεσίες και δεδομένα, που συμβάλουν σημαντικά στην ομαλή λειτουργία της σημερινής κοινωνίας, από φυσική μορφή που είχαν παλαιότερα, πλέον αναπαρίστανται σε ψηφιακή μορφή, χάρη στην ανάπτυξη της επιστήμης των υπολογιστών. Πολλές από αυτές τις αλλαγές, έγιναν με κύριο στόχο, να διευκολύνουν και να βελτιώσουν την καθημερινότητα των ανθρώπων.

Ταυτόχρονα, με την ανάπτυξη αυτή όμως, δημιουργήθηκαν νέες απαιτήσεις και ανάγκες [16]. Μια από τις σημαντικότερες ανάγκες που δημιουργήθηκαν, ήταν αυτή, της διασφάλισης της ομαλής λειτουργίας των πληροφοριακών υπηρεσιών και συστημάτων. Μια από τις πρώτες και σημαντικότερες απειλές για την ομαλή λειτουργία των συστημάτων αυτών ήταν οι κακόβουλοι χρήστες. Κακόβουλοι χρήστες από τις αρχές εμφάνισης πληροφοριακών συστημάτων προσπαθούσαν να προκαλέσουν φθορές ηλεκτρονικά πλέον, παραβιάζοντας τις υπηρεσίες αυτές με σκοπό να τις αχρηστεύουν.

Κάποια από τα συχνότερα κίνητρα των κακόβουλων χρηστών είναι:

- Χρηματικό όφελος. Είτε επειδή λάμβαναν χρηματικό κέρδος από κάποιον τρίτο για να προκαλέσουν φθορές, είτε γιατί η υπηρεσία που προσπαθούσαν να βλάψουν ήταν κάποιου ανταγωνιστή και έτσι θα είχαν έμμεσο χρηματικό κέρδος, είτε για να κερδίσουν χρήματα αξιοποιώντας δεδομένα που έκλεψαν παραβιάζοντας το σύστημα

- Προσωπική εκδίκηση. Πολλές φορές ο μόνος λόγος που χρειαζόταν ένας χρήστης για να προκαλέσει κακόβουλες ενέργειες σε ένα σύστημα ήταν η προσωπική εκδίκηση για δικούς του λόγους απέναντι στον ιδιοκτήτη συνήθως, του οργανισμού στον οποίον ανήκει μια υπηρεσία.

- Λόγοι ηθικής. Ορισμένοι κακόβουλοι χρήστες υποκινούνται από συναισθήματα ηθικής, πιστεύοντας ότι με τέτοιο τρόπο θα υποστηρίξουν τις προσωπικές τους απόψεις και πεποιθήσεις.

Έτσι δημιουργήθηκε μία νέα ανάγκη για την διασφάλιση της ομαλής λειτουργίας των υπολογιστικών συστημάτων. Το γεγονός αυτό οδήγησε και στη δημιουργία του τομέα της ασφάλειας των υπολογιστών για να καλύψει αυτή την ανάγκη και να δώσει τις λύσεις που απαιτούνται.

Τα δεδομένα, που πλέον στην πλειοψηφία τους αναπαρίστανται σε ψηφιακή μορφή, αποτελούν σημαντικότατο και αναπόσπαστο κομμάτι της σημερινής κοινωνίας, γεγονός που έχει ως συνέπεια να δίνεται μεγάλη βαρύτητα στην ομαλή λειτουργεία των πληροφοριακών συστημάτων που

επεξεργάζονται, αποθηκεύουν και χειρίζονται τον τεράστιο αυτό όγκο δεδομένων που υπάρχει. Στον τομέα της ασφάλειας αν και με τον καιρό υπήρξαν πολλές διαφορετικές εννοείς και πρωτόκολλα σχετικά με την διασφάλιση των συστημάτων και των υπηρεσιών, μια από τις σημαντικότερες και επικρατέστερες έννοιες είναι το γνωστό τρίπτυχο CIA (Confidentiality, Integrity, Availability). Το μοντέλο αυτό περιγράφει την ασφάλεια ενός συστήματος, ως ένα σύνολο που αποτελείται από τις τρείς αυτές επιμέρους έννοιες.

Το μοντέλο αυτό περιγράφει την ασφάλεια ενός συστήματος, ως ένα σύνολο που αποτελείται από τις τρείς αυτές επιμέρους έννοιες :

- Availability (Διαθεσιμότητα): Περιγράφει την διασφάλιση της διαθεσιμότητας ενός συστήματος ανά πάσα στιγμή. Πολλά συστήματα λόγω της υπηρεσίας που προσφέρουν, πρέπει να είναι διαθέσιμα συνέχεια, ανά μεγάλα χρονικά διαστήματα.

- Integrity (Ακεραιότητα): Περιγράφει την διασφάλιση πως τα δεδομένα που χρησιμοποιούνται από ένα σύστημα δεν έχουν αλλαχθεί χωρίς την απαραίτητη άδεια.

- Confidentiality (Εμπιστευτικότητα): Περιγράφει την διασφάλιση πως πρόσβαση στα δεδομένα θα πρέπει να έχουν μόνο συγκεκριμένες οντότητες.

Πολλά μοντέλα και πρωτόκολλα ασφαλείας που υλοποιηθήκαν από διάφορους οργανισμούς, με στόχο την διασφάλιση των υπηρεσιών τους, βασίστηκαν πάνω στο μοντέλο CIA. Είναι σημαντικό να αναφερθεί πως κάθε έννοια του μοντέλου αυτού παρουσιάζει ξεχωριστές απειλές και τρόπους με τους οποίους μπορεί να παραβιαστεί το μοντέλο αυτό. Όταν πρώτο-άρχισαν να εφαρμόζονται διάφορα πρωτόκολλα ασφαλείας, οι οργανισμοί ανάθεσαν την επίβλεψη των συστημάτων τους με βάση τα πρωτόκολλα αυτά, σε εργαζόμενους ανθρώπους. Σύντομα όμως άρχισαν να εμφανίζονται αρκετά προβλήματα σε αυτή προσέγγιση. Αρχικά τα περισσότερα συστήματα παρέχουν άμεσα ή έμμεσα υπηρεσίες σε πραγματικό χρόνο, για μεγάλα χρονικά διαστήματα, κάτι που σημαίνει πως απαιτούν 24ώρη επίβλεψη από κάποιον άνθρωπο. Αυτό για πολλούς οργανισμούς αποτελούσε πρόβλημα καθώς έτσι αυξανόταν σημαντικά το κόστος για την λειτουργία και την συντήρηση του πληροφοριακού του συστήματος. Αυτό όμως δεν ήταν το μόνο πρόβλημα. Με την αύξηση της πολυπλοκότητας των συστημάτων, αυξήθηκαν και οι απειλές, γεγονός που απαιτούσε με την σειρά του ιδιαίτερα εξειδικευμένο προσωπικό. Ακόμα και έτσι όμως, η επίβλεψη ενός τόσο σύνθετου και πολύπλοκου συστήματος ήταν μία πολύ δύσκολη διαδικασία.

Οι δυσκολίες αυτές, δημιούργησαν την ανάγκη για τη δημιουργία κάποιου μηχανισμού, ο οποίος θα επιβλέπει το σύστημα, ελαχιστοποιώντας την ανθρώπινη συμβολή όσο το δυνατόν περισσότερο. Υπήρξαν πολλά μεταβατικά στάδια και πολλοί μηχανισμοί που προσπαθούσαν να δώσουν λύση στο πρόβλημα ή έστω να τη προσεγγίσουν. Ένας από τους μηχανισμούς που υλοποιήθηκε και πλέον χρησιμοποιείται από πολλούς φορείς είναι ο μηχανισμός των πληροφοριών ασφάλειας και διαχείρισης συμβάντων ή όπως είναι γνωστός, SIEM (Security Information and Event Management). Ο μηχανισμός αυτός [2] επιβλέπει ένα σύστημα, καταγράφει συμβάντα, και ενημερώνει ανθρώπους αν κρίνει πως είναι απαραίτητο, με βάση τους κανόνες με τους οποίους λειτουργεί, και υλοποιεί πολλές ακόμα πιο σύνθετες

διαδικασίες και ενέργειες. Στα επόμενα κεφάλαια ο μηχανισμός αυτός θα είναι το επίκεντρο της μελέτης.

1.2 Δομή της πτυχιακής εργασίας

Η συγκεκριμένη εργασία ακολουθεί την εξής δομή. Το πρώτο κεφάλαιο είναι εισαγωγικό και περιγράφει γενικά κάποιες από τις έννοιες που θα αναλυθούν στα επιμέρους κεφάλαια. Το δεύτερο κεφάλαιο αναλύει σε βάθος την δομή των SIEM, τα πρότυπα που στηρίζουν την δια λειτουργική επικοινωνία του SIEM σε ένα οικοσύστημα μηχανισμών και υπηρεσιών ασφάλειας, τα συστατικά που τα απαρτίζουν και το πως αυτά λειτουργούν ως σύνολο για να πετύχουν το επιθυμητό αποτέλεσμα. Το τρίτο κεφάλαιο καταγράφει και αναλύει κάποια από τα γνωστά συστήματα SIEM ανοιχτού κώδικα που χρησιμοποιούνται από διάφορους οργανισμούς, και κατόπιν τα αξιολογεί και τα συγκρίνει μεταξύ τους ανάλογα με συγκεκριμένες παραμέτρους. Το τέταρτο κεφάλαιο αναλύει και περιγράφει την στοίβα υπηρεσιών ELK και τις υπηρεσίες που την απαρτίζουν και μετέπειτα καταγράφει τους λόγους για τους οποίους μπορεί να χρησιμοποιηθεί η συγκεκριμένη στοίβα στην υλοποίηση ενός συστήματος SIEM. Το πέμπτο κεφάλαιο αναλύει την διαδικασία υλοποίησης ενός συστήματος SIEM με την χρήση της στοίβας υπηρεσιών ELK, προσθέτοντας ορισμένες λειτουργίες. Το έκτο κεφάλαιο προσομοιώνει κάποιες περιπτώσεις χρήσης του υλοποιημένου, στο πέμπτο κεφάλαιο, συστήματος και το αξιολογεί με βάση τα αποτελέσματα. Τέλος το έβδομο κεφάλαιο περιλαμβάνει τα συμπεράσματα μετά το πέρας της εργασίας καθώς και πιθανές μελλοντικές βελτιώσεις και αλλαγές στους μηχανισμούς SIEM.

1.3 Στόχοι της πτυχιακής εργασίας

Η συγκεκριμένη εργασία στοχεύει στην μελέτη και την ανάλυση της λειτουργίας των συστημάτων SIEM,των επιμέρους μηχανισμών τους και των προτύπων τους. Επίσης στοχεύει στην έρευνα και τη σύγκριση γνωστών συστημάτων που χρησιμοποιούνται ως SIEM από πολλούς οργανισμούς και την αξιολόγηση των συστημάτων αυτών. Μετά τη μελέτη διάφορων συστημάτων ο στόχος είναι η αναλυτική έρευνα της λειτουργίας και αρχιτεκτονικής της στοίβας ELK που αποτελεί βάση για πολλούς μηχανισμούς SIEM, καθώς και η μελέτη των υπηρεσιών της. Ο τελικός στόχος είναι η υλοποίηση ενός συστήματος SIEM με τη χρήση της στοίβας ELK ως βάση του, η παραμετροποίηση του και η προσθήκη επιπλέον λειτουργιών, καθώς και η αξιολόγησή του με βάση συγκεκριμένες περιπτώσεις χρήσης αλλά και άλλες παραμέτρους όπως η εφαρμοσημότητα του και ευκολία υλοποίησης. Επιπλέον στοχεύει στην υλοποίηση, παραμετροποίηση και την αξιολόγηση ενός συστήματος SIEM με την χρήση της στοίβας υπηρεσιών ELK, καθώς και τη δοκιμή του σε συγκεκριμένες περιπτώσεις χρήσης.

Κεφάλαιο 2ο: Συστήματα SIEM και τεχνολογίες στο χώρο της ασφάλειας

2.1 Εισαγωγή

Τα συστήματα SIEM αποτελούν ένα από τα σημαντικότερα εργαλεία που χρησιμοποιούν πολλές εταιρίες με στόχο να διασφαλίσουν την ομαλή λειτουργία των πληροφοριακών τους συστημάτων. Τα συστήματα αυτά επιβλέπουν, καταγράφουν και ενημερώνουν σχετικά με πιθανές απειλές. [5] Είναι σημαντικό να αναφερθεί, πως τα συστήματα αυτά αποτελούν μέρος ενός μεγαλύτερου οικοσυστήματος υπηρεσιών και μηχανισμών ασφαλείας ενός οργανισμού. Στο κεφάλαιο αυτό θα αναλυθεί η δομή και η λειτουργία των SIEM, τα χαρακτηριστικά που θα πρέπει να έχει ένα τέτοιο σύστημα καθώς και τα πρότυπα που στηρίζουν την διαλειτουργική αυτή επικοινωνία των SIEM με τις υπόλοιπες υπηρεσίες και μηχανισμούς του οικοσυστήματος το οποίο απαρτίζουν.

2.2 Συστήματα SIM και SEM

Λόγω των απαιτήσεων που καλούνται να καλύψουν τα συστήματα SIEM, αποτελούν ένα λογισμικό αρκετά πολύπλοκο και σύνθετο. Εκτελούν πολλές λειτουργίες διάφορου είδους. Σε συγκεκριμένες περιπτώσεις κάποιες από τις επιμέρους υπηρεσίες τους, μπορούν να χρησιμοποιηθούν ακόμη και ως αυτόνομα συστήματα με λιγότερες βέβαια δυνατότητες [6]. Πριν αναλυθούν όλες οι λειτουργίες που εκτελούν τα SIEM, θα γίνει η αρχή από τις δύο ίσως βασικότερες, οι οποίες σε πολλές περιπτώσεις λειτουργούν και ως αυτόνομα συστήματα. Τα συστήματα αυτά λοιπόν, είναι γνωστά ως SIM (Security Information Management) και SEM (Security Event Management). Τα SIEM με την σειρά τους αξιοποιούν τις λειτουργίες από τα προαναφερθέντα συστήματα SIM και SEM σε συνδυασμό με κάποιες ακόμα λειτουργίες που θα αναλυθούν αργότερα στο τρέχον κεφάλαιο. Τα SIEM χαρακτηρίζονται συχνά ως μια σύμπτυξη των συστημάτων SIM και SEM

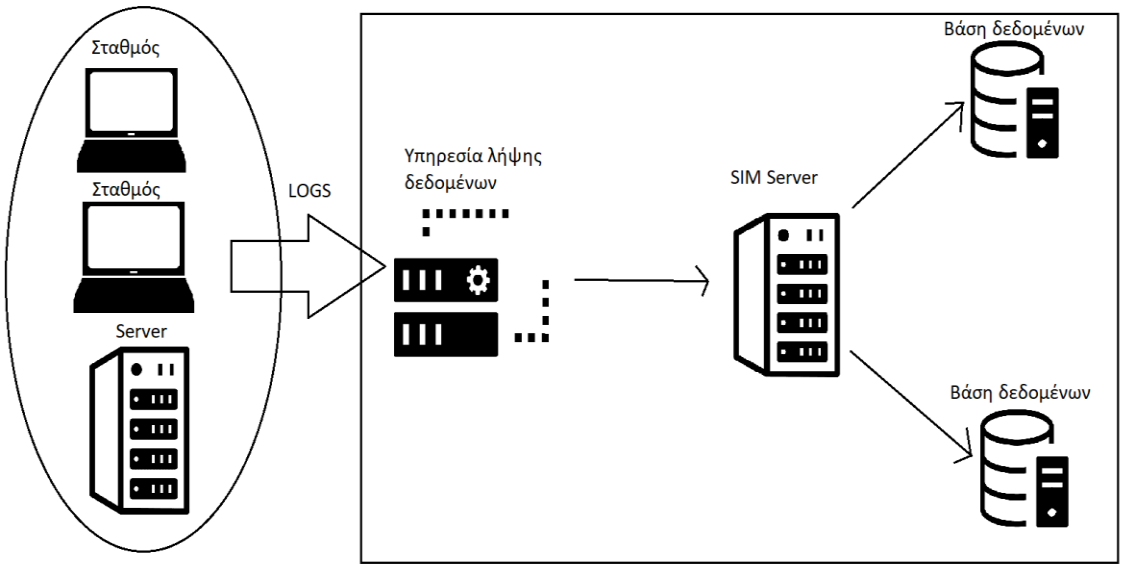
2.2.1 SIM (Security Information Management)

Ο βασικός στόχος των συστημάτων SIM [6] είναι η συλλογή και η διατήρηση δεδομένων (logs) που σχετίζονται με την ασφάλεια του συστήματος. Τα δεδομένα αυτά προέρχονται κυρίως από άλλα συστήματα ή μηχανισμούς όπως τείχη προστασίας (Firewalls), διακομιστές μεσολάβησης (proxy servers), προγράμματα antivirus και άλλους παρόμοιους μηχανισμούς. Ένα ερώτημα που τίθεται είναι η μορφή των δεδομένων από όλες αυτές τις διαφορετικές πηγές. Τη λύση στο πρόβλημα δίνουν διάφοροι μηχανισμοί φιλτραρίσματος ή κανονικοποίησης. Οι μηχανισμοί αυτοί μετατρέπουν δεδομένα από διαφορετικές πηγές, διαφορετικής μορφής σε δεδομένα κοινής δομής που μπορούν αν αξιοποιηθούν ευκολότερα. Είναι σημαντικό να αναφερθεί, πως το χρονικό διάστημα της διατήρησης των δεδομένων μπορεί να είναι από βδομάδες έως και μήνες, ανάλογα με τις ανάγκες και τα πρωτόκολλα της εκάστοτε

εταιρίας. Η διατήρηση των δεδομένων για μεγάλα χρονικά διαστήματα προσθέτει έμμεσα μία ακόμη ευθύνη στα συστήματα SIM. Η ευθύνη αυτή είναι η ασφαλής διατήρηση των δεδομένων που συλλέγονται. Ο κύριος λόγος που πραγματοποιείται η συλλογή και η διατήρηση των δεδομένων, είναι η μελλοντική τους ανάλυση.

Η συλλογή των δεδομένων πραγματοποιείται από υπηρεσίες που ρυθμίζονται σχετικά με το από ποια αρχεία να αντλήσουν τα απαραίτητα δεδομένα (logs). Τα logs συνήθως αποθηκεύονται σε συγκεκριμένα αρχεία ανάλογα με το είδος τους. Για παράδειγμα ο Apache2 HTTP server, σε λειτουργικά συστήματα Linux βασισμένα σε Debian Linux, κρατάει αυτές τις πληροφορίες στο αρχείο /var/log/apache2/error.log. Ομοίως διαφορετικά συστήματα και υπηρεσίες έχουν αντίστοιχα αρχεία, από τα οποία το SIM αντλεί πληροφορίες για να τις αναλύσει

Η ανάλυση πραγματοποιείται συνήθως είτε από άνθρωπο, είτε από το ίδιο SIM εφόσον υπάρχει αυτή η δυνατότητα. Συνήθως γίνεται ο συνδυασμός πολλαπλών αναλύσεων και η διασταύρωση με στόχο την βελτιστοποίηση των αποτελεσμάτων.



Σχήμα 2.1. Παράδειγμα αρχιτεκτονική συστήματος SIM

Πολλές έρευνες [4][6], προσπάθησαν να περιγράψουν τη πλήρη διαδικασία της λειτουργίας συστημάτων αποθήκευσης και ανάλυσης δεδομένων ασφαλείας καθώς και το τρόπο με τον οποίο μπορεί να αξιοποιηθεί από έναν οργανισμό.

Σε μια γενική προσέγγιση η διαδικασία μπορεί να διαχωριστεί σε πέντε (5) στάδια:

1. Ρύθμιση μηχανισμών συλλογής και διατήρησης δεδομένων. Σε αυτό το στάδιο οι διαχειριστές του συστήματος πρέπει να ρυθμίσουν διάφορες παραμέτρους του συστήματος SIM. Συγκεκριμένα οι παράμετροι που χρήζουν ρυθμίσεων πριν την λειτουργία και την συλλογή δεδομένων είναι οι εξής:

    - Ρύθμιση παραμέτρων σχετικών με την συλλογή των δεδομένων. Παραδείγματα τέτοιων παραμέτρων αποτελούν, τα συστήματα που θα λειτουργήσουν ως πηγές δεδομένων ή ο ρυθμός καταγραφής των δεδομένων, όπου η κακή ρύθμιση τους μπορεί να προκαλέσει προβλήματα στο σύστημα όπως να το αφήσουν ευάλωτο σε επιθέσεις άρνησης υπηρεσιών (DoS) αν ο ρυθμός καταγραφής είναι λανθασμένα μεγάλος.

    - Ρυθμίσεις αποθήκευσης και διατήρησης των δεδομένων. Οι ρυθμίσεις αυτές καθορίζονται σε μεγάλο βαθμό από την πολιτική και τα πρωτόκολλα της κάθε εταιρίας. Έτσι, κάποια δεδομένα διατηρούνται με δικαιώματα πρόσβασης σε όλους τους χρήστες, κάποια με δικαιώματα μόνο για τους διαχειριστές, ενώ κάποια άλλα δεν διατηρούνται καθόλου.

    - Ρυθμίσεις ασφάλειας των δεδομένων. Εδώ η διαχείριση δικαιωμάτων σε αντίθεση με τις ρυθμίσεις διατήρησης, αφορά τα δικαιώματα στην χρήση και την ρύθμιση του συστήματος και των μηχανισμών που χρησιμοποιεί. Επίσης η αποθήκευση των δεδομένων σε εξωτερικές κεντροποιημένες βάσεις δεδομένων είναι ακόμα μια χρήσιμη λειτουργία που βελτιώνει την ασφάλεια του συστήματος. Τέλος, η χρήση πρωτοκόλλων και μηχανισμών που διασφαλίζουν την ασφαλή μετάδοση των δεδομένων από και προς το σύστημα SIM όπως για παράδειγμα το IPsec και το SSL, είναι επίσης καλή πρακτική.

2. Ανάλυση των αποθηκευμένων δεδομένων. Σε αυτό το στάδιο πραγματοποιείται η ανάλυση των δεδομένων. Οι σημαντικότερες ενέργειες που συμβάλλουν στην αποτελεσματικότερη ανάλυση των δεδομένων είναι οι εξής:

    - Κατανόηση των δεδομένων. Πολλές φορές τα δεδομένα συνδέονται μεταξύ τους και αλληλοσυμπληρώνονται. Αυτό σημαίνει πως για να είναι αποτελεσματική η ανάλυση τους, ο διαχειριστής θα πρέπει να αντιληφθεί την σύνδεση και να αναλύσει τα δεδομένα συνολικά και όχι μεμονωμένα. Ένα ακόμα πρόβλημα που μπορεί να προκύψει είναι η διαφορετική μορφή των δεδομένων. Όπως είχε αναφερθεί προηγουμένως, τα συστήματα SIM συλλέγουν δεδομένα από πολλές διαφορετικές πηγές. Επίσης τα συστήματα SIM σε κάποιες περιπτώσεις συμπεριλαμβάνουν και τα ίδια μηχανισμούς ανάλυσης. Αυτό μπορεί να οδηγήσει σε περιπτώσεις όπου τα δεδομένα είναι κατανοητά από το σύστημα SIM αλλά δεν είναι κατανοητά για τον διαχειριστή του συστήματος, ειδικά όταν πρόκειται για δεδομένα που δεν έχουν καταγραφεί στο παρελθόν.

- Ιεράρχηση προτεραιοτήτων δεδομένων. Κάθε εταιρία ανάλογα με τις ανάγκες της και την πολιτική της, θέτει διαφορετική προτεραιότητα σε κάθε καταγραφή δεδομένων. Έτσι κάποιες έγγραφες μπορούν να θεωρηθούν πολύ σημαντικές ενώ άλλες ασήμαντες. Παράγοντες που συνήθως καθορίζουν την σημαντικότητα είναι η ώρα και η μέρα καταγραφής, η πηγή της καταγραφής και η IP διεύθυνση της που μπορεί να είναι σε κάποια λίστα με κακόβουλες διευθύνσεις, το αν τα δεδομένα έχουν καταγραφεί στο παρελθόν και άλλοι παράγοντες ανάλογα με τα πρωτόκολλα της εταιρίας. Οι παράγοντες αυτοί λοιπόν, δημιουργούν μια ιεραρχία στα δεδομένα με βάση την προτεραιότητα τους. Με αυτόν τον τρόπο κατά την ανάλυση, δίνεται μεγαλύτερή βάση στα ιεραρχικά ψιλότερα δεδομένα.

3. Λήψη ενεργειών σύμφωνα με τα αποτελέσματα της ανάλυσης. Αφού έρθει εις πέρας η ανάλυση των καταγεγραμμένων δεδομένων ο διαχειριστής του συστήματος καλείται να λάβει κάποιες αποφάσεις και ενέργειες ανάλογα με τα αποτελέσματα της ανάλυσης αυτής. Οι ενέργειες στις οποίες μπορεί να προβεί ένας διαχειριστής ώστε να αυξήσει την ασφάλεια των μηχανισμών της εταιρίας, από πιθανές απειλές που έδειξε η ανάλυση, διαχωρίζονται σε ενέργειες επιπέδου υποδομής, και ενέργειες επιπέδου συστήματος. Οι ενέργειες επιπέδου υποδομής αφορούν κυρίως αλλαγές ρυθμίσεων σε συσκευές που συνδέουν το δίκτυο του οργανισμού με τα εξωτερικά δίκτυα, τέτοιες συσκευές είναι για παράδειγμα οι δρομολογητές και τα τείχη προστασίας (Firewalls). Από την άλλη, οι ενέργειες σε επίπεδο συστήματος αφορούν κυρίως ρυθμίσεις εσωτερικού δικτύου, λογισμικού χρηστών της εταιρίας, και λειτουργικών συστημάτων.

4. Μακροπρόθεσμη διαχείριση των δεδομένων. Όπως είχε αναφερθεί σε προηγούμενη ενότητα, τα SIM διατηρούν τα δεδομένα για μεγάλα χρονικά διαστήματα, μέχρι να πραγματοποιηθεί η ανάλυση από κάποιον διαχειριστή ή από το σύστημά εφόσον υπάρχει αυτή η δυνατότητα. Ένα ερώτημα που προκύπτει μετά την ανάλυση των δεδομένων είναι, το τι θα γίνει με τα δεδομένα αυτά μετά την ανάλυση. Σε αυτό το ερώτημα η απάντηση εξαρτάται κυρίως από τις ανάγκες της εταιρίας και προφανώς από την σημαντικότητα τους. Σε πολλές περιπτώσεις μετά την ανάλυση οι διαχειριστές αποθηκεύουν τα δεδομένα σε εξωτερικές βάσεις δεδομένων για ακόμη μεγαλύτερα διαστήματα. Άλλοι τρόποι αποθήκευσης είναι διάφορα φυσικά μέσα όπως CDs, DVDs, σκληροί δίσκοι HDD. Επίσης είναι σημαντικό να αναφερθεί πως ο διαχειριστής θα πρέπει να επιλέξει μια συγκεκριμένη μορφή για την αποθήκευση των δεδομένων που θα πραγματοποιείται ανά διαστήματα. Η ασφαλής αποθήκευση των καταγεγραμμένων δεδομένων είναι επίσης ευθύνη του διαχειριστή, καθώς η απώλεια ή η μη εξουσιοδοτημένη τροποποίηση των δεδομένων μπορούν να επιφέρουν σοβαρά προβλήματα.

5. Υποστήριξη και συντήρηση των μηχανισμών καταγραφής. Είναι σημαντικό ο διαχειριστής να ελέγχει την σωστή λειτουργία όσο του συστήματος SIM όσο και των υπόλοιπων μηχανισμών που παρέχουν πληροφορίες στο SIM. Ενέργειες που συμβάλουν στην ομαλή λειτουργία του οικοσυστήματος είναι:

- Τακτικός έλεγχος για ενημερώσεις στο λογισμικό των μηχανισμών του οικοσυστήματος της εταιρίας. Είναι γνωστό πως η ενημέρωση των συστημάτων είναι ζωτικής σημασίας, όσο για την ασφάλεια ενός λογισμικού τόσο και για την ομαλή λειτουργία σε ένα ραγδαία εξελισσόμενο περιβάλλον.

- Αλλαγή στις διάφορες ρυθμίσεις των μηχανισμών ανάλογα με τις αλλαγές στην πολιτική και τα πρωτόκολλα της εταιρίας. Πολλές φορές οι εταιρίες αλλάζουν διάφορες πολιτικές σχετικές με δεδομένα και άλλες λειτουργίες. Τα συστήματα που χρησιμοποιούν θα πρέπει να συμβαδίζουν με τις πολιτικές αυτές.

- Τροποποίηση μηχανισμών που παράγουν μη επιθυμητά αποτελέσματα δεδομένων. Συχνά η ανάλυση δεδομένων αποκαλύπτει δεδομένα τα οποία είναι κατεστραμμένα. Αυτό μπορεί να οφείλεται είτε από κάποια λάθος ρύθμιση σε κάποιον μηχανισμό, είτε από κάποιο κακόβουλο λογισμικό. Και στις δυο περιπτώσεις η ανάλυση θα αποκαλύψει το πρόβλημα και ο διαχειριστής θα πρέπει να το λύσει με συγκεκριμένο τρόπο ανάλογα με την περίπτωση.

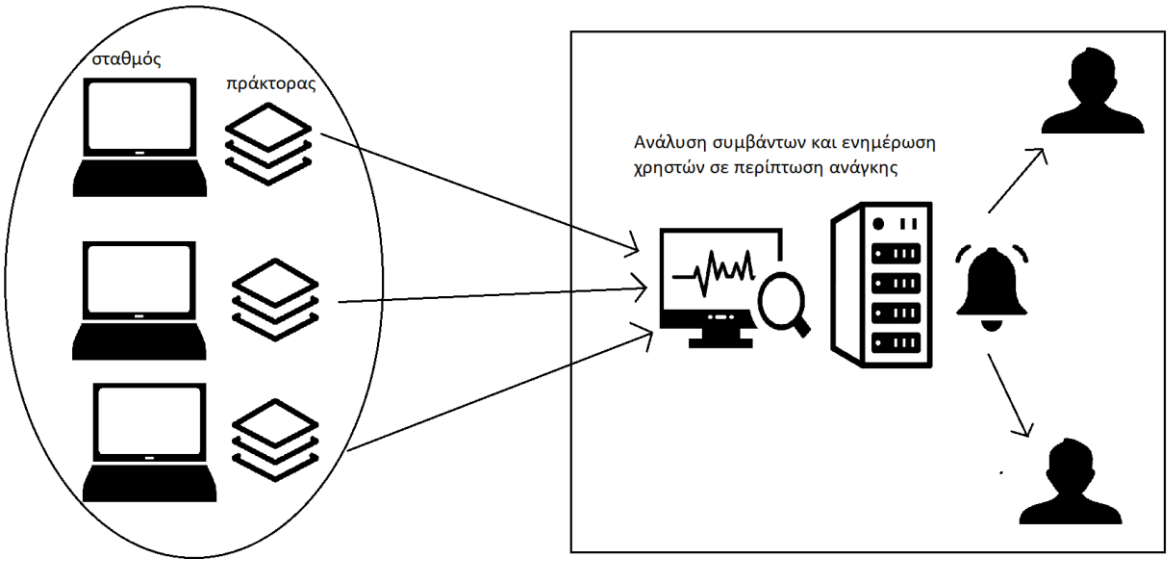
2.2.2 SEM (Security Event Management)

Σε αντίθεση με τα συστήματα SIM, τα συστήματα SEM (Security Event Management) [5], έχουν ως βασική υποχρέωση την επίβλεψη των υποδομών και των συστημάτων του οργανισμού σε πραγματικό χρόνο. Αν και οι μηχανισμοί SEM έχουν και αυτοί την δυνατότητα αποθήκευσης δεδομένων σχετικών με περιστατικά ασφάλειας, υστερούν σε δυνατότητες ανάλυσης των δεδομένων αυτών καθώς και του όγκου αποθήκευσης δεδομένων. Οι βασικές λειτουργίες ενός SEM συστήματος που το διαχωρίζουν από τα συστήματα SIM [5][6] είναι:

- Η εποπτεία των συστημάτων του οργανισμού, σχετικά με περιστατικά και δεδομένα ασφαλείας σε πραγματικό χρόνο. Ίσως αποτελεί την βασικότερη λειτουργία των μηχανισμών SEM. Η λειτουργία αυτή υλοποιείται συνήθως με τους πράκτορες (agents) λογισμικού. Οι πράκτορες αυτοί είναι προγραμματισμένοι και ρυθμισμένοι ώστε με βάση τα δεδομένα που προσκομίζονται από τα διάφορα υποσυστήματα, να ανιχνεύουν γεγονότα ασφαλείας. Τα δεδομένα αυτά μεταφέρονται με ένα είδος λογισμικού που ονομάζεται σύνδεσμοι (connectors). Οι σύνδεσμοι διαφέρουν ανάλογα με το λειτουργικό σύστημα στο οποίο εκτελείται το σύστημα SIEM. Για παράδειγμα σε μηχανήματα που έχουν Microsoft Windows ως λειτουργικό σύστημα συνήθως οι διαδικασίες Windows System Log και Windows Application Log, χρησιμοποιούνται ως σύνδεσμοι από το σύστημα SEM. Οι πράκτορες των συστημάτων SEM διαφέρουν με τις υπηρεσίες άντλησης και αποθήκευσης logs, καθώς συνήθως οι λειτουργίες τους είναι πολύ πιο σύνθετες.

- Η ενημέρωση των κατάλληλων παραγόντων σε περιπτώσεις εντοπισμού συγκεκριμένων περιστατικών ασφαλείας. Μια επίσης σημαντική λειτουργία αποτελεί η ενημέρωση ανθρώπων από το σύστημα σε ορισμένες περιπτώσεις. Οι περιπτώσεις αυτές καθορίζονται κατά την εγκατάσταση του συστήματος από τους διαχειριστές. Το ποιες προϋποθέσεις καθορίζονται κρίσιμες και χρίζουν άμεσης αντιμετώπισης, εξαρτάται από την πολιτική και το πρωτόκολλο του κάθε οργανισμού. Αφού το σύστημα εντοπίσει μια κρίσιμη κατάσταση, ενημερώνει τους διαχειριστές με διαφορετικούς πιθανούς τρόπους. Κάποιοι δημοφιλείς τρόποι είναι μήνυμα SMS, e-mail, τηλεφωνική κλίση. Οι τρόποι αυτοί εξαρτώνται από το κάθε σύστημα SEM και το αν αυτό τους υποστηρίζει.

- Η ανίχνευση απειλών για την ασφάλεια του επιβλεπόμενου συστήματος. Αυτή ίσως είναι και πιο περίπλοκη λειτουργία των συστημάτων SEM. Τα δεδομένα που λαμβάνουν σε πραγματικό χρόνο από τους πράκτορες που είναι τοποθετημένοι σε διάφορους σταθμούς, θα πρέπει να κατατάσσονται σε απειλές και μη. Υπάρχουν περιπτώσεις που ενώ δεν υπάρχει κάποια απειλή για το σύστημα τα SIM αντιλαμβάνονται πως υπάρχει και δημιουργούν λανθασμένους συναγερμούς (false positives). Βέβαια στην ασφάλεια των υπολογιστών είναι προτιμότερο να υπάρχουν λανθασμένοι συναγερμοί ενώ δεν υπάρχει απειλή, παρά να μην υπάρξει συναγερμός ενώ υπάρχει απειλή. Οι κανόνες που καθορίζουν τις απειλές ρυθμίζονται από τους διαχειριστές. Στον χώρο της ασφάλειας υπάρχουν διάφορα πρότυπα τα οποία καθορίζουν και περιγράφουν απειλές. Πολλά SEM υποστηρίζουν ένα υποσύνολο των προτύπων ώστε να είναι ευκολότερη η προσθήκη κανόνων σχετικά με το ποια συμβάντα θεωρούνται απειλές. Η ενότητα 2.4 αναλύει κάποια από τα πιο διαδεδομένα πρότυπα στο χώρο της ασφάλειας.



Εικόνα 2.2. Παράδειγμα αρχιτεκτονικής SEM

Σε αυτό το σημείο είναι γίνεται εμφανές πως η δυσκολία στην εφαρμογή ενός συστήματος SEM έναντι αυτής ενός συστήματος SIM είναι πολύ μεγαλύτερη.

Οι δύο (2) βασικοί παράγοντες που καθιστούν την χρήση του SEM πιο δύσκολη είναι:

1. Για την επίβλεψη πολλαπλών μηχανημάτων ή συσκευών απαιτούνται πολλαπλοί πράκτορες που θα πρέπει εγκατασταθούν στα συστήματα αυτά. Σε περιπτώσεις περίπλοκων συστημάτων με πολλές συσκευές και διαφορετικά δίκτυα μεταξύ τους η διαδικασία αυτή γίνεται αρκετά περίπλοκη.

2. Οι προσθήκη και η διαχείριση των κανόνων ασφαλείας. Πλέον καθώς υπάρχουν πολλές διαφορετικές συσκευές με διαφορετικά λειτουργικά συστήματα, είναι προφανές πως ο αριθμός των απειλών είναι ανάλογος των συσκευών αυτών. Επίσης πολλές φορές δημιουργούνται νέες απειλές ή προστίθενται νέες συσκευές στο σύστημα, κάτι που σημαίνει πως οι κανόνες θα πρέπει να ενημερώνονται κάθε φορά που συμβαίνει κάτι τέτοιο. Σε μεγάλη κλίμακα η διαδικασία αυτή είναι ιδιαίτερα δύσκολη, λαμβάνοντας υπόψη και την υπευθυνότητα που χρειάζεται σε μια τέτοια διαδικασία.

2.3 Λειτουργίες και αρχιτεκτονική των συστημάτων SIEM

Αξιοποιώντας και συνδυάζοντας τις λειτουργίες και τα πλεονεκτήματα των συστημάτων SIM και SEM [9], τα συστήματα SIEM προσφέρουν μια πιο ολοκληρωμένη λύση ως σύστημα ασφαλείας για μία επιχείρηση ή έναν οργανισμό.

Η χρήση τους συναντάται σε πολλούς οργανισμούς, ειδικά μεσαίας προς μεγάλης κλίμακας οπού υπάρχουν περίπλοκα συστήματα με πολλαπλές συσκευές και ως συνέπεια, πολλές πιθανές απειλές. Στην αγορά υπάρχουν πολλά συστήματα SIEM από διαφορετικούς κατασκευαστές με διαφορετικές επιπρόσθετες λειτουργίες και δυνατά σημεία. Ταυτόχρονα υπάρχουν και λύσεις ανοιχτού κώδικα ομοίως διαφορετικά υλοποιημένες που η κάθε μια προσφέρει τις δικές της λειτουργίες σε έναν οργανισμό.

2.3.1 Λειτουργίες συστημάτων SIEM

Από το καιρό που άρχισαν να υλοποιούνται τα πρώτα συστήματα SIEM, υπήρχαν διάφορες έρευνες και εργασίες [5],[6],[9],[11] σχετικά με τα συστήματα αυτά, το πως λειτουργούν και το ποιες είναι οι βασικές τους λειτουργίες και ποιες οι συμπληρωματικές. Οι απόψεις πάνω σε αυτό το θέμα ποικίλουν, διαφοροποιώντας σε μικρό βαθμό το ποιες λειτουργίες θα πρέπει να υλοποιεί ένα SIEM. Αυτό έχει ως συνέπεια η γραμμή που διαχωρίζει τις λειτουργίες σε βασικές και συμπληρωματικές [12] να μην είναι ξεκάθαρη. Μεγάλο ρόλο σε αυτό παίζει και η ραγδαία ανάπτυξη του τομέα της πληροφορικής και ως

συνέπεια και των SIEM. Τα συστήματα SIEM σήμερα έχουν πολλές παραπάνω δυνατότητες από αυτά τις προηγούμενης δεκαετίας.

Μια κοινώς αποδεκτή προσέγγιση είναι πως το SIEM αντιμετωπίζεται ως σύστημα που θα πρέπει να έχει τις επόμενες, έξι (6) βασικές λειτουργίες:

- Συλλογή δεδομένων (Data collection).

  Για να υλοποιήσουν αυτή τη λειτουργία τα συστήματα SIEM χρησιμοποιούν πράκτορες στα συστήματα από τα οποία πρέπει να αντλήσουν πληροφορίες. Οι πράκτορες είναι λογισμικό προσαρμοσμένο στο λειτουργικό σύστημα από το οποίο συλλέγει δεδομένα. Για παράδειγμα ένα πράκτορας που επιβλέπει τη λειτουργία μιας βάσης δεδομένων που τρέχει σε έναν Linux server, διαφέρει από έναν πράκτορα που επιβλέπει τις υπηρεσίες που τρέχουν σε έναν υπολογιστή με λειτουργικό σύστημα Windows.

  Η συλλογή δεδομένων γίνεται αυτόματα, καθώς οι πράκτορες έχουν τη δυνατότητα να αναγνωρίζουν συμβάντα και αλλαγές στο σύστημα, τα οποία μεταφέρουν ως δεδομένα στο σύστημα SIEM.

- Κανονικοποίηση δεδομένων (Data normalization).

  Τα δεδομένα που συλλέγονται από τους πράκτορες, προέρχονται από διαφορετικές πηγές και έχουν διαφορετική μορφή. Τα logs, μιας MySQL βάσης δεδομένων διαφέρουν σημαντικά με τα logs, ενός τοίχους προστασίας (Firewall).

  Για ένα σύστημα SIEM είναι απαραίτητο να διαθέτει έναν μηχανισμό με τον οποίο πριν αποθηκεύσει τα δεδομένα αυτά να τα μετατρέψει σε μια συγκεκριμένη μορφή, κοινή για όλα τα δεδομένα, ανεξαρτήτου προέλευσης.

- Αποθήκευση δεδομένων (Data storage).

  Είναι σημαντικό τα δεδομένα που συλλέγει και κανονικοποιεί ένα SIEM να αποθηκεύονται.

  Ο βασικός λόγος είναι η βελτίωση της ανάλυσης των δεδομένων. Συχνά η χρήση παλαιότερων δεδομένων από συμβάντα και περιστατικά ασφαλείας μπορεί να δώσει διαφορετική ερμηνεία στα αποτελέσματα της ανάλυσης από ότι η χρήση μόνο των πρόσφατων δεδομένων ή δεδομένων πραγματικού χρόνου.

  Επίσης το σύστημα SIEM θα πρέπει να διασφαλίσει τα δεδομένα που αποθηκεύει.

  Πολλά από τα δεδομένα αυτά μπορεί να είναι εμπιστευτικά, και δεν θα πρέπει να είναι ελεύθερη η πρόσβαση σε αυτά. Τέλος η επεξεργασία και η διαγραφή των δεδομένων θα πρέπει να γίνεται μόνο από τους διαχειριστές ή άλλα δικαιοδοτημένα πρόσωπα.

- Ανάλυση και οπτικοποίηση δεδομένων.

  Η πλειοψηφία των συστημάτων SIEM που χρησιμοποιούνται, έχουν μηχανισμό ανάλυσης δεδομένων και οπτικοποίηση δεδομένων.

  Κατά την ανάλυση χρησιμοποιούνται υποσύνολα αποθηκευμένων δεδομένων ή δεδομένων πραγματικού χρόνου ανάλογα με το είδος της ανάλυσης.

  Η ανάλυση δεδομένων χωρίζεται σε δύο (2) βασικές κατηγορίες:

  - Αυτόματη ανάλυση. Η διαδικασία της ανάλυσης πραγματοποιείται αυτόματα καθώς το SIEM λαμβάνει δεδομένα. Η ανάλυση αυτή μπορεί να είναι είτε σε πραγματικό χρόνο, είτε περιοδικά υπό προϋποθέσεις

  - Χειροκίνητη ανάλυση. Αυτό το είδος ανάλυσης πραγματοποιείται συνήθως από το διαχειριστή του συστήματος σε περιπτώσεις που υπάρχει ανάγκη για στοχευμένη ανάλυση δεδομένων ή δημιουργία αναφορών.

  Η οπτικοποίηση είναι η διαδικασία της αναπαράστασης των δεδομένων σε μια διεπαφή χρήστη (user interface). Τα δεδομένα αυτά μπορεί να είναι είτε αποτελέσματα ανάλυσης, είτε άλλα δεδομένα, γραφήματα, σχήματα, μπάρες μετρήσεων κτλ.

- Alerting.
  Η λειτουργία alerting , δηλαδή ειδοποίησης των καθορισμένων προσώπων είναι ιδιαίτερα σημαντική, για έναν μηχανισμό όπως ο SIEM που επιβλέπει ένα σύστημα σε πραγματικό χρόνο.

  Οι συνθήκες υπό τις οποίες θα ειδοποιηθεί ένα πρόσωπο ρυθμίζονται και καθορίζονται από τους διαχειριστές του εκάστοτε συστήματος SIEM, συνήθως χειροκίνητα καθώς οι συνθήκες αυτές εξαρτώνται κυρίως από τις λεπτομερείς του συστήματος.

  Κάποιοι βασικοί τρόποι ειδοποιήσεων που υπάρχουν στα περισσότερα συστήματα SIEM είναι:

  - Αποστολή mail σε καθορισμένες διευθύνσεις ηλεκτρονικού ταχυδρομείου. Αυτός ο τρόπος είναι ο προκαθορισμένος στα περισσότερα συστήματα SIEM και είναι ο πιο εύκολος στην υλοποίηση και στη ρύθμιση.

  - Τηλεφωνικές κλήσεις και μηνύματα SMS.

    Αυτός ο τρόπος δεν υποστηρίζεται από όλα τα συστήματα SIEM και είναι δυσκολότερος στο να υλοποιηθεί.
    Χρησιμοποιείται κυρίως είτε σε ιδιαίτερα σημαντικά υπολογιστικά συστήματα, είτε σε ιδιαίτερα κρίσιμες καταστάσεις
  - Ειδικές εφαρμογές για κινητά ή υπολογιστές.

Ορισμένα συστήματα SIEM έχουν τη δυνατότητα να συνδέονται με εφαρμογές στο κινητό ενός διαχειριστή και να στέλνουν μέσω της εφαρμογής ειδοποιήσεις σχετικά με συμβάντα.

- Συσχέτιση δεδομένων και συμβάντων (correlation).

Καθώς το σύστημα SIEM λαμβάνει δεδομένα σχετικά με συμβάντα, αφού τα κανονικοποιήσει, θα πρέπει να είναι σε θέση να αναγνωρίσει αν ένα συμβάν προέρχεται από κακόβουλη ενέργεια ή όχι. Αυτό το κομμάτι είναι ιδιαίτερα δύσκολο και περίπλοκο και συχνά υπάρχουν περιπτώσεις όπου κανονικά συμβάντα κατατάσσονται ως κακόβουλα.

Επίσης συχνά ένα μεμονωμένο συμβάν δεν αποτελεί απειλή, αλλά ένας συνδυασμός από συμβάντα είναι πιθανό να είναι κακόβουλα.

Χαρακτηριστικό παράδειγμα είναι οι διάφορες επιθέσεις DoS(Denial of Service). Ένα απλό ping προς έναν σταθμό εντός του επιβλεπόμενου συστήματος δεν αποτελεί ύποπτη συμπεριφορά η απειλή. Πολλαπλές ταυτόχρονες τέτοιες ενέργειες συχνά από διαφορετικές τοποθεσίες, παραπέμπουν σε επίθεση DoS.

Το σύστημα SIEM θα πρέπει να είναι σε θέση να αντιλαμβάνεται τέτοιες περιπτώσεις και να προβαίνει στις απαραίτητες ενέργειες.

Για τη συσχέτιση των συμβάντων το SIEM χρησιμοποιεί κανόνες. Όταν οι προϋποθέσεις που καθορίζονται στους κανόνες πληρούνται, τότε το σύστημα καταλαβαίνει πως η ενέργεια είναι κακόβουλη.

Οι κανόνες αυτοί συχνά προθέτονται χειροκίνητα από τους διαχειριστές.

Στη δημιουργία ενός αποτελεσματικού συνόλου κανόνων συσχέτισης συμβάλουν σημαντικά τα διάφορα πρότυπα συλλογής και αναπαράστασης πληροφοριών ασφαλείας.

Με τη χρήση των προτύπων αυτών, χρήστες μπορούν να καταγράψουν νέους κανόνες για νέες απειλές. Οι κανόνες αυτοί ακολουθούν συγκεκριμένη μορφή και συχνά χρησιμοποιούνται για την αποτελεσματική εναλλαγή πληροφοριών ασφαλείας μεταξύ ατόμων, ομάδων ή οργανισμών.

Εκτός από τις βασικές λειτουργίες πολλά SIEM παρέχουν και επιπρόσθετες λειτουργίες, κάποιες από τις οποίες αποκτούν όλο και μεγαλύτερη δημοτικότητα.

Ορισμένες από αυτές τις λειτουργίες είναι:

- Αλγόριθμοι μηχανικής μάθησης.

Ορισμένα SIEM, χρησιμοποιούν εκτός από γραμμένους κανόνες συσχέτισης, εκπαιδευμένους αλγόριθμους μηχανικής μάθησης. Οι αλγόριθμοι αυτοί εκπαιδεύονται και δοκιμάζονται σε μεγάλα σύνολα δεδομένων ασφαλείας από διάφορες πηγές.

Ένα από τα βασικότερα πλεονεκτήματα των αλγορίθμων μηχανικής μάθησης η δυνατότητα εντοπισμού απειλής που δεν ήταν προηγουμένως γνωστή ή καταγεγραμμένη στους κανόνες συσχέτισης.

- Αυτόματη διαχείριση συμβάντων.

  Μια ακόμη χρήσιμη επιπρόσθετη λειτουργία είναι αυτή της αυτόματης διαχείρισης συγκεκριμένων συμβάντων.

  Ορισμένα SIEM δίνουν τη δυνατότητα στο χρήση να δημιουργήσει κανόνες και ενέργειες που θα ληφθούν αυτόματα όταν εντοπιστεί ένα συγκεκριμένο συμβάν ασφαλείας.

  Συνήθως οι ενέργειες αυτές στοχεύουν στην έγκαιρη επέμβαση σε περιπτώσεις που απειλείται το σύστημα.

  Για παράδειγμα:

  Αν το σύστημα εντοπίσει προσπάθεια brute force επίθεσης και η υπηρεσία που δέχεται την επίθεση δεν διαθέτει μηχανισμό timeout, το σύστημα θα αποκλείσει αυτόματα τη διεύθυνση από την οποία προέρχεται η επίθεση.

  Αν το σύστημα αντιληφθεί παραβίαση ενός δικτύου ή μόλυνση κάποια συσκευής ενός δικτύου από κακόβουλο λογισμικό, έχει τη δυνατότητα να αποκλείσει το συγκεκριμένο δίκτυο από το υπόλοιπο, ώστε να σταματήσει έγκαιρα την πιθανή εξάπλωση του κακόβουλου λογισμικού

- Μηχανισμοί υποστήριξης διάφορων πολιτικών λειτουργίας.

  Πλέον υπάρχουν διαφορετικές πολιτικές και πρότυπα λειτουργίας. Ορισμένα από αυτά καθορίζονται από τον οργανισμό, και άλλα από κρατικούς φορείς.

  Συνήθως οι εταιρίες καθορίζουν μόνες τους τα εσωτερικά πρότυπα και πολιτικές λειτουργίες, ενώ είναι υποχρεωμένες να τηρούν ταυτόχρονα αυτές που καθορίζονται από το κράτος.

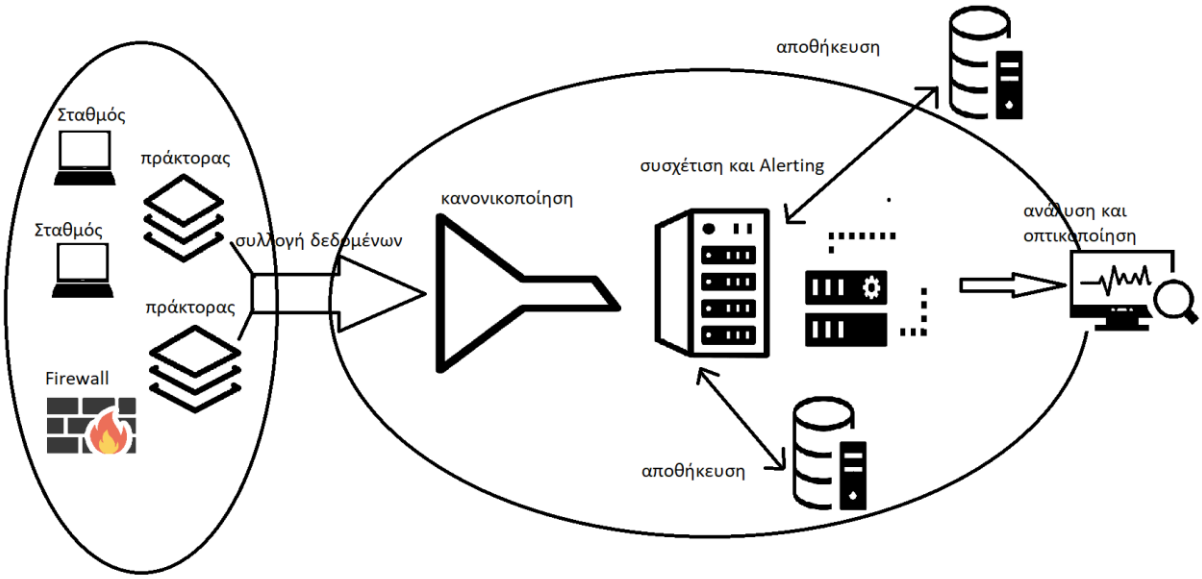
  Χαρακτηριστικό παράδειγμα είναι η πολιτική προστασίας προσωπικών δεδομένων GDPR, που καθορίστηκε από την Ευρωπαϊκή Ένωση και ισχύει και στην Ελλάδα.

  Πολλά σύγχρονα συστήματα SIEM παρέχουν μηχανισμούς υποστήριξης γνωστών προτύπων και πολιτικών ενώ δίνουν τη δυνατότητα ανάπτυξης εταιρικών πολιτικών και προτύπων.

- Μηχανισμός δημιουργίας security assessments. Η δημιουργία assessment είναι αρκετά σημαντική καθώς επιτρέπει στο χρήστη να εξάγει τα δεδομένα σχετικά με περιστατικά ασφαλείας, σε συγκεκριμένα πρότυπα καθώς ο μηχανισμός αυτός συνήθως υποστηρίζει τα ποιο γνωστά πρότυπα.

2.3.2 Αρχιτεκτονική συστημάτων SIEM

Η αρχιτεκτονική που συναντάται στα σύγχρονα SIEM [7], αποτελείται από αρκετούς ξεχωριστούς μηχανισμούς που αλληλοεπιδρούν μεταξύ τους δίνοντας ένα ολοκληρωμένο αποτέλεσμα, όπως για παράδειγμα στο σχήμα 2.3



Σχήμα 2.3. Παράδειγμα αρχιτεκτονικής SIEM

Στο σχήμα 2.3, το δεξί κυκλικό διάγραμμα αναπαριστά το εσωτερικό ενός συστήματος SIEM. Οι πράκτορες συλλέγουν τα δεδομένα και είτε να στέλνουν κατευθείαν στο SIEM είτε τα στέλνουν σε κάποιον μηχανισμό κανονικοποίησης που διαθέτουν τα περισσότερα SIEM.

Αρκετοί πράκτορες έχουν τη δυνατότητα να φιλτράρουν και οι ίδιοι τα δεδομένα, αλλά αυτό δεν αντικαθιστά τους μηχανισμούς κανονικοποίησης οι οποίοι είναι πιο ευέλικτοι με περισσότερες λειτουργίες.

Αφού τα δεδομένα περάσουν τους κατάλληλους ελέγχους και τα κατάλληλα φιλτραρίσματα, ελέγχονται με βάση τους κανόνες του συστήματος συσχέτισης συμβάντων. Αυτό συμβάλει στον άμεσο εντοπισμό πιθανής απειλής και ενημέρωση του μηχανισμού Alerting για να κάνει τις προγραμματισμένες ενέργειες ανάλογα με την περίπτωση.

Είναι σημαντικό να σημειωθεί πως πολλές από τις σχέσεις αυτές είναι αμφίδρομες. Για παράδειγμα μπορεί ο μηχανισμός ανάλυσης και οπτικοποίησης να ζητήσει παλαιότερα δεδομένα από κάποια βάση δεδομένων.

Το αριστερό σχήμα αναπαριστά διάφορες συσκευές στο ίδιο ή διαφορετικό δίκτυο, στις οποίες υπάρχουν εγκατεστημένοι οι κατάλληλοι πράκτορες που στέλνουν δεδομένα και επιβλέπουν.

Πρέπει να σημειωθεί πως το σχήμα 2.3 αναπαριστά μια σχετικά απλοϊκή αρχιτεκτονική συστήματος SIEM. Σε μεγαλύτερους οργανισμούς, πολλοί από τους μηχανισμούς μπορεί να βρίσκονται σε ξεχωριστά δίκτυα οι να υπάρχουν περισσότερα από ένα συστήματα SIEM (cluster architecture).

2.4 Πρότυπα στο πεδίο της ασφάλειας

Τα πρότυπα στο χώρο της ασφάλειας των υπολογιστικών συστημάτων, παίζουν ιδιαίτερα σημαντικό ρόλο. Με τη χρήση προτύπων επιστήμονες, εργαζόμενοι και οργανισμοί αυτού του τομέα μπορούν να μοιράζονται αποτελεσματικότερα πληροφορίες.

Τα πρότυπα αναπαράστασης πληροφοριών και δεδομένων ασφάλειας υπολογιστών, άρχισαν να χρησιμοποιούνται από τις αρχές της ανάπτυξης του επιστημονικού τομέα της ασφάλειας των υπολογιστών. Πλέον υπάρχουν πολλά διαφορετικά πρότυπα, με διαφορετικές εξειδικεύσεις και πλεονεκτήματα.

Σε αυτή την ενότητα θα αναλυθούν κάποια από τα πιο γνωστά πρότυπα που χρησιμοποιούνται. Είναι σημαντικό να αναφερθεί πως και τα συστήματα SIEM αξιοποιούν τα διάφορα πρότυπα που υπάρχουν για να προσαρμόσουν τους κανόνες συσχέτισης συμβάντων και όχι μόνο. Εκτός από αυτό βέβαια είναι αναγκαίο ένα σύστημα SIEM να ανανεώνει τους κανόνες που χρησιμοποιεί καθώς δημιουργούνται νέες απειλές οι οποίες συνήθως γνωστοποιούνται με τη χρήση προτύπων.

2.4.1 CVE

Το πρότυπο CVE [27] είναι από τα παλαιότερα και τα πιο γνωστά στο χώρο της ασφάλειας. Δημιουργήθηκε από το μη κερδοσκοπικό MITRE. Ο οργανισμός αυτός είναι χρηματοδοτούμενος από κρατικούς φορείς ως ερευνητικός οργανισμός και ήταν από τους πρώτους οργανισμούς που ασχολήθηκαν με τον επιστημονικό τομέα της ασφάλειας, καθώς ιδρύθηκε το 1958.

Το CVE που είναι συντομογραφία της πρότασης Common Vulnerabilities and Exposures, ορίζει ένα πρότυπο αναφοράς σε κενά ασφαλείας και τρωτότητες.

Οι αναφορές CVE έχουν συγκεκριμένη μορφή και η διαδικασία για τη δημιουργία αναφοράς είναι επίσης συγκεκριμένη και ορισμένη από τον οργανισμό MITRE.

2.4.1.1 Μορφή αναφοράς CVE

Όπως αναφέρθηκε προηγουμένως, οι αναφορές CVE ακολουθούν μια προκαθορισμένη μορφή [27].

Το βασικό στοιχείο κάθε αναφοράς είναι το CVE identifier number, ή αλλιώς CVE ID. Έχει τη μορφή CVE-1234, όπου το αριθμητικό μέρος αποτελεί έναν μοναδικό αριθμό που ξεχωρίζει την αναφορά από τις υπόλοιπες.

Εκτός από τον CVE ID μια αναφορά περιέχει ορισμένα ακόμα πεδία.

Τα πεδία αυτά είναι:

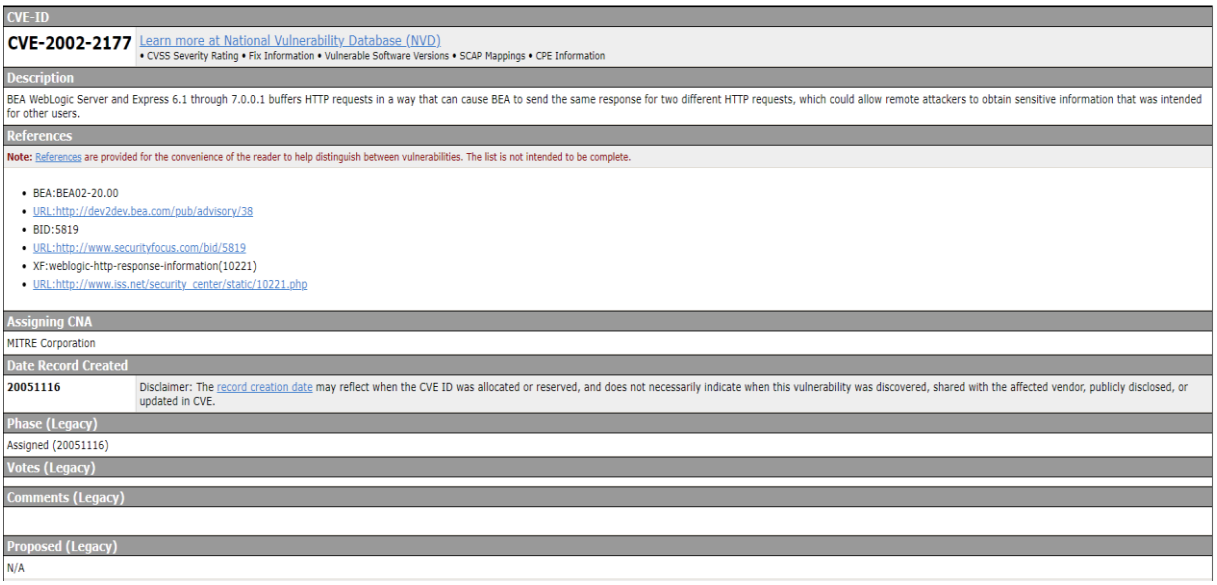
- Description (Περιγραφή). Περιέχει μια σύντομη περιγραφή σχετικά με τη τρωτότητα ή το κενό ασφαλείας.

- References (Αναφορές). Συνήθως περιέχει αναφορές σε εξωτερικού ιστότοπους ή άρθρα, τα οποία βοηθάνε στη κατανόηση της αναφοράς.

- Assigning CNA. Το πεδίο Assigning CNA καθορίζει τον οργανισμό που δημιούργησε την αναφορά. Υπάρχουν ορισμένοι οργανισμοί εγκεκριμένοι από τη MITRE που έχουν τη δικαιοδοσία να καταχωρούν αναφορές. Τέτοιοι οργανισμοί είναι συνήθως μεγάλες εταιρίες στο τομέα της πληροφορικής, όπως για παράδειγμα Microsoft, Apple, Oracle κτλ.

- Data record created. Περιέχει την ημερομηνία δημιουργίας της συγκεκριμένης αναφοράς.

- Legacy πεδία. Επίσης σε κάποιες παλαιότερες εγγραφές υπάρχουν διάφορα πεδία που πλέον δεν χρησιμοποιούνται όπως για παράδειγμα πεδία ψήφων ή σχολίων.



CVE-ID

CVE-2002-2177 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

BEA WebLogic Server and Express 6.1 through 7.0.0.1 buffers HTTP requests in a way that can cause BEA to send the same response for two different HTTP requests, which could allow remote attackers to obtain sensitive information that was intended for other users.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BEA:BEA02-20.00
- URL:http://dev2dev.bea.com/pub/advisory/38
- BID:5819
- URL:http://www.securityfocus.com/bid/5819
- XF:weblogic-http-response-information(10221)
- URL:http://www.iss.net/security_center/static/10221.php

Assigning CNA

MITRE Corporation

Date Record Created

20051116 Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20051116)

Votes (Legacy)

Comments (Legacy)

Proposed (Legacy)

N/A

Σχήμα 2.4 Παράδειγμα αναφοράς CVE [27]

2.4.1.2 Διαδικασία καταχώρισης αναφοράς CVE

Για να καταχωρηθεί μια νέα αναφορά CVE υπάρχει συγκεκριμένη διαδικασία.

Αρχικά όπως προαναφέρθηκε και στην ενότητα 2.4.1.1, οι καταχωρίσεις των αναφορών πραγματοποιούνται από συγκεκριμένους οργανισμούς που ονομάζονται CNA (CVE Numbering Authority). Οι οργανισμοί αυτοί εγκρίνονται από τον οργανισμό MITRE μετά από αίτηση τους.

Το πρώτο βήμα για τη καταχώριση μιας αναφοράς είναι η επικοινωνία με έναν οργανισμό καταχώρισης. Συνήθως με τον MITRE. Εφόσον η αναφορά εγκριθεί από τον οργανισμό δημιουργείται ένα νέο CVE ID για τη συγκεκριμένη αναφορά και η αναφορά τοποθετείτε στην ιστοσελίδα της CVE και καταχωρείτε στη βάση δεδομένων NVD του οργανισμού NIST. Η βάση δεδομένων NVD (National Vulnerability Database) περιέχει αναφορές CVE και άλλες αναφορές σχετικά με κενά ασφαλείας.

Ένας γενικός κανόνας καταχωρίσεων αναφορών είναι πως μια αναφορά θα πρέπει να περιγράφει ένα συγκεκριμένο κενό ασφαλείας ή πρόβλημα. Αυτό σε πολλές περιπτώσεις δεν είναι απλό να γίνει καθώς υπάρχουν περιπτώσεις που τα κενά ασφαλείας σχετίζονται άμεσα μεταξύ τους. Παρόλα αυτά υπάρχει η δυνατότητα ένωσης πολλαπλών CVE σε ένα ή διαχωρισμού ενός CVE σε πολλά.

Είναι σημαντικό να αναφερθεί πως για να καταχωρηθεί μια αναφορά στη βάση δεδομένων NVD και στη CVE θα πρέπει να αφορά λογισμικό ή υπηρεσία που είναι δημόσια προσβάσιμη από όλους, άσχετα με το αν είναι δωρεάν ή όχι.

2.4.2 STIX και TAXII

Το STIX και το TAXII [26]είναι αποτελούν δύο (2) πρότυπα σχεδιασμένα με σκοπό να βελτιώσουν την προστασία από επιθέσεις σε υπολογιστικά συστήματα, μέσω της οργανωμένης ανταλλαγής πληροφοριών σχετικά με απειλές στο χώρο της κυβερνοασφάλειας Το STIX ορίζει της απειλές καθ' αυτές, ενώ το TAXII αποτελεί ένα πρωτόκολλο επικοινωνίας των πληροφοριών σχετικά με απειλές και επιθέσεις. Το STIX και το TAXII αρχικά δημιουργήθηκαν κατά τη συνεργασία των οργανισμών MITRE και OASIS Open. Η αρχική πρώτη τους έκδοση βρισκόταν στο domain του οργανισμού MITRE, ενώ η δεύτερη τρέχουσα έκδοση συντηρείτε κυρίως από τον OASIS Open.

2.4.2.1 STIX

Το STIX [26] όπως αναφέρθηκε προηγουμένως είναι ένα πρότυπο που ορίζει μια γλώσσα και κωδικοποίηση με σκοπό την ανταλλαγή πληροφοριών σχετικά με απειλές στο χώρο της κυβερνοασφάλειας. Εν συντομία αυτές οι πληροφορίες ονομάζονται CTI (cyber threat intelligence). Το STIX είναι ανοιχτού κώδικα, ο οποίος είναι ελεύθερα διαθέσιμος.

Το STIX έχει δύο διαφορετικές εκδόσεις την STIX 1 και τη STIX 2. Πλέον χρησιμοποιείται κυρίως η STIX 2 ενώ η STIX 1 αντιμετωπίζεται κυρίως ως legacy έκδοση.

Οι δύο εκδόσεις έχουν κάποιες διαφορές μεταξύ τους. Οι σημαντικότερες είναι οι εξής:

- Η STIX 1 χρησιμοποιεί XML για να ορίσει τις πληροφορίες ενώ η STIX 2 JSON.
- Όλα τα αντικείμενα στη STIX 2 είναι του ανώτερου επιπέδου και όχι εμφωλευμένα όπως αυτά της STIX 1
- Η STIX 2 έχει τη δυνατότητα να χρησιμοποιεί patterns, κάτι που κάνει συχνά τον καθορισμό ενός πεδίου πολύ ευκολότερο, σε αντίθεση με τη STIX 1.

Η τρέχουσα έκδοση του STIX 2.1, ορίζει 18 αντικείμενα Domain (SDO) και 2 αντικείμενα σχέσεων (SRO). Στους παρακάτω πίνακες περιγράφονται τα αντικείμενα εν συντομία. Πρώτα είναι ο πίνακας με τα αντικείμενα SDO και ακολουθεί ο πίνακας με τα αντικείμενα SRO [26].

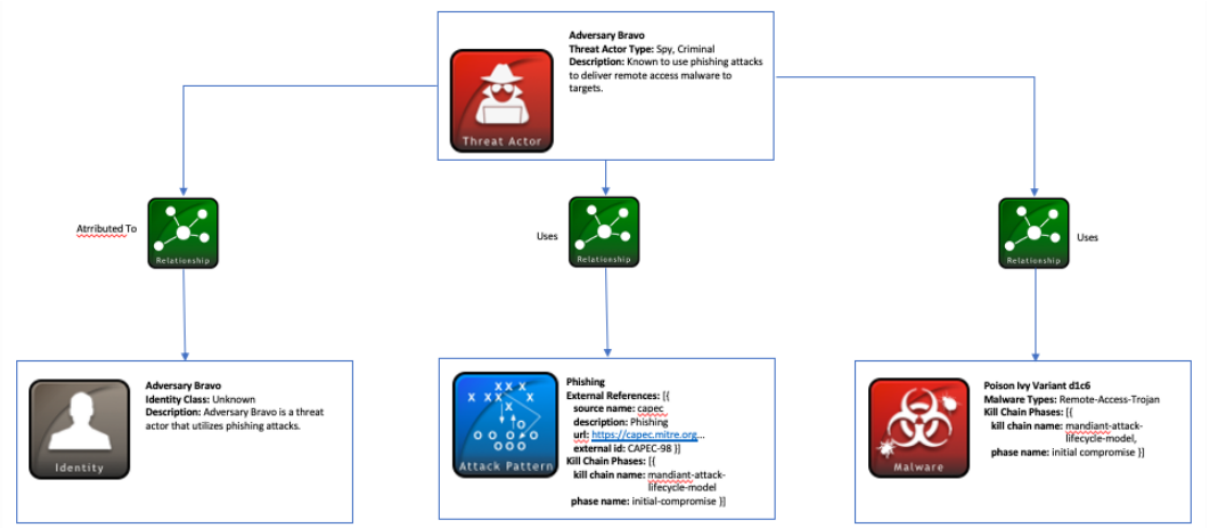
Αντικείμενο	Όνομα	Περιγραφή
	**Attack Pattern**	Περιγράφει τους τρόπους με τους οποίους γίνονται προσπάθειες για τη παραβίαση ενός στόχου
	**Campaign**	Ομάδα από συμπεριφορές που περιγράφουν ένα σύνολο από κακόβουλες ενέργειες ή επιθέσεις (μερικές φορές ονομάζονται κύματα) που συμβαίνουν για μια συγκεκριμένη χρονική διάρκεια απέναντι σε ένα σύνολο στόχων.
	**Course of Action**	Συμβουλή από έναν δημιουργό πληροφορίας προς έναν παραλήπτη σχετικά με τις ενέργειες που πρέπει να πράξουν ως απάντηση σε αυτές τις πληροφορίες.
	**Grouping**	Διαβεβαιώνει πως τα αντικείμενα STIX έχουν κοινό περιεχόμενο
	**Identity**	Πραγματικά πρόσωπα, οργανισμοί, ομάδες ή υποσύνολα
	**Indicator**	Περιέχει πρότυπο που μπορεί να χρησιμοποιηθεί για να εντοπίσει ύποπτες ή κακόβουλες ενέργειες.

	**Infrastructure**	Παρουσιάζει ένα σύστημα, μια υποδομή λογισμικού η οποιαδήποτε φυσική ή λογική πηγή.
	**Intrusion Set**	Ομαδοποιημένα σύνολα από συμπεριφορές και πόρους με κοινά χαρακτηριστικά που θεωρείται πως ελέγχονται από μία οργάνωση.
	**Location**	Περιγράφει μια γεωγραφική τοποθεσία.
	**Malware**	Περιγράφει κακόβουλο λογισμικό
	**Malware Analysis**	Τα μετα-δεδομένα που προέρχονται από μία στατική ή δυναμική ανάλυση κακόβουλου λογισμικού
	**Note**	Σημειώσεις για γενικές πληροφορίες είτε για επιπλέον πληροφορίες σχετικά με κάποιο αντικείμενο STIX
	**Observed Data**	Μεταφέρει πληροφορίες σχετικά με οντότητες που έχουν να κάνουν με την κυβερνοασφάλεια, όπως αρχεία, συστήματα, δίκτυα, με τη χρήση STIX Cyber-observable Objects (SCOs)
	**Opinion**	Εκτίμηση ορθότητας πληροφοριών σε ένα αντικείμενο STIX που δημιουργήθηκε από άλλη οντότητα.
	**Report**	Σύνολα από πληροφορίες ασφαλείας πάνω σε ένα ή περισσότερα θέματα, όπως κακόβουλο λογισμικό, τεχνικές επιθέσεων κτλ.
	**Threat Actor**	Πραγματικά πρόσωπα, ομάδες, οργανισμοί που εκτιμάται να έχουν κακόβουλες προθέσεις.
	**Tool**	Πραγματικό λογισμικό που μπορεί να χρησιμοποιηθεί για επιθέσεις.
	**Vulnerability**	Τρωτότητα στο λογισμικό που μπορεί εκμεταλλευτεί επιτιθέμενος για να πάρει πρόσβαση στο σύστημα ή στο δίκτυο

Πίνακας με STIX 2.1 SRO's [26].

Αντικείμενο	Όνομα	Περιγραφή
	**Relationship**	Χρησιμοποιείται για να ενώσει δύο (2) SDO ή SCO με σκοπό να περιγράψει πως συνδέονται μεταξύ τους
	**Sighting**	Δείχνει την πεποίθηση πως κάτι στο CTI έχει ήδη παρατηρηθεί

Η παρακάτω εικόνα περιγράφει μια περίπτωση όπου ένας κακόβουλος χρήστης χρησιμοποιεί συγκεκριμένες τεχνικές με κακόβουλες προθέσεις. Στο σχήμα περιγράφονται τα βασικά χαρακτηριστικά του χρήστη καθώς και οι κακόβουλες ενέργειες που προσπαθεί να πετύχει.



Εικόνα 2.5 Παράδειγμα διαγράμματος STIX [26]

2.4.2.2 TAXII

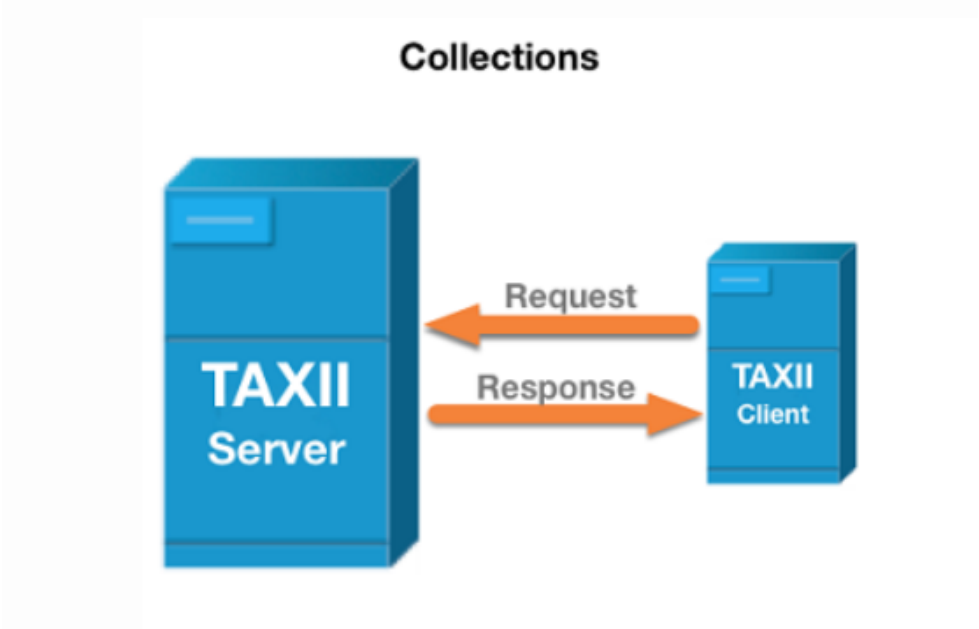
Το TAXII (Trusted Automated Exchange of Intelligence Information) [26] αποτελεί πρωτόκολλο επιπέδου εφαρμογής, το οποίο χρησιμοποιείται για την αποστολή και λήψη πληροφοριών CTI μέσω HTTPS, αλλά μπορεί να χρησιμοποιηθεί και για αποστολή άλλων δεδομένων μέσω HTTPS.

Η λειτουργία της αποστολής υλοποιείται με τη χρήση του RESTful API που ορίζει το TAXII. Το API αυτό λειτουργεί σε ένα μοντέλο client-server με τη χρήση Web requests. Οι clients στέλνουν requests στον server είτε για να ζητήσουν ένα έγγραφο CTI, είτε για να δημιουργήσουν, είτε για οποιαδήποτε άλλη λειτουργία CRUD, ο Server επεξεργάζεται το request και δίνει το κατάλληλο Response.

Τόσο για τους TAXII Clients όσο και για τους TAXII Servers, υπάρχουν συγκεκριμένες προϋποθέσεις για να συμμετέχουν στη διαδικασία ανταλλαγής πληροφοριών CTI που ορίζονται από το TAXII.

Το TAXII, χρησιμοποιεί δύο (2) βασικά μοντέλα [26] για να πραγματοποιήσει τις λειτουργίες του.

1. Collection (συλλογή). Ένα Collection είναι μια διεπαφή για ένα λογικό repository που περιέχει αντικείμενα CTI, τα οποία παρέχει ένας Server. Ο δημιουργός των αρχείων CTI έχει τη κατάλληλη πρόσβαση στον Server, και μπορεί να κρατάει εκεί το υποσύνολο των CTI που επιθυμεί να διανέμει στους χρήστες. Όσοι χρήστες έχουν πρόσβαση στο συγκεκριμένο server, μπορούν να αντλήσουν τα CTI που επιθυμούν. Οι χρήστες με τους Servers ανταλλάζουν πληροφορίες με το μοντέλο Request – Response. Στην εικόνα 2.13 φαίνεται ένα παράδειγμα υλοποίησης Collection.

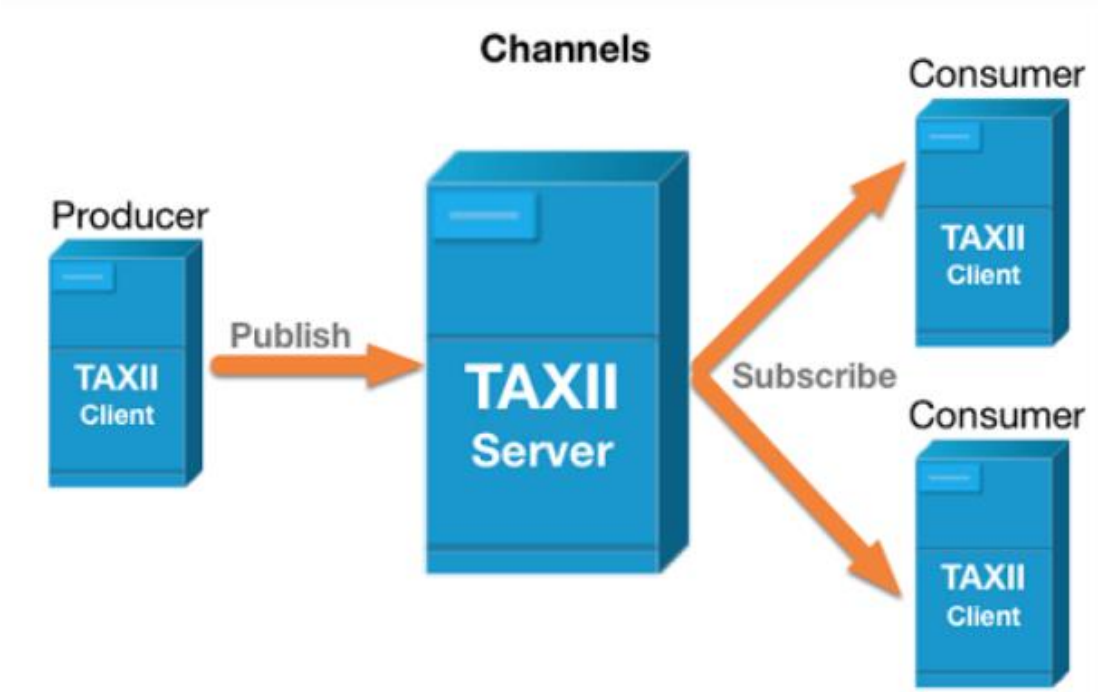


Εικόνα 2.6. Παράδειγμα Αρχιτεκτονικής TAXII Collection [26]

2. Channel (κανάλι). Αυτό το μοντέλο, είναι πιο ευέλικτο και συντηρείται από έναν TAXII Server, που δίνει τη δυνατότητα σε πολλούς δημιουργούς CTI, να προωθούν δεδομένα τους στο server. Ο server με τη σειρά του μπορεί να διανέμει τα δεδομένα CTI προς πολλούς χρήστες. Αυτό σημαίνει πως αφού οι δημιουργοί που προωθούν τα δεδομένα στο server μπορεί να είναι περισσότεροι από έναν, οι χρήστες έχουν τη δυνατότητα με της πρόσβαση σε έναν server, να έχουν πρόσβαση σε CTI από πολλαπλούς δημιουργούς

Για να οργανωθεί καλύτερα αυτό το μοντέλο χρησιμοποιείται η λογική του publisher – subscriber. Οι TAXII clients έχουν πρόσβαση στα δεδομένα μόνο από τους δημιουργούς στους οποίους έχουν εγγραφεί. Με αυτό το τρόπο ο κάθε χρήστης έχει πρόσβαση μόνο σε ένα υποσύνολο από τα CTI του server, δηλαδή αυτά για τα οποία έχει κάνει εγγραφή

Χαρακτηριστικό παράδειγμα φαίνεται στην εικόνα 2.7



Εικόνα 2.7. Παράδειγμα Αρχιτεκτονικής TAXII Channel [26]

2.4.3 OVAL

Το OVAL (Open Vulnerability and Assessment Language) [28] είναι ένα παγκόσμιο πρότυπο που στοχεύει στην ανοιχτή ανταλλαγή πληροφοριών σχετικά με το τομέα της ασφάλειας. Το OVAL δημιουργήθηκε από τον οργανισμό MITRE και είναι ένα πρότυπο ανοιχτό και ελεύθερο στη χρήση του. Το OVAL ορίζει συγκεκριμένα αρχεία που περιγράφουν δεδομένα ασφαλείας. Τα αρχεία αυτά υπάρχουν σε δημοσίως προσβάσιμο repository, και μπορούν να χρησιμοποιηθούν από οποιονδήποτε. Αυτός ήταν και ο βασικός σκοπός του OVAL, η δημιουργία τέτοιο αρχειοθηκών όπου οι χρήστες θα προσθέτουν και θα χρησιμοποιούν αρχεία δημιουργώντας μια πλούσια βιβλιοθήκη αρχείων OVAL.

Τα τρία (3) βασικά στοιχεία του OVAL είναι [28]:

- Η γλώσσα XML για την αποτύπωση πληροφοριών σχετικά με συστήματα υπολογιστών και σχετικών με αυτά πληροφοριών.

- Το repository που περιέχει αρχεία γραμμένα σε OVAL

- Πρόγραμμα συμβατότητας OVAL, που ορίζει κανόνες και τεχνικές για την αποτελεσματικότερη συγγραφή αρχείων OVAL.

Η γλώσσα OVAL ορίζει τα τρία βασικά βήματα [28] της διαδικασίας δημιουργίας ενός assessment αρχείου.

Τα βήματα αυτά είναι:

- Αποτύπωση πληροφοριών σχετικά με το σύστημα που θα αναλυθεί.

- Ανάλυση του συστήματος και προσδιορισμός της κατάστασης του, όπως (ανανεωμένο, τρωτό, μη συμμορφούμενο, κτλ.)

- Καταγραφή των αποτελεσμάτων της ανάλυσης στο αρχείο assessment.

Με βάση αυτά τα βήματα, ένα αρχείο OVAL αποτελείται από τρία (3) XML σχήματα.

Τα σχήματα αυτά είναι:

- OVAL Definition Schema
  XML αρχείο που καταγράφει τις διαβεβαιώσεις σχετικά με το σύστημα

- OVAL System Characteristics Schema
  XML αρχείο που καταγράφει τα στοιχεία του συστήματος

- OVAL Results Schema
  XML αρχείο που καταγράφει τα αποτελέσματα

Ένα θετικό στοιχείο του OVAL είναι πως η δομή του, επιτρέπει να χρησιμοποιηθεί ως ένα κομμάτι ενός συνόλου από πρότυπα και τεχνολογίες ασφάλειας. Η γενική περίπτωση χρήσης του OVAL αποτελείται από πέντε (5) βασικά βήματα [28], όπως φαίνονται και στην εικόνα 3.1.

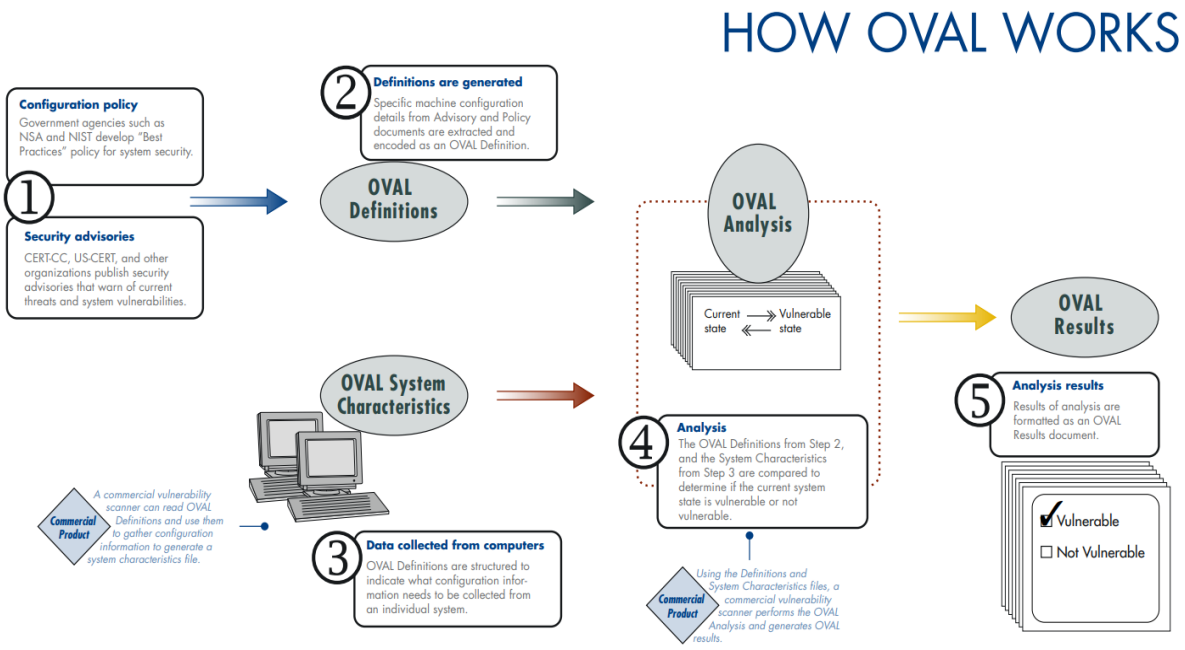
1. Γνωστοί μεγάλοι κρατικοί και μη, οργανισμοί δημοσιοποιούν ιδανικές πολιτικές και πρακτικές για την ασφάλεια συστημάτων, σε συνδυασμό με οργανισμούς που δημοσιοποιούν πληροφορίες σχετικά με γνωστές τρωτότητες και ευπάθειες συστημάτων, καθώς και πιθανούς τρόπους επιθέσεων από κακόβουλους χρήστες.

2. Οι πληροφορίες που δημοσιοποιούνται σχετικά με τις πρακτικές και πολιτικές ασφαλείας καθώς και για τις πιθανές απειλές που αναφέρθηκαν στο 1ο βήμα, κωδικοποιούνται ως αρχεία OVAL.

3. Τα αρχεία OVAL μορφοποιούνται ώστε να ορίζουν το ποιες πληροφορίες για το εξεταζόμενο σύστημα πρέπει να συμπεριληφθούν εξεταστούν.

4. Οι πληροφορίες από τα βήματα 2 και 3, αναλύονται και συγκρίνονται για να κρίνουν αν τη κατάσταση του συστήματος που εξετάζεται.

5. Τα αποτελέσματα της ανάλυσης του συστήματος καταγράφονται σε ένα αρχείο OVAL που περιέχει όλες τις πληροφορίες της ανάλυσης



Εικόνα 2.8. Παράδειγμα λειτουργίας OVAL [28]

Βέβαια εκτός από την τη περίπτωση χρήσης της εικόνας 3.5, το OVAL χρησιμοποιείται και σε ορισμένες άλλες περιπτώσεις.

Κάποιες από τις βασικότερες περιπτώσεις χρήσης [28] του OVAL είναι:

- Διανομή εγγράφων ενημέρωσης ασφάλειας. Τέτοια έγγραφα συνήθως δημοσιοποιούνται από ερευνητές ή οργανισμούς στο χώρο της ασφάλειας. Αυτά τα έγγραφα περιγράφουν ευπάθειες, τρωτότητες, πιθανούς κινδύνους και άλλες πληροφορίες σχετικά με μια τεχνολογία, με σκοπό την ενημέρωση και προστασία από τις απειλές αυτές. Ένα

χαρακτηριστικό παράδειγμα χρήσης είναι, όταν κάποιος εφόσον καταγράψει μία γνωστή ευπάθεια και τη δημοσιοποιήσει στο CVE, μπορεί να αξιοποιήσει τις είδη υπάρχουσες πληροφορίες που αναγράφονται στο CVE για να δημιουργήσει ένα σχετικό OVAL αρχείο που θα περιγράφει την ευπάθεια, και θα μπορεί να χρησιμοποιηθεί από άλλους χρήστες μέσω του OVAL.

- Δημιουργία ενός vulnerability assessment. Τέτοια assessments αναλύουν σε βάθος ένα σύστημα ή μια υπηρεσία, με σκοπό να ανακαλύψουν πιθανές ευπάθειες, τρόπους παραβίασης ή άλλα λειτουργικά προβλήματα που μπορούν να έχουν αντίκτυπο στην ασφάλεια του συστήματος αυτού. Η διαδικασία πραγματοποιείται ελέγχοντας το σύστημα με διάφορα εργαλεία. Χαρακτηριστικά παραδείγματα περιπτώσεων χρήσης του OVAL είναι:

    - Δημοσίευση των αποτελεσμάτων του assessment σε μορφή OVAL. Με αυτό το τρόπο το assessment θα μπορεί να αξιοποιηθεί ευκολότερα από άλλους χρήστες.
    - Συνεργασία κατά τη διάρκεια ενός assessment. Συχνά ένα assessment πραγματοποιείται από μια ομάδα μηχανικών ασφαλείας. Σε τέτοιες περιπτώσεις είναι χρήσιμο να υπάρχει ένα κοινό πρότυπο καταγραφής των αποτελεσμάτων. Το OVAL σε αυτή τη περίπτωση αποδεικνύεται ιδιαίτερα χρήσιμο.

- Διαχείριση ενημερώσεων λογισμικού. Οι ενημερώσεις λογισμικού μπορούν να επηρεάσουν άμεσα την ασφάλεια ενός συστήματος. Μια κακόβουλη ενέργεια ή επίθεση μπορεί να εκμεταλλευτεί κάποια ευπάθεια ή κενό ασφάλειας του λογισμικού και να τη χρησιμοποιήσει ώστε να βλάψει και άλλες υπηρεσίες του συστήματος. Συχνά οι προμηθευτές του λογισμικού διανέμουν νέες εκδόσεις που λύνουν τα γνωστά προβλήματα ασφαλείας και όχι μόνο που μπορεί να έχει ένα λογισμικό. Σε αυτή περίπτωση είναι κρίσιμο το σύστημα να περιέχει λογισμικό που είναι πάντα ενημερωμένο, ειδικά όταν οι ενημερώσεις αφορούν θέματα ασφάλειας. Χαρακτηριστικό παράδειγμα χρήσης του OVAL, είναι ο η χρήση της αρχειοθήκης του OVAL για τον έλεγχο των ενημερώσεων του συστήματος και τη σύγκριση τους με πιθανές γνωστές προβληματικές εκδόσεις λογισμικού που υπάρχουν ακόμα στο σύστημα.

- Διαχείριση ρυθμίσεων ενός συστήματος. Ομοίως με τις ενημερώσεις, οι ρυθμίσεις του συστήματος και του λογισμικού που χρησιμοποιεί είναι ζωτικής σημασίας για την ασφάλεια του. Γνωστοί κρατικοί οργανισμοί, δημοσιεύουν πρακτικές και οδηγίες για τη σωστή ρύθμιση και παραμετροποίηση του συστήματος. Ακολουθώντας αυτές τις πρακτικές βελτιώνεται σημαντικά η ασφάλεια του συστήματος. Το OVAL μπορεί να χρησιμοποιηθεί για να αναλύσει τις ρυθμίσεις του συστήματος και να συγκρίνει με τις προτεινόμενες ή τις βέλτιστες οδηγίες που υπάρχουν στο repository και έχουν δημοσιοποιηθεί από έγκριτους οργανισμούς στο χώρο της ασφάλειας..

Κεφάλαιο 3ο: Στοίβα υπηρεσιών ELK

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει περιγραφή, της στοίβας υπηρεσιών ELK. Αρχικά θα γίνει η παρουσίαση γενικών πληροφοριών σχετικά με την στοίβα, με τις υπηρεσίες που την απαρτίζουν καθώς και άλλες γενικές πληροφορίες. Ύστερα θα μελετηθούν αναλυτικά όλες οι υπηρεσίες της στοίβας. Τέλος θα παρουσιαστεί η αρχιτεκτονική της στοίβας ELK, καθώς και το πως αυτές οι υπηρεσίες συνδέονται λειτουργικά και προσφέρουν έναν ολοκληρωμένο μηχανισμό.

3.2 Στοίβα ELK

Η στοίβα ELK υλοποιήθηκε από την εταιρία Elastic. Η εταιρία αυτή ιδρύθηκε το 2012 και από τότε δραστηριοποιείται κυρίως στον χώρο της συλλογής, ανάλυσης και της ασφάλειας δεδομένων [21]. Το αρχικό σημαντικό project της εταιρίας ήταν το λογισμικό Elasticsearch το οποίο χρησιμοποιείται ως μηχανισμός αναζήτησης και ανάλυσης δεδομένων σε πραγματικό χρόνο και ήταν αυτό που καθόρισε και την ονομασία της εταιρίας, η οποία αρχικά ήταν Elasticsearch inc. και το 2015 άλλαξε σε Elastic. Το λογισμικό ή η υπηρεσία αυτή ανήκει στην στοίβα ELK και θα εξεταστεί η λειτουργία της πιο αναλυτικά, παρακάτω. Συνολικά η στοίβα ELK απαρτίζεται από τρεις υπηρεσίες. Η ονομασία ELK αποτελείται από τα αρχικά γράμματα των υπηρεσιών που την απαρτίζουν. Συγκεκριμένα οι υπηρεσίες αυτές είναι [21]:

- (E) Elasticsearch, η οποία όπως προαναφέρθηκε, χρησιμοποιείται στην στοίβα ως μηχανισμός αναζήτησης και ανάλυσης δεδομένων.
- (L) Logstash, η οποία αποτελεί μια υπηρεσία συλλογής δεδομένων, διαφόρων μορφών, από πολλαπλές πηγές.

- (K) Kibana, χρησιμοποιείται για την οπτικοποίηση των δεδομένων που συλλέγονται και επεξεργάζονται από τις πρώτες δύο υπηρεσίες.

Είναι σημαντικό να αναφερθεί πως και οι τρεις υπηρεσίες της στοίβας ELK είναι ανοιχτού κώδικα. Αυτό σημαίνει πως είναι δωρεάν η χρήση τους, υπό τους όρους που ορίζει η άδεια με την οποία διανέμεται ο κώδικας. Επίσης το γεγονός, ότι είναι ανοιχτού κώδικα δίνει την δυνατότητα σε χρήστες να προσαρμόσουν τις υπηρεσίες προς τις ανάγκες τους, να τις τροποποιήσουν με βάση τις ανάγκες τους ή και να προτείνουν βελτιώσεις και αλλαγές στις υπάρχουσες υπηρεσίες [21].

Ακόμη, πρέπει να σημειωθεί πως αν και η στοίβα ELK αποτελείται από τρεις βασικές υπηρεσίες, πολλές φορές η αναφορά στη στοίβα ELK παραπέμπει και στην χρήση μιας τέταρτης υπηρεσίας που ονομάζεται Beats [21], και λειτουργεί σε συνδυασμό με την υπηρεσία Logstash, και συμβάλει στη συλλογή δεδομένων από διάφορες πηγές. Η υπηρεσία Beats είναι και αυτή υλοποιημένη από την εταιρία Elastic, και είναι και αυτή ανοιχτού κώδικα όπως και οι υπόλοιπες υπηρεσίες της στοίβας.

3.3 Elasticsearch

Η υπηρεσία Elasticsearch [21], αποτελεί έναν μηχανισμό αναζήτησης, αποθήκευσης, ανάλυσης και διαχείρισης δεδομένων. Είναι μια RESTful υπηρεσία που βασίζεται σε HTTP requests για να μεταφέρει τα ζητούμενα δεδομένα. Το λογισμικό αυτό είναι υλοποιημένο στη γλώσσα προγραμματισμού Java [30], και στη βάση του χρησιμοποιεί εκτενώς τη βιβλιοθήκη ανοιχτού κώδικα Apache Lucene [30] που με τη σειρά της αποτελεί έναν πιο απλό μηχανισμό αναζήτησης, πάνω στον οποίο βασίζεται η υπηρεσία Elasticsearch και προσθέτοντας επιπλέον λειτουργίες, προσφέρει μια πιο ολοκληρωμένη υπηρεσία με περισσότερες δυνατότητες. Χαρακτηρίζεται συχνά ως μια NoSQL βάση δεδομένων που αποθηκεύει τα δεδομένα σε μορφή αρχείων (documents).

Η υπηρεσία Elasticsearch, εμφανίζεται όλο και πιο συχνά σε σύνθετα και περίπλοκα πληροφοριακά συστήματα. Το γεγονός αυτό οφείλεται στα πλεονεκτήματα της.

Τα πιο σημαντικά πλεονεκτήματα της Elasticsearch είναι [18][21]:

- Ταχύτητα. Λόγω του σχεδιασμού της υπηρεσίας, την χρήση της Apache Lucene, του ειδικού μηχανισμού ευρετηρίασης (indexing) που διαθέτει η Elasticsearch, δίνει την δυνατότητα, αναζήτησης δεδομένων σχεδόν σε πραγματικό χρόνο. Αυτό την καθιστά καλή επιλογή ως μηχανισμό αναζήτησης και διαχείρισης δεδομένων σε συστήματα όπου τα χρονικά περιθώρια είναι ιδιαίτερα σημαντικά. Ένα χαρακτηριστικό παράδειγμα είναι τα συστήματα επίβλεψης σε , τα οποία λαμβάνουν, επεξεργάζονται και στέλνουν δεδομένα σε πραγματικό χρόνο.

- Η δυνατότητα κατανομής φόρτου. Η αρχιτεκτονική της Elasticsearch, επιτρέπει την κατανομή φόρτου ανάλογα με τις ανάγκες του κάθε συστήματος. Η ανώτατη οντότητα στην ιεραρχία είναι το Cluster (συστάδα). Κάθε συστάδα περιέχει κόμβους οι οποίοι με την σειρά τους περιέχουν δείκτες και αρχεία τα οποία μπορεί να βρίσκονται σε διαφορετικά μηχανήματα και με τέτοιο τρόπο ανακατανέμουν το φόρτο του συστήματος σε περιπτώσεις που ο ρυθμός ανάκτησης η εγγραφής δεδομένων είναι μεγάλος. Συνήθως τα Cluster

αποτελούνται από αρκετούς υπολογιστές, ενώ πολλές φορές σε περίπλοκα συστήματα υπάρχουν πολλαπλά Cluster από υπολογιστές που ανακατανέμουν τον φόρτο δεδομένων μεταξύ τους. [i]

- Επεκτασιμότητα. Εκτός από τη δυνατότητα κατανομής φόρτου, η αρχιτεκτονική της Elasticsearch επιτρέπει την ομαλή επέκταση της ίδιας της υπηρεσίας, σε περιπτώσεις που οι ανάγκες ενός συστήματος αυξηθούν. Σε τέτοιες περιπτώσεις η προσθήκη νέων κόμβων και Cluster είναι ομαλή και δεν προκαλεί προβλήματα στην τρέχουσα λειτουργικότητα και διαθεσιμότητα της υπηρεσίας. Προφανώς το ίδιο ισχύει και σε περιπτώσεις που υπάρχει η ανάγκη μείωσης των πόρων του συστήματος.

- Η πληθώρα επιπρόσθετων λειτουργιών [21]. Πέρα από τα πλεονεκτήματα που αναφέρθηκαν προηγουμένως, η Elasticsearch διαθέτει διάφορες επιπλέον λειτουργίες. Οι πιο σημαντικές επιπρόσθετες λειτουργίες είναι:

    - Λειτουργίες ασφάλειας. Μερικές εκ των οποίων αποτελούν λειτουργίες φιλτραρίσματος IP διευθύνσεων, κρυπτογράφηση επικοινωνίας, υποστήριξης υπηρεσιών ασφάλειας τρίτων, υποστήριξης προτύπου GDPR.

    - Λειτουργίες διαχείρισης: Δυνατότητα διαχείρισης χρηστών και ρόλων, εργαλεία CLI, λειτουργία πολλαπλών snapshot που δίνει τη δυνατότητα να γίνει επαναφορά της υπηρεσίας στο επιθυμητό σημείο..

    - Λειτουργίες alerting. Προσφέρεται ειδικό περιβάλλον χρήστη (UI) για τη διαχείριση alerts, επίσης υποστηρίζεται η αποστολή ειδοποιήσεων με διάφορους τρόπους όπως e-mail, Jira, webhooks.

    - Λειτουργίες deployment. Οι λειτουργίες αυτές βοηθούν στην ανάπτυξη της υπηρεσίας σε διαφορετικά περιβάλλοντα όπως Docker containers, Elastic Cloud και φυσικά τοπικά σε υπολογιστές διαφόρων λειτουργικών συστημάτων.

    - Λειτουργίες client. Αποτελούν λειτουργίες με της οποίες ένας client μπορεί να έχει πρόσβαση και να αλληλοεπιδρά με την Elasticsearch, όπως JDBC client, ODBC client, Elasticsearch SQL queries..

    - Λειτουργίες monitoring. Η Elasticsearch διαθέτει λειτουργίες, αυτόματου monitoring, πλήρους monitoring ενός Cluster ή και multi-stack monitoring σε πολλαπλά Cluster.

    Καθώς η Elasticsearch είναι υπηρεσία ανοιχτού κώδικα, υπάρχουν επιπλέον λειτουργίες για την Elasticsearch υλοποιημένες από τρίτους..

- Η Elasticsearch αποτελεί κομμάτι ενός μεγαλύτερου οικοσυστήματος υπηρεσιών οι οποίες είναι σχεδιασμένες για να λειτουργούν σε συνδυασμό δημιουργώντας έναν σύνθετο

μηχανισμό [14]. Ένα τέτοιο παράδειγμα είναι και η στοίβα υπηρεσιών ELK, όπου αν και κάθε υπηρεσία μπορεί να λειτουργήσει και ξεχωριστά, η δυνατότητα συνδυασμού υπηρεσιών αποτελεί αναμφισβήτητο πλεονέκτημα στη χρήση τους.

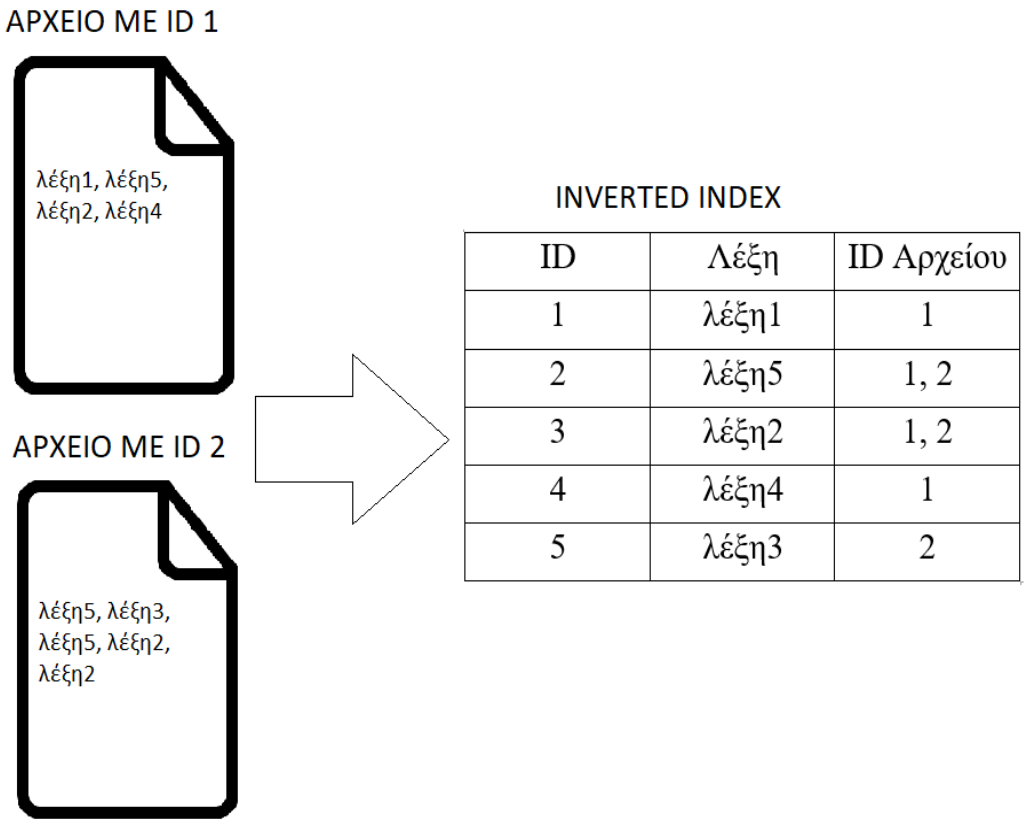
Όπως φαίνεται, η υπηρεσία Elasticsearch είναι ιδιαίτερα σύνθετη, με πολλαπλές λειτουργίες και λεπτομέρειες. Επίσης είναι μια υπηρεσία που εξελίσσεται συνέχεια [21] και προσθέτει νέες λειτουργίες ή αλλάζει τις υπάρχουσες καθώς υπάρχει ευρύ κοινό που τη χρησιμοποιεί. Παρόλα αυτά η Elasticsearch έχει έναν σταθερό πυρήνα στη βάση της πάνω στον οποίο χτίζονται οι επιπρόσθετες λειτουργίες. Παρακάτω αναλύονται τα βασικά στοιχεία του πυρήνα σε επίπεδο λογικού δείκτη (index) και σε επίπεδο υποδομής (infrastructure).

Μια από τις βασικότερες έννοιες στη δομή της Elasticsearch, είναι οι δείκτες [18][21]. Ιεραρχικά είναι η πρώτη οντότητα που δημιουργείται με σκοπό την αναπαράσταση και διαχείριση πληροφοριών. Οι δείκτες συνδέουν λογικά τις επιμέρους οντότητες που περιλαμβάνουν.

Οι οντότητες αυτές είναι:

- Documents (αρχεία). Τις αναφέρθηκε νωρίτερα, τα δεδομένα αποθηκεύονται ως αρχεία τα οποία ακολουθούν τη μορφή JSON. Αναλογικά με μια παραδοσιακή βάση δεδομένων τα αρχεία θα μπορούσαν να χαρακτηριστούν ως οι γραμμές τις Table, που περιγράφει μια συγκεκριμένη οντότητα. Προφανώς, ένα document μπορεί να περιλαμβάνει διαφορετικές μορφές δεδομένων τις αριθμούς, χαρακτήρες, ημερομηνίες. Κάθε document έχει ένα μοναδικό id που το ξεχωρίζει από τα υπόλοιπα documents, και έναν τύπο ανάλογα με την οντότητα την οποία αναπαριστά.

- Inverted Indices. Τα Inverted indices ουσιαστικά είναι instances απο Lucene indices. Κάθε inverted index είναι τις αυτόνομος μηχανισμός αναζήτησης. Είναι μια δομή δεδομένων που συνδέει τα δεδομένα με τα αρχεία που τα περιέχουν, επιταχύνοντας έτσι την αναζήτηση τις. Ο τρόπος υλοποίησης τις δομής τις θυμίζει πολύ πίνακες κατακερματισμού hash-maps, οι οποίοι αποτελούν μια δομή δεδομένων που χρησιμοποιείται εκτενώς και περιλαμβάνεται τις βασικές βιβλιοθήκες των περισσότερων γλωσσών προγραμματισμού. Στο παρακάτω σχήμα φαίνεται ένα απλουστευμένο παράδειγμα τις λειτουργίας των inverted indices..



Εικόνα 3.1. Παράδειγμα λειτουργίας Inverted Index

Τις φαίνεται στο σχήμα, η δομή καταχωρεί τις λέξεις και τις συνδέει με τα μοναδικά id, των documents τα οποία τις περιλαμβάνουν. Στο πραγματικό περιβάλλον τις Elasticsearch, χρησιμοποιούνται επιπλέον παράμετροι, τις για παράδειγμα η συχνότητα των λέξεων. Οι επιπλέον παράμετροι χρησιμοποιούνται κυρίως για να βελτιώσουν την επίδοση του μηχανισμού αναζήτησης.

Shards. Τα shards, τις προκύπτει και από την ονομασία τις, αποτελούν μικρότερα κομμάτια τις λογικού index. Στην πραγματικότητα κάθε shard περιλαμβάνει ένα inverted index (Lucene index). Ένα ερώτημα που προκύπτει είναι, γιατί χρειάζονται και shards και τα inverted indexes ως δύο ξεχωριστές δομές; Η απάντηση έχει να κάνει με θέματα σταθερότητας τις υπηρεσίας Elasticsearch και με θέματα επεκτασιμότητας. Ένα inverted index, κατά την δημιουργία του τις φορές αποθηκεύεται σε περισσότερα από ένα shards, τα οποία ονομάζονται replicas. Τα shards μπορούν να κατανεμηθούν σε διαφορετικούς κόμβους σε ένα Cluster, και η ύπαρξη περισσότερων από τις αντιγράφου κάθε inverted index, προσφέρει μεγαλύτερη αξιοπιστία όσον αφορά την διαθεσιμότητα τις υπηρεσίας σε περίπτωση που κάποιος κόμβος σταματήσει να λειτουργεί.

Εκτός από τις έννοιες που αναλύθηκαν προηγουμένως, υπάρχουν και ορισμένες έννοιες που οργανώνουν και περιγράφουν την υπηρεσία σε επίπεδο υποδομής [18][21].

Οι έννοιες αυτές είναι:

- Κόμβοι (nodes). Ένας κόμβος αποτελεί μια διεργασία Elasticsearch που εκτελείται σε ένα μηχάνημα. Η εκκίνηση μίας διεργασίας σημαίνει την δημιουργία του αντίστοιχου κόμβου. Σε ένα μηχάνημα μπορούν να τρέχουν περισσότερες από μία (1) διεργασίες Elasticsearch, κάτι που σημαίνει πως ένα μηχάνημα μπορεί να έχει περισσότερους από έναν (1) ενεργούς κόμβους. Εσωτερικά η Elasticsearch περιγράφει τους κόμβους ως δομές με συγκεκριμένα πεδία, τα οποία καθορίζουν και τη λειτουργία τους. Συγκεκριμένα το πεδίο που καθορίζει το μεγαλύτερο λειτουργικό κομμάτι ενός κόμβου είναι το πεδίο node roles. Το πεδίο αυτό περιγράφει τον ρόλο του κόμβου μέσα στο Cluster το οποίο ανήκει.

  Τα δύο (2) βασικότερα είδη κόμβων που πρέπει να υπάρχουν σε κάθε Cluster είναι:

- Master node. Αυτό το είδος κόμβου διαχειρίζεται ολόκληρο το Cluster στο οποίο ανήκει. Δημιουργεί λογικούς δείκτες, κρατάει πληροφορίες σχετικά με τους υπόλοιπους κόμβους του Cluster, καθορίζει σε ποιόν κόμβο θα τοποθετηθεί κάθε shard. Οι Master κόμβοι καθορίζονται από την διαδικασία εκλογών που γίνεται μέσα στο Cluster, και στην οποία συμμετέχουν όσοι κόμβοι περιλαμβάνουν τη τιμή master στο πεδίο node roles.

- Data node. Αυτό το είδος κόμβου αναλαμβάνει την διαχείριση δεδομένων. Εκτελεί διαδικασίες CRUD (Create, Read, Update, Delete), δεδομένων και αναζήτησης δεδομένων. Η κατηγορία αυτή έχει υποκατηγορίες ανάλογα με τις ανάγκες του κάθε συστήματος και την απαιτούμενη λειτουργία που θα έχει ο κόμβος. Για παράδειγμα η τιμή data_hot σε αντίθεση με απλό data θα δημιουργήσει έναν κόμβο με περισσότερους πόρους που θα διαχειρίζεται ταχύτερα διαδικασίες εγγραφής και διαβάσματος πληροφοριών.

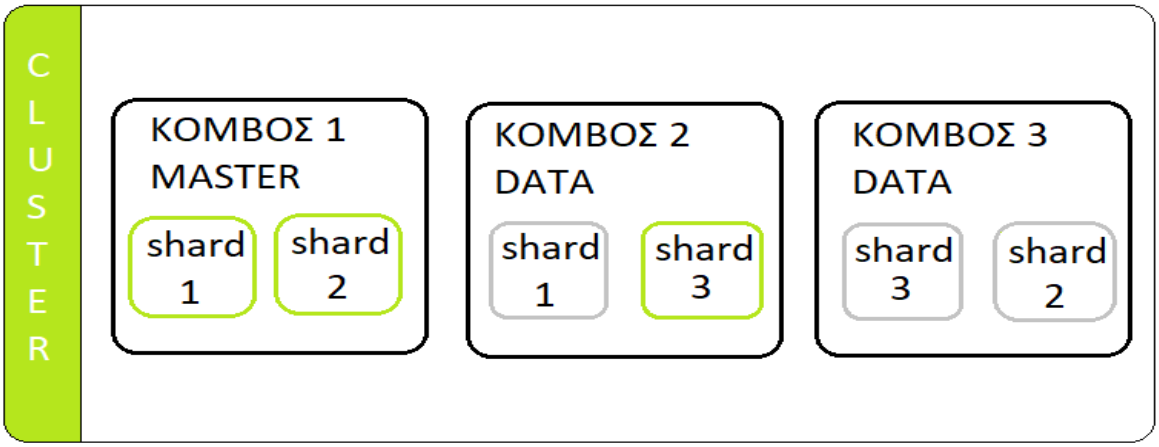
  Εκτός από αυτά τα είδη κόμβων υπάρχουν και άλλα, που χρησιμοποιούνται σε συγκεκριμένες περιστάσεις. Τέτοιοι κόμβοι είναι:

  - Machine learning node, που είναι κόμβοι που υποστηρίζουν λειτουργίες μηχανικής μάθησης.

  - Transform node, οι οποίοι χειρίζονται Transform API requests.

  - Remote-eligible node, οι οποίοι είναι κόμβοι που μπορούν να συνδεθούν σε περισσότερα από ένα Cluster. Πολλές φορές χρησιμοποιούνται για να μεταφέρουν δεδομένα μεταξύ Clusters.

- Clusters (συστάδες). Ένα Cluster, αποτελείται από ένα σύνολο κόμβων. Αρχικά ένα Cluster δημιουργείται με τη δημιουργία ενός κόμβου. Ο πρώτος κόμβος που δημιουργείται είναι ταυτόχρονα και Master κόμβος και Data node, καθώς είναι ο μοναδικός. Κατά τη δημιουργία επιπλέον κόμβων μπορούν να ρυθμιστούν κατάλληλα ώστε να υπάρχει ισορροπία στο εσωτερικό του Cluster. Προφανώς ανάλογα με τις ανάγκες του χρήστη μπορούν να δημιουργηθούν πολλαπλά Cluster με πολλαπλούς κόμβους στο κάθε Cluster.



Εικόνα 3.2 Παράδειγμα Elasticsearch Cluster

Η αρχιτεκτονική της υποδομής της Elasticsearch καθορίζεται από τον χρήστη. Υπάρχει η δυνατότητα ρύθμισης των περισσότερων παραμέτρων που αφορούν τους κόμβους και τα Cluster. Οι παράμετροι αυτοί καθορίζονται σε ένα ειδικό αρχείο Elasticsearch.yml που διαβάζεται κατά το ξεκίνημα μιας διεργασίας Elasticsearch.

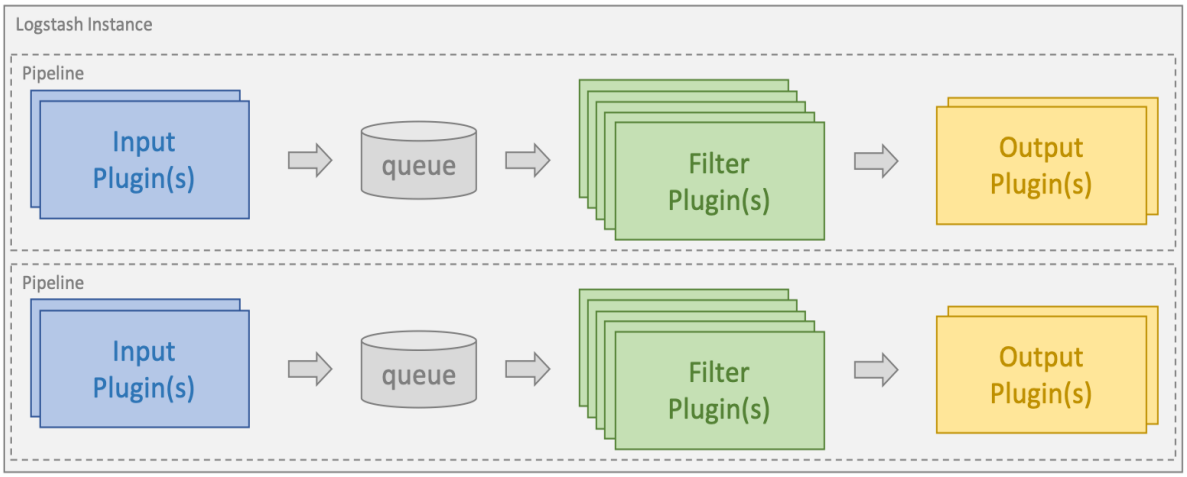
3.4 Logstash και Beats

Ο βασικός μηχανισμός διαχείρισης, αποθήκευσης και αναζήτησης δεδομένων για τη στοίβα ELK, είναι η υπηρεσία Elasticsearch [21]. Για να λειτουργήσει όμως ο μηχανισμός αυτός χρειάζεται πραγματικά δεδομένα. Η υπηρεσία η οποία προωθεί δεδομένα στην Elasticsearch, είναι η υπηρεσία Logstash, η

οποία με τη σειρά της λαμβάνει δεδομένα από διάφορες πηγές. Μια από τις βασικότερες πηγές λήψης δεδομένων για μια στοίβα ELK είναι τα Beats [21].

3.4.1 Logstash

Η υπηρεσία Logstash [21] αποτελεί λογισμικό ανοιχτού κώδικα, γραμμένο στις γλώσσες προγραμματισμού Java και jRuby [30] . Στη λειτουργία της είναι ένα pipeline δεδομένων. Δηλαδή λαμβάνει δεδομένα από διάφορες πηγές, τα φιλτράρει ή τα τροποποιεί και τα προωθεί προς συγκεκριμένες πηγές ανάλογα με τις ανάγκες του χρήστη.



Εικόνα 3.3. Παράδειγμα Logstash αρχιτεκτονικής [21]

Όπως φαίνεται στο σχήμα σε πρώτο στάδιο η Logstash δέχεται δεδομένα από πηγές τα οποία αφού μπουν σε ουρά, προωθούνται στα φίλτρα. Οι πηγές μπορεί να είναι διαφορετικού είδους και πολλές ταυτόχρονα. Οι πιο συνήθεις πηγές εισόδου δεδομένων είναι [21]:

- Beats. Τα Beats είναι πράκτορες (agents) που χωρίζονται σε συγκεκριμένες κατηγορίες ανάλογα με το είδος δεδομένων που στέλνουν. Θα αναλυθούν εκτενώς στο κεφάλαιο 4.4.2.

- Αρχεία. Η Logstash μπορεί να δεχτεί δεδομένα από διάφορα αρχεία είτε του συστήματος, είτε από διάφορα προγράμματα, είτε αρχεία που δημιουργεί ο χρήστης.

- HTTP & HTTPS requests. Δεδομένα και γεγονότα μέσω ορισμένων και ρυθμισμένων HTTP ή HTTPS requests

- Syslogs. Η Logstash παρέχει τη δυνατότητα εισόδου δεδομένων απο την port 514, τα οποία επεξεργάζεται και φιλτράρει σύμφωνα με το πρότυπο RFC3164

- Redis server. Το Redis χρησιμοποιείται σε πολλά συστήματα ως κατανεμημένη βάση δεδομένων. Η logstash μπορεί να δέχεται δεδομένα από έναν διακοσμητή redis.

Όπως προαναφέρθηκε, αφού η Logstash λάβει τα δεδομένα τα τοποθετεί σε μια ουρά, και ο χρόνος τον οποίο θα αναμένουν επεξεργασίας, εξαρτάται από τους πόρους τους οποίους διαθέτει το σύστημα και το πόσα δεδομένα μπορούν να φιλτραριστούν ταυτόχρονα.

Τα φίλτρα είναι ο μηχανισμός επεξεργασίας δεδομένων της Logstash. Μπορούν να χρησιμοποιηθούν σε συνδυασμούς μεταξύ τους ή να χρησιμοποιηθούν υπό συγκεκριμένες προϋποθέσεις τις οποίες πρέπει να πληρούν τα δεδομένα. Τα ποιο σημαντικά είδη φίλτρων είναι:

- Φίλτρα drop. Τα φίλτρα αυτά διαγράφουν και δεν προωθούν προς την έξοδο τα δεδομένα. Συνήθως χρησιμοποιούνται για να μη συμπεριλάβουν δεδομένα από debugging.

- Φίλτρα clone. Δημιουργούν αντίγραφα των δεδομένων, συχνά προσθέτοντας η αφαιρώντας πεδία από δεδομένα ανάλογα με τις ανάγκες του χρήστη.

- Φίλτρα mutate. Τροποποιούν και επεξεργάζονται τα πεδία στα δεδομένα, χωρίς να κρατάνε αντίγραφο της πηγής.

- Φίλτρα grok. Τα φίλτρα grok, δέχονται δεδομένα οποιασδήποτε μορφής και με τη χρήση regular expressions, μετατρέπουν τα δεδομένα στην επιθυμητή μορφή, για παράδειγμα σε εγγραφές postgresql, HTTP logs, Maven logs κτλ.

- Φίλτρα geoip. Προσθέτουν πληροφορίες σχετικά με γεωγραφικές τοποθεσίες IP διευθύνσεων.

Μετά το φιλτράρισμα, το τελευταίο στάδιο είναι τα δεδομένα να μεταφερθούν στην έξοδο του pipeline. Όπως και με τις πηγές εισόδου δεδομένων, έτσι και οι έξοδοι ποικίλουν.

Κάποιες από τις πιο συχνές εξόδους δεδομένων είναι [21]:

- Elasticsearch. Όπως και στη στοίβα ELK, έτσι και σε πολλές άλλες αρχιτεκτονικές που περιλαμβάνουν την υπηρεσία Elasticsearch, αυτή αποτελεί την έξοδο δεδομένων της

Logstash. Εκεί τα δεδομένα αποθηκεύονται, κατανέμονται και επεξεργάζονται με τις διαδικασίες που αναλύθηκαν στην ενότητα 4.3.

- Αρχεία. Συχνά η έξοδος της Logstash μπορεί να είναι αρχεία σε ένα δίσκο, τα οποία με τη σειρά τους μπορούν να χρησιμοποιηθούν από άλλες υπηρεσίες η μηχανισμούς.

- E-mail. Η Logstash παρέχει τη δυνατότητα αποστολής των δεδομένων εξόδου σε ορισμένη από το χρήστη διεύθυνση ηλεκτρονικού ταχυδρομείου.

- statsd. Το statsd είναι μία υπηρεσία συστήματος που δέχεται δεδομένα μέσω πακέτων UDP, και τα προωθεί σε διάφορες συνδεδεμένες συσκευές στο σύστημα.

Μια σημαντική επιπλέον λειτουργία της Logstash, είναι τα codecs, τα οποία είναι φίλτρα ροής δεδομένων. Σε αντίθεση με τα φίλτρα της pipeline, φιλτράρουν τα δεδομένα πριν την είσοδο και κατά την έξοδο τους. Ένα χαρακτηριστικό παράδειγμα χρήσης codecs, είναι το φιλτράρισμα δεδομένων εισόδου και εξόδου σε μορφή JSON για την ευκολότερη χρήση τους από την Elasticsearch η οποία επίσης χρησιμοποιεί τη μορφή JSON για την οργάνωση των αρχείων της.

Όλες οι παράμετροι που αναφέρθηκαν προηγουμένως καθορίζονται από τις ρυθμίσεις της Logstash. Οι ρυθμίσεις συνήθως αποθηκεύονται σε .conf αρχεία, και κατά την εκκίνηση της υπηρεσίας Logstash, καθορίζεται το αρχείο ρυθμίσεων που θα χρησιμοποιηθεί. Στο σχήμα φαίνεται η μορφή που έχει ένα αρχείο ρυθμίσεων Logstash.

3.4.2 Beats

Τα Beats είναι λογισμικό ανοιχτού κώδικα, υλοποιημένα στη γλώσσα προγραμματισμού Go [30], και χρησιμοποιούνται ως πράκτορες (agents) σε συστήματα[21], από τα οποία στέλνουν δεδομένα σχετικά με διάφορα συμβάντα στο σύστημα το οποίο επιβλέπουν. Ο λόγος που χρησιμοποιείται πληθυντικός στην ονομασία Beats είναι η ύπαρξη διαφορετικών Beats ανάλογα με το σύστημα αλλά και το στόχο επίβλεψης σε ένα σύστημα. Η βασική βιβλιοθήκη που χρησιμοποιείται από κάθε Beat είναι η libbeat [30].

Από εκεί και πέρα κάθε Beat διαφοροποιείται στη λειτουργία του αλλά και στο είδος δεδομένων που στέλνει. Τα βασικά Beats που υλοποιήθηκαν από την εταιρία Elastic είναι επτά (7) [21] και είναι τα εξής:

1. Auditbeat. Το Auditbeat είναι αποστολέας δεδομένων που λειτουργεί ως πράκτορας σε υπολογιστές ή servers, που χρησιμοποιούν λειτουργικό σύστημα βασισμένο σε Linux. Χρησιμοποιεί το Linux Audit Framework για να λαμβάνει πληροφορίες σχετικά με τα γεγονότα. Μπορεί να ρυθμιστεί ώστε να καταγράφει γεγονότα σχετικά με χρήστες και διεργασίες που τρέχουν στο σύστημα. Επίσης μπορεί να αντιλαμβάνεται αλλαγές σε

σημαντικά αρχεία του συστήματος, όπως για παράδειγμα σε binaries ή σε διάφορα αρχεία ρυθμίσεων.

2. Filebeat. Το Filebeat σε αντίθεση με το Auditbeat λαμβάνει πληροφορίες σχετικά με γεγονότα που αφορούν μόνο αρχεία και όχι διεργασίες του συστήματος. Είναι πιο απλό στη χρήση και τη ρύθμιση σε σχέση με το Auditbeat, καθώς απλά επιβλέπει συγκεκριμένα αρχεία που θα οριστούν από το χρήστη.

3. Packetbeat. Το Packetbeat είναι ένας πράκτορας που χρησιμοποιείται για την ανάλυση πακέτων επιπέδου εφαρμογής, ενός δικτύου σε πραγματικό χρόνο. Λειτουργεί λαμβάνοντας πακέτα μεταξύ σταθμών σε ένα δίκτυο, και εξάγοντας τα επιθυμητά πεδία από ένα πακέτο. Μπορεί να χρησιμοποιηθεί όσο για τη καταγραφή γεγονότων ασφαλείας, τόσο και για τον έλεγχο της σωστής λειτουργίας διάφορων εφαρμογών που μεταφέρουν δεδομένα μέσω δικτύου. Κάποια από τα ποιο γνωστά πρωτόκολλα επιπέδου εφαρμογής, που προς το παρόν υποστηρίζει το Packetbeat είναι:

    - DNS
    - DHCP (v4)
    - HTTP
    - NFS
    - TLS
    - ICMP (v4 & v6)

4. Winlogbeat. Το Winlogbeat, είναι πράκτορας για λειτουργικά συστήματα Windows. Γίνεται εγκατάσταση στο σύστημα ως service, και παρακολουθεί, φιλτράρει και στέλνει τα συμβάντα στη Logstash. Κάποια από τα συμβάντα τα οποία μπορεί να καταγράψει είναι:

    Συμβάντα εφαρμογών του συστήματος
    Συμβάντα ασφάλειας των Windows
    Συμβάντα διεργασιών του συστήματος
    Συμβάντα hardware συσκευών που αλληλοεπιδρούν με το σύστημα

5. Metricbeat. Το Metricbeat, είναι πράκτορας που συνήθως τοποθετείται σε servers, και επιβλέπει γεγονότα κρίσιμων εφαρμογών, όπως:

    - Apache
    - Redis
    - Docker
    - MySQL, PostgresDB, MongoDB

6. Functionbeat. Το Functionbeat είναι πράκτορας που τοποθετείται σε ένα περιβάλλον χωρίς servers και συλλέγει δεδομένα από cloud εφαρμογές

7. Heartbeat. Το Heartbeat χρησιμοποιείται για να διαβεβαιώσει ότι μια υπηρεσία ή ένας πόρος είναι διαθέσιμος, στέλνοντας πακέτα ανά συγκεκριμένα χρονικά διαστήματα που ορίζει ο χρήστης. Τα πρωτόκολλα που προς το παρόν υποστηρίζει είναι:

   ICMP (v4 & v6)
   TCP
   HTTP

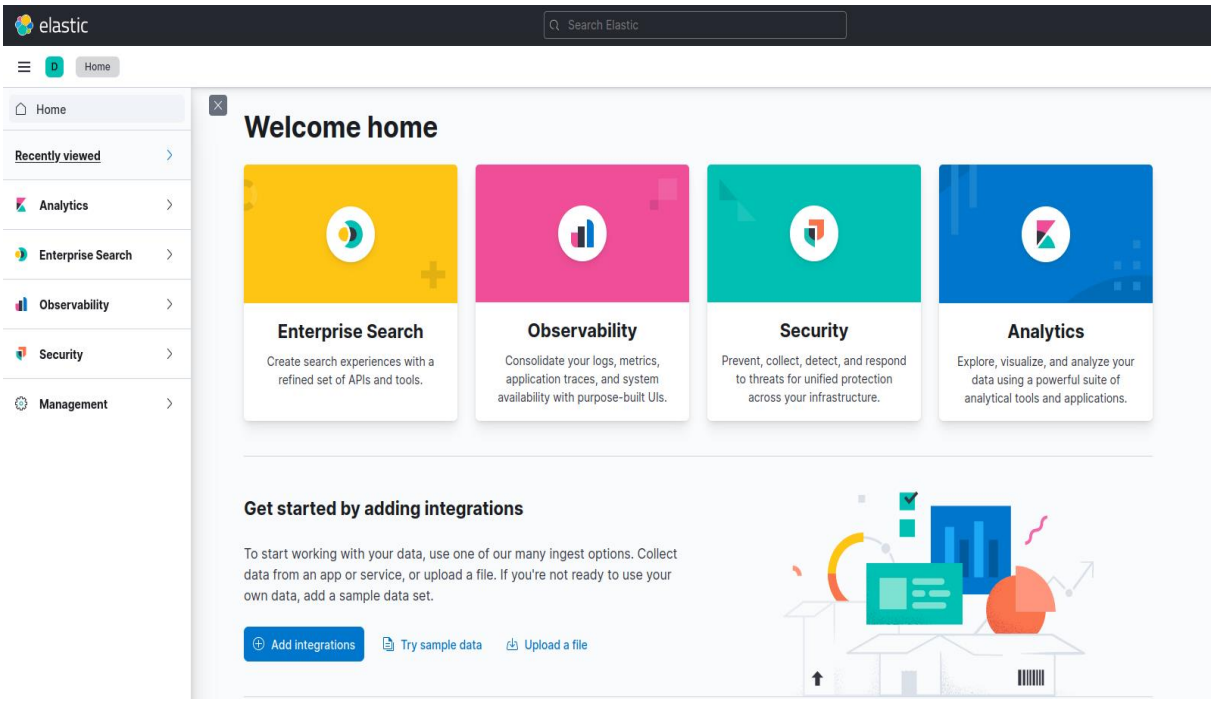
Όπως και όλη η στοίβα ELK έτσι και τα Beats είναι ανοιχτού κώδικα. Η βασική βιβλιοθήκη libbeat [30], πάνω στην οποία βασίζονται όλα τα Beats που υλοποίησε η Elastic, είναι και αυτή ανοιχτή και προσβάσιμη από όλους. Αυτό οδήγησε στη δημιουργία πολλών Beats, από τρίτους. Τα Beats αυτά στη βιβλιογραφία της Elastic ονομάζονται Community Beats [21].

Γνωστά Community Beats είναι:

- Dockbeat, το οποίο συλλέγει στατιστικά σχετικά με Docker containers.
- Githubbeat, το οποίο καταγράφει τα συμβάντα για τα επιλεγμένα από το χρήστη, github repositories.
- Nginxbeats, παρόμοια με τα Heartbeats, συλλέγει πληροφορίες σχετικά με τη κατάσταση ενός Nginx διακοσμητή.
- Twitterbeat, το οποίο καταγράφει τα tweets, για τα επιλεγμένα απο το χρήστη ονόματα χρηστών.

3.5 Kibana

Το Kibana είναι η τρίτη και τελευταία υπηρεσία της στοίβας ELK [21]. Είναι λογισμικό ανοιχτού κώδικα υλοποιημένο από την Elastic, στη γλώσσα προγραμματισμού Typescript [30].Το Kibana, δίνει μορφή σε όλα τα δεδομένα της στοίβας που συλλέγονται μέσω των Beats, φιλτράρονται μέσω της Logstash, και αποθηκεύονται στην Elasticsearch. Πιο απλοϊκά θα μπορούσε να χαρακτηριστεί ως μια web διεπαφή χρήστη που επιτρέπει την επισκόπηση δεδομένων με διαφορετικού τρόπους.



Εικόνα 3.4. Διεπαφή χρήστη Kibana [21]

Οι τρεις βασικές λειτουργίες του Kibana, είναι [21]:

1. Αναζήτηση και προεπισκόπηση δεδομένων. Το Kibana διαθέτει αρκετά εργαλεία για την αναζήτηση των δεδομένων μέσω της Elasticsearch και τη προεπισκόπηση τους. Τα εργαλεία που χρησιμοποιούνται συχνότερα είναι:

    - Kibana Lens. Αποτελεί το βασικό περιβάλλον του Kibana, για την προεπισκόπηση δεδομένων. Παρέχει τη δυνατότητα δημιουργίας, διαχείρισης και αποθήκευσης διάφορων πινάκων δεδομένων. Τα δεδομένα μπορεί να προέρχονται από διαφορετικές πηγές και να υπάρχουν πολλαπλοί πίνακες ταυτόχρονο στη διεπαφή του χρήστη, τους οποίους μπορεί να διαχειρίζεται.

    - TSVB (Time Series Visual Builder). Είναι εργαλείο που προσφέρει τη δυνατότητα οπτικοποίησης επιλεγμένων συμβάντων με κλίμακα το χρόνο. Δημιουργεί χρονοδιαγράμματα στα οποία εμφανίζονται τα επιλεγμένε συμβάντα και μέσω των οποίων φαίνεται το χρονικό περιθώριο τους καθώς και η συχνότητα τους.

    - Geospatial Analysis. Ένα σημαντικό πλεονέκτημα του Kibana, είναι ότι προσφέρει γεωγραφική ανάλυση και χάρτες σχετικά με τα ορισμένα καταγεγραμμένα συμβάντα.

- Διαγράμματα και Μετρήσεις. Το Kibana δίνει τη δυνατότητα αναπαράστασης των δεδομένων με μια πληθώρα από διαγράμματα ανάλογα με τις ανάγκες του χρήστη. Επίσης παρέχει μηχανισμό μετρήσεων συμβάντων και άλλων δεδομένων.

2. Ανάλυση των δεδομένων. Η ανάλυση των δεδομένων συνδέεται άμεσα με την επισκόπηση των δεδομένων. Όλα τα δεδομένα που συλλέγονται για την επισκόπηση τους, περνάνε απο το στάδιο της ανάλυσης. Οι περισσότεροι αλγόριθμοι ανάλυσης δεδομένων της Kibana είναι συνδεδεμένοι με γραφήματα, σχήματα και διαγράμματα.

   Παρόλα αυτά όμως το Kibana προσφέρει κάποια εργαλεία χειροκίνητης ανάλυσης από το χρήστη.

   Τα εργαλεία αυτά είναι:

   - KQL (Kibana Query Language). Η KQL αποτελεί μια γλώσσα αναζήτησης δεδομένων σχεδιασμένη για το Kibana. Είναι σχετικά απλή στη χρήση της και έχει αρκετά κοινά σημεία με την SQL.

   - Γράφους Ανάλυσης. Οι γράφοι ανάλυσης συνδέουν μεταξύ τους δεδομένα και συμβάντα, και μπορούν να αναλυθούν οπτικά από το χρήστη.

   - Dev κονσόλα εντολών. Το Kibana δίνει τη δυνατότητα δημιουργία requests τα οποία αποστέλλονται μέσω της dev κονσόλας στην Elasticsearch, η οποία επιστρέφει απαντήσεις στα requests οι οποίες αναλύονται απο το χρήστη.

3. Διαχείριση, επίβλεψη και ασφάλεια των δεδομένων. Όπως είναι προφανές το Kibana, έχει πρόσβαση στο μεγαλύτερο μέρος των δεδομένων της στοίβας ELK. Αυτό σημαίνει πως μόνο εξουσιοδοτημένα πρόσωπα θα πρέπει να έχουν πρόσβαση στα δεδομένα αυτά, κάποια από τα οποία μπορεί να είναι και εμπιστευτικά δεδομένα.

   Τα σημαντικότερα εργαλεία για την επίβλεψη και την ασφάλεια των δεδομένων είναι:

   - Κρυπτογράφηση της επικοινωνίας. Από τη στιγμή που το Kibana είναι web διεπαφή χρήστη, για τη διασφάλιση της εμπιστευτικότητας των δεδομένων χρησιμοποιούνται τα πρωτόκολλα SSL/TLS για τη κρυπτογράφηση της επικοινωνίας.

   - Ρόλοι χρηστών που καθορίζουν τη πρόσβαση. Το Kibana, δίνει τη δυνατότητα παραμετροποίησης των δικαιωμάτων του κάθε χρήστη καθώς και τη δημιουργία ομάδων χρηστών με συγκεκριμένα δικαιώματα. Τα δικαιώματα αυτά αφορούν δικαιώματα πρόσβασης στα δεδομένα καθώς και δικαιώματα επεξεργασίας και διαγραφής δεδομένων.

   - Εξαγωγή δεδομένων. Το Kibana μπορεί να εξάγει συγκεκριμένα δεδομένα σε αρχεία CSV, JSON και άλλα.

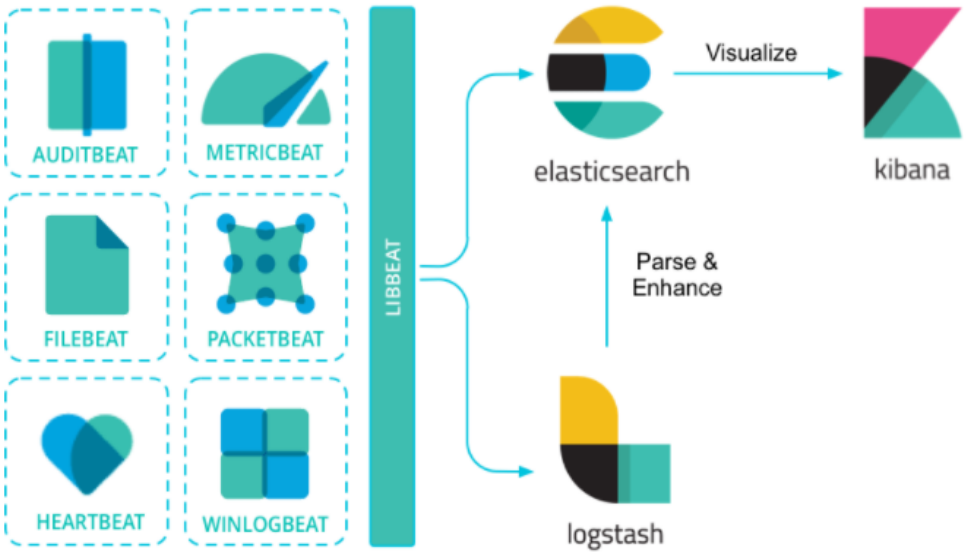
Alerting. Στις ρυθμίσεις του Kibana, μπορούν να καθοριστούν συνθήκες ή χρονικά περιθώρια στα οποία θα λειτουργήσει συναγερμός και θα ενημερώσει το χρήστη με το καθορισμένο τρόπο. Ο προκαθορισμένος τρόπος ενημέρωσης είναι η ενημέρωση με e-mail

3.6 Αρχιτεκτονική στοίβας ELK

Η βασική αρχιτεκτονική της στοίβας ELK [21], βασίζεται στις υπηρεσίες που αναφέρθηκαν προηγουμένως. Όπως φαίνεται και στο σχήμα 4.6.1, η είσοδος των δεδομένων προέρχεται από τα Beats ή από αρχεία που θα ορίσει ο διαχειριστής της στοίβας. Σε περίπτωση χρήσης Beats, ο πράκτορας θα πρέπει να εγκατασταθεί και να ρυθμιστεί σε συγκεκριμένα μηχανήματα που θα επιλέξει ο διαχειριστής, ανάλογα με την υποδομή του συστήματος και τις ανάγκες του οργανισμού.

Έπειτα αυτά τα δεδομένα προωθούνται στη Logstash. Η Logstash με τη σειρά της πρέπει να έχει τις κατάλληλες ρυθμίσεις φιλτραρίσματος και προώθησης των δεδομένων στην Elasticsearch η οποία θα κατατάξει και θα αποθηκεύσει τα δεδομένα.

Τέλος το Kibana, θα ρυθμιστεί ώστε να λαμβάνει δεδομένα από την Elasticsearch τα οποία μπορεί να οπτικοποιεί και να διαχειρίζεται.



Εικόνα 3.5. Παράδειγμα αρχιτεκτονικής στοίβας ELK [21]

Είναι σημαντικό να αναφερθεί πως η ροή των δεδομένων μεταξύ της Elasticsearch και του Kibana είναι αμφίδρομη [21], καθώς μέσω της διεπαφής του Kibana δίνεται η δυνατότητα στο χρήστη να επεξεργάζεται και να τροποποιεί τα δεδομένα της στοίβας.

Σε αυτό το σημείο φαίνεται πως μια στοίβα ELK, στη βασική της μορφή που μόλις παρουσιάστηκε, έχει πολλά στοιχεία ενός συστήματος SIEM που αναλύθηκε στο 2ο κεφάλαιο.

Συγκεκριμένα η στοίβα ELK υλοποιεί τις παρακάτω λειτουργίες που είναι απαραίτητες σε ένα σύστημα SIEM [14] [17] [21]:

- Συλλογή δεδομένων σχετικών με περιστατικά. Οι πράκτορες Beats, που είναι εγκατεστημένοι στα επιλεγμένα μηχανήματα, καθώς τα επιλεγμένα αρχεία που καθορίζονται στις ρυθμίσεις της Logstash, συλλέγουν όλα τα απαραίτητα δεδομένα που θα μπορούσε να αξιοποιήσει ένας μηχανισμός SIEM.

- Φιλτράρισμα και κανονικοποίηση δεδομένων. Το Logstash με τις κατάλληλες ρυθμίσεις φιλτράρει όλα τα εισερχόμενα δεδομένα, ώστε να μπορούν να χρησιμοποιηθούν από τις υπόλοιπες υπηρεσίες της στοίβας, και τελικά να αξιοποιηθούν από το χρήστη.

- Αποθήκευση και αναζήτηση δεδομένων. Όλα τα δεδομένα που εισέρχονται στη στοίβα από διάφορες πηγές, αφού περάσουν από το φιλτράρισμα της Logstash, προωθούνται προς την Elasticsearch, όπου περνάνε το στάδιο του indexing και αποθηκεύονται.
  Η Elasticsearch, εκτός από μηχανισμός αποθήκευσης δεδομένων, λειτουργεί και ως μηχανισμός αναζήτησης δεδομένων. Η αναζήτηση δεδομένων συνήθως αιτείται μέσω της διεπαφής χρήστη του Kibana.

- Οπτικοποίηση και ανάλυση δεδομένων. Το Kibana προσφέρει μια πλούσια διεπαφή χρήστη που επιτρέπει την ανάλυση με πολλούς τρόπους ανάλογα με τις ανάγκες του χρήστη.

Είναι προφανές πως αν και η στοίβα ELK δεν αποτελεί έναν πλήρη μηχανισμό SIEM, είναι μια πολύ καλή βάση πάνω στην οποία μπορεί να χτιστεί ένας λειτουργικός και πλήρης μηχανισμός SIEM [14][17][19][20].

Επίσης είναι σημαντικό να αναφερθεί πως με τις τελευταίες αλλαγές της Elastic το 2021, η άδεια διανομής για την Elasticsearch και για το Kibana άλλαξαν, από open σε άδεια SSPL [21]. Η άδεια αυτή επιτρέπει τη δωρεάν χρήση τους αλλά σε περίπτωση που χρησιμοποιηθούν για την υλοποίηση μιας υπηρεσίας που είναι προσβάσιμη δημοσίως, θα πρέπει και η υπηρεσία αυτή να είναι ανοιχτού κώδικα.

Κεφάλαιο 4ο: Υλοποίηση συστήματος SIEM με τη χρήση της στοίβας ELK

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα υλοποιηθεί το σύστημα SIEM με τη χρήση των υπηρεσιών της στοίβας ELK. Θα γίνει η περιγραφή της διαδικασίας, ξεκινώντας από την εγκατάσταση των υπηρεσιών της στοίβας ELK, και συνεχίζοντας προσθέτοντας επιπλέον λειτουργίες που πρέπει να υλοποιεί ένα ολοκληρωμένο σύστημα SIEM. Η εγκατάσταση θα γίνει σε λογισμικό Linux Ubuntu 20.04 LTS, το οποίο θα είναι και ο host για το σύστημα SIEM και στις περιπτώσεις χρήσης του 6^{ου} κεφαλαίου.

4.2 Εγκατάσταση και ρύθμιση της στοίβας ELK

4.2.1 Εγκατάσταση των υπηρεσιών της στοίβας ELK

Η εγκατάσταση των βασικών υπηρεσιών Elasticsearch, Logstash και Kibana, είναι σχετικά απλή διαδικασία. Η Elastic παρέχει ακριβή πληροφορίες για τη διαδικασία εγκατάστασης των υπηρεσιών της [21].

Προϋπόθεση για την εγκατάσταση των υπηρεσιών της στοίβας είναι η ύπαρξη της Java, συγκεκριμένα τουλάχιστον της έκδοσης openjdk-8. Συγκεκριμένα οι εντολές που χρησιμοποιήθηκαν για την εγκατάσταση ήταν:

```
$ apt install -y openjdk-8-jdk
$ apt install elasticsearch
$ apt install logstash
$ apt install kibana
```

Σε αυτό το σημείο εγκαταστάθηκαν επιτυχώς οι υπηρεσίες της στοίβας. Το επόμενο βήμα είναι να γίνουν οι απαραίτητες ρυθμίσεις και η εκκίνηση τους.

Για αρχή είναι αναγκαίο να ρυθμιστεί η Elasticsearch και το Kibana, ώστε να μπορεί να παίρνει δεδομένα από την Elasticsearch. Το Logstash προς το παρόν δεν απαιτεί συγκεκριμένες ρυθμίσεις καθώς δεν υπάρχουν συγκεκριμένα δεδομένα τα οποία να απαιτούν φιλτράρισμα και κανονικοποίηση.

Για το συγκεκριμένο SIEM, η Elasticsearch θα μπορεί χρησιμοποιήσει την διεύθυνση IP του τοπικού δικτύου που ανήκει ο SIEM host με τις υπηρεσίες. Με αυτό το τρόπο η Elasticsearch θα μπορεί να δεχτεί δεδομένα από πολλές πηγές του τοπικού δικτύου. Αυτές οι πηγές θα είναι κυρίως Beats που είτε θα μεταφέρουν δεδομένα απευθείας στην Elasticsearch είτε η μεταφορά θα περνάει από τη Logstash για την απαραίτητη κανονικοποίηση των δεδομένων. Αντίθετα το Kibana, θα τρέχει στη localhost loopback διεύθυνση, ώστε να μην υπάρχει πρόσβαση από άλλους σταθμούς του δικτύου.

Οι απαραίτητες ρυθμίσεις για το Kibana στο σχήμα 4.1.

```
# Διεύθυνση IP για το Kibana server
server.host: "localhost"
# Port στην οποία θα ακούει ο Kibana server
server.port: 5601

# Διεύθυνση IP της υπηρεσίας Elasticsearch, από την οποία μπορεί να αντλίσει
# δεδομένα το Kibana με web GET requests.
elasticsearch.hosts: "http://192.168.1.16:9200"

# Το index που δημιουργή το Kibana στην Elasticsearch για να αποθηκεύει τα δεδομένα
kibana.index: ".kibana"
```

Εικόνα 4.1. Ρυθμίσεις Kibana

Και οι απαραίτητες ρυθμίσεις για την Elasticsearch στην εικόνα 4.2

```
# Διεύθυνση IP της υπηρεσίας Elasticsearch
network.host: 192.168.1.16
# port στο οποίο ακούει η υπηρεσία Elasticsearch
http.port: 9200

# ---------------------------------- Cluster -----------------------------------

# Ονομασία του Cluster της Elasticsearch
cluster.name: ELK_SIEM
# Ορισμός αρχικού master κόμβου.
cluster.initial_master_nodes: ["master"]

# ------------------------------------ Node ------------------------------------
# Δημιουργία master κόμβου
node.name: master
```

Εικόνα 4.2. Ρυθμίσεις Elasticsearch

Σε αυτό το σημείο όλες οι υπηρεσίες της στοίβας μπορούν να ξεκινήσουν, με την εξής σειρά

```
$ service elasticsearch start
$ service kibana start
```

```
$ service logstash start
```

Μετά από μικρό χρονικό διάστημα οι υπηρεσίες θα πρέπει να έχουν ξεκινήσει. Η Logstash προς το παρόν δεν έχει ρυθμιστεί και δεν θα εξεταστεί σε αυτό το σημείο.

Για να επιβεβαιωθεί η λειτουργία της Elasticsearch μπορεί να χρησιμοποιηθεί είτε η εντολή curl είτε να χρησιμοποιηθεί κάποιος φυλλομετρητής (browser).

Με τη χρήση της εντολής curl τα αποτελέσματα πρέπει να έχουν παρόμοια μορφή με την εικόνα 4.3

```
it144267@siem:~$ curl 192.168.1.16:9200
{
  "name" : "master",
  "cluster_name" : "ELK_SIEM",
  "cluster_uuid" : "5f752hn0R9GtJl37lJSqCw",
  "version" : {
    "number" : "7.17.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "bee86328705acaa9a6daede7140defd4d9ec56bd",
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Εικόνα 4.3. Επαλήθευση λειτουργίας Elasticsearch

Για να ελεγχτεί η λειτουργία του Kibana, αρκεί να χρησιμοποιηθεί ένας browser όπου θα εισαχθεί η διεύθυνση του Kibana server. Στη περίπτωση μας είναι η localhost:5601. Εφόσον το Kibana service, έχει ξεκινήσει με επιτυχία θα πρέπει να εμφανιστεί η web διεπαφή χρήστη του Kibana στο φυλλομετρητή.

Στη διεπαφή εμφανίζονται τα υποσύνολα των λειτουργειών της στοίβας ELK. Είναι σημαντικό να τονιστεί ότι οι περισσότερες από αυτές τις λειτουργίες δεν είναι ενεργοποιημένες και απαιτούν επιπλέον ρυθμίσεις και προσθήκες.

Τέλος το Logstash κατά την εκκίνηση περιμένει ένα αρχείο ρυθμίσεων φιλτραρίσματος που περνιέται σαν παράμετρος στην εντολή εκκίνησης.

Οι απαραίτητες γενικές ρυθμίσεις για το Logstash είναι στην εικόνα 4.4

```
# ------------ API Settings -------------
# Define settings related to the HTTP API here.
#
# Ενεργοποίηση του HTTP API του Logstash.
api.enabled: true
#
# IP διεύθυνση στην οποία θα τρέχει το Logstash API
api.http.host: 127.0.0.1
#
# Ports στα οποία θα ακούει το API του Logstash
# Σε περίπτωση που είναι σύνολο απο ports, το Logstash επιλέγη τη πρώτη διαθέσιμη
api.http.port: 9600-9700
```

Εικόνα 4.4. Ρυθμίσεις Logstash

Ενώ για το αρχείο που παίρνει σαν παράμετρο με την εντολή στο παρακάτω πίνακα , η αρχική του μορφή φαίνεται στο σχήμα 4.5. Επιπλέον ρυθμίσεις θα γίνουν όταν ρυθμιστούν οι πηγές εισόδου δεδομένων εφόσον αυτό χρειαστεί

```
$ logstash -f elk_siem_main.conf
```

Το περιεχόμενο του αρχείου ρυθμίσεων elk_siem_main.conf απεικονίζεται στην εικόνα 4.5

```
input { stdin { } }
output {
  elasticsearch { hosts => ["192.168.1.16:9200"] }
  stdout { codec => rubydebug }
}
```

Εικόνα 4.5. Ρυθμίσεις φιλτραρίσματος Logstash

Σε αυτό το σημείο το σύστημα SIEM αποτελείται μόνο από τις πολύ βασικές λειτουργίες των υπηρεσιών της στοίβας ELK και σε καμία περίπτωση δεν μπορεί να χαρακτηριστεί ως ένα πλήρες σύστημα.

Στον παρακάτω πίνακα καταγράφονται ποιες λειτουργίες ενός SIEM εκτελεί το παρόν σύστημα, και ποιες όχι.

Βασική λειτουργία SIEM	
Συλλογή δεδομένων	**ΟΧΙ**
Κανονικοποίηση δεδομένων	**ΝΑΙ**
Αποθήκευση δεδομένων	**ΝΑΙ***
Ανάλυση και οπτικοποίηση δεδομένων	**ΝΑΙ**
Alerting	**ΟΧΙ**
Συσχέτιση συμβάντων	**ΟΧΙ**

Στη βασική λειτουργία της αποθήκευσης δεδομένων υπάρχει αστερίσκος καθώς αν και το σύστημα έτσι όπως είναι υλοποιημένο έχει τη δυνατότητα να αποθηκεύει δεδομένα στην Elasticsearch, τα δεδομένα αυτά δεν είναι ασφαλισμένα καθώς οποιοσδήποτε έχει πρόσβαση στο Kibana έχει αυτόματα πρόσβαση και στα δεδομένα του Elasticsearch. Η ασφάλεια στην αποθήκευση των δεδομένων είναι ιδιαίτερα σημαντική λειτουργία ενός SIEM [15][16], και στην επόμενη ενότητα προστίθεται αυτή η λειτουργία.

Τα τελευταία δύο (2) έτη η Elastic έκανε ριζικές αλλαγές στις υπηρεσίες της [21]. Η Elastic εκτός από τις υπηρεσίες ανοιχτού κώδικα που διαθέτει, είχε και υπηρεσίες κλειστού κώδικα όπως για παράδειγμα οι υπηρεσίες X-Pack και Elastic Endpoint. Αυτές οι υπηρεσίες σε συνδυασμό με τη στοίβα ELK δημιουργούσαν μια ολοκληρωμένη λύση SIEM κλειστού κώδικα την οποία η Elasticsearch διένεμε επί πληρωμή [21].

Μετά τις αλλαγές κάποιες από τις υπηρεσίες κλειστού κώδικα η Elastic αποφάσισε να τις μετατρέψει σε υπηρεσίες ανοιχτού κώδικα και τις προσθέσει στις υπάρχουσες υπηρεσίες της στοίβας ELK. Ένας από τους βασικούς λόγους ήταν και ευκολία ανάπτυξης και συντήρησης των υπηρεσιών αυτών από τη στιγμή που δούλευαν σε συνδυασμό με τις υπηρεσίες ανοιχτού κώδικα.

Συγκεκριμένα η Elastic πρόσθεσε πακέτα της υπηρεσίας x-pack στην Elasticsearch [30]. Ορισμένες λειτουργίες της X-Pack και γενικά της ELK στοίβας είναι δωρεάν στη χρήση τους κάτω από το SSPL License που αναλύθηκε στο κεφάλαιο 3. Τα πακέτα αυτά είναι απενεργοποιημένα από την αρχική εγκατάσταση των υπηρεσιών της στοίβας [21].

Προφανώς οι χρήστες που χρησιμοποιούν τη στοίβα ELK ως βάση ενός SIEM δεν είναι υποχρεωμένοι να χρησιμοποιήσουν τις λειτουργίες για παράδειγμα της υπηρεσίας X-Pack για security. Μπορούν είτε να προσθέσουν άλλες υπηρεσίες που επιθυμούν είτε να διαχειριστούν τις λειτουργίες του συστήματος τους με οποιονδήποτε τρόπο τους βολεύει.

4.3 Προσθήκη λειτουργειών ασφαλείας

Για να γίνει η προσθήκη λειτουργιών ασφαλείας ο χρήστης μπορεί να επιλέξει ανάμεσα σε έναν μεγάλο αριθμό υπηρεσιών που προσφέρουν κάποιες, είτε όλες τις υπηρεσίες ασφαλείας που απουσιάζουν από

το σύστημα SIEM που έχει ως βάση τη στοίβα ELK. Κάποιες από αυτές είναι ανοιχτού κώδικα και δωρεάν ενώ κάποιες άλλες διανέμονται επί πληρωμή.

Κάποιες γνωστές υπηρεσίες που καλύπτουν το κομμάτι ασφάλειας ενός SIEM είναι :

- X-Pack. Η υπηρεσία X-Pack [21], είναι υπηρεσία της Elastic που παλαιότερα ήταν υπηρεσία κλειστού κώδικα και η διανομή της γινόταν μόνο με πληρωμή, ενώ το 2021, έγινε και αυτή ανοιχτού κώδικα και συμπεριλήφθηκε στην Elasticsearch. Πλέον στη βασική της μορφή είναι δωρεάν, ενώ κάποιες άλλες παρέχονται μόνο με ειδική άδεια που αγοράζεται από την Elastic.

- ElastAlert. Η ElastAlert [32] είναι μια υπηρεσία ανοιχτού κώδικα που δουλεύει με τη στοίβα ELK και αναλαμβάνει τη λειτουργία του Alerting καθώς και τη δημιουργία κανόνων σχετικά με τα Alerts.

- OSSEC. Το OSSEC [22] που είναι ένα HIDS (Host intrusion detection system) ανοιχτού κώδικα και μπορεί να χρησιμοποιηθεί με την Elasticsearch

Στο σύστημα SIEM της συγκεκριμένης εργασίας θα χρησιμοποιηθεί η open-source δωρεάν έκδοση του X-Pack. Ο βασικός λόγος για την επιλογή αυτή είναι πως το X-Pack είναι σχεδιασμένο από την Elastic, την ίδια εταιρία που υλοποίησε και τις υπόλοιπες υπηρεσίες της στοίβας. Αυτό έχει ως αποτέλεσμα την καλύτερη μεταξύ τους συμβατότητα.

Για την ενεργοποίηση των βασικών λειτουργειών ασφαλείας του X-Pack, χρειάζεται να προστεθεί η κατάλληλη ρύθμιση, στο αρχείο ρυθμίσεων της Elasticsearch.

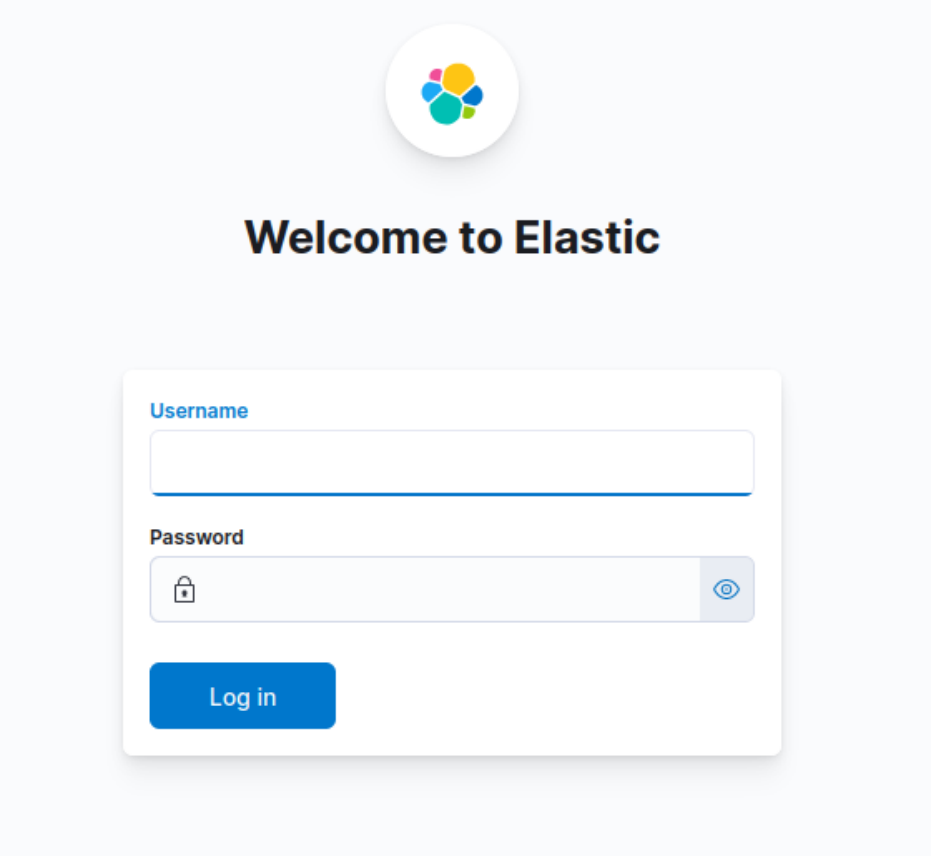
Η διαδικασία για την προσθήκη των λειτουργειών του X-Pack στο τρέχον σύστημα είναι η εξής [21]:

1. Διακοπή των υπηρεσιών Elasticsearch και Kibana
2. Προσθήκη της ρύθμισης για ενεργοποίηση του X-Pack στο αρχείο ρυθμίσεων της Elasticsearch
3. Προσθήκη ρύθμισης για υποστήριξη API κλειδιών στο αρχείο ρυθμίσεων της Elasticsearch
4. Εκκίνηση της υπηρεσίας Elasticsearch
5. Δημιουργία κωδικών πρόσβαση για τους χρήστες της Elasticsearch
6. Ρύθμιση του Kibana ώστε να συνδέεται στην Elasticsearch με χρήση κωδικού πρόσβασης.

7. Δημιουργία και ρύθμιση κλειδιού κρυπτογράφησης αντικειμένων για το Kibana

8. Εκκίνηση της υπηρεσίας Kibana

Μετά την ολοκλήρωση της διαδικασίας, κατά τη προσπάθεια εισόδου στη διεπαφή χρήστη Kibana, ο χρήστης θα πρέπει να εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης, όπως φαίνεται στην εικόνα 5.8.

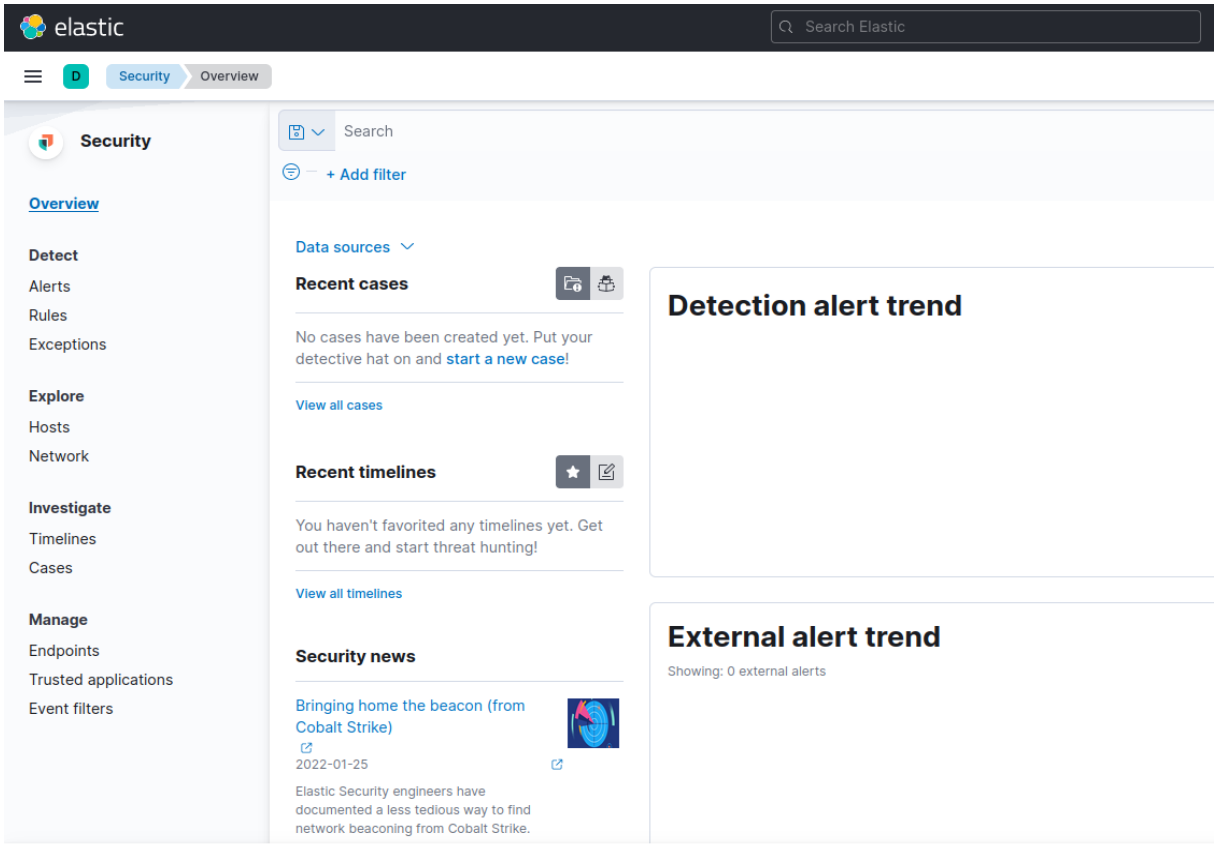


Εικόνα 5.6. Αυθεντικοποίηση χρήστη Kibana [21]

Ο αρχικός διαχειριστής του συστήματος είναι ο χρήστης με το όνομα elastic. Ο συγκεκριμένος χρήστης μπορεί να διαχειρίζεται τα δικαιώματα των υπόλοιπων χρηστών, η ακόμα και να προσθέτει η να διαγράφει χρήστες από το σύστημα.

Μετά τις ρυθμίσεις ασφάλειας, όταν ένας χρήστης εισέλθει πια στο Kibana, θα έχει πρόσβαση στις επιπλέον λειτουργίες ασφάλειας που μόλις ενεργοποιήθηκαν.

Για τη πρόσβαση των λειτουργιών αυτών υπάρχει ένα ειδικό μενού στο Dashboard του Kibana, όπως φαίνεται στην εικόνα 4.7



Εικόνα 4.7. Μενού ασφάλειας Kibana [21]

Από το συγκεκριμένο μενού θα αξιοποιηθούν κυρίως οι λειτουργίες Detection και Explore.

Οι λειτουργίες Detection όπως φαίνεται και στην εικόνα 4.7 είναι τρεις (3)΄:

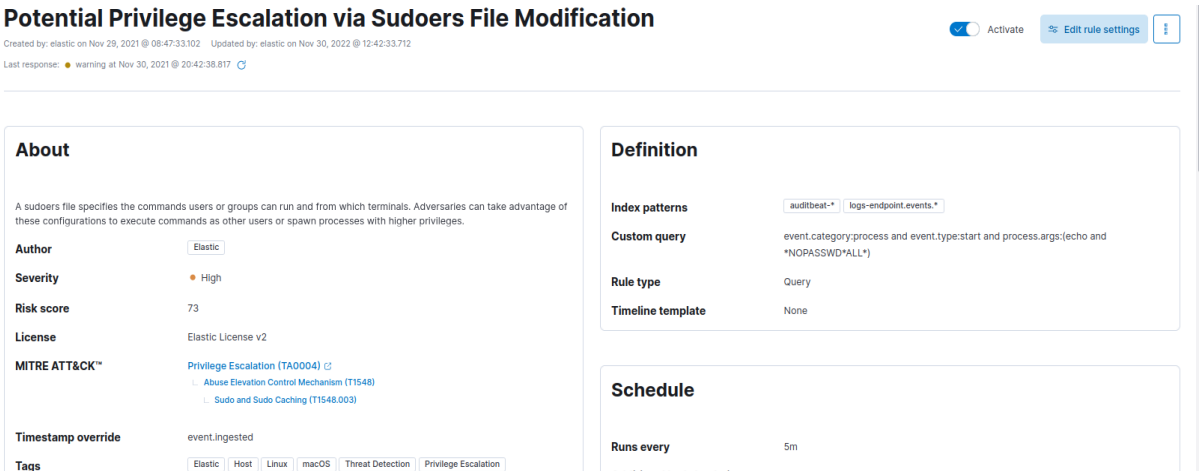
- Alerts. Η λειτουργία αυτή παρουσιάζει στο χρήστη όλα τα συμβάντα που θεωρήθηκαν απειλές με βάση τους κανόνες. Υπάρχει δυνατότητα φιλτραρίσματος των αποτελεσμάτων που εμφανίζονται για πιο στοχευμένη αναζήτηση

  .

- Rules. Ίσως η σημαντικότερη από τις 3 λειτουργίες. Με τη λειτουργία αυτή δημιουργούνται κανόνες που συσχετίζουν δεδομένα και συμβάντα που λαμβάνονται, με απειλές. Τα βασικά πεδία ενός κανόνα είναι:

  - Σύντομη περιγραφή για το συμβάν.

- Δείκτης επικινδυνότητας (risk score). Αριθμός από 1-100 που καθορίζει πόσο σοβαρή είναι μια απειλή. Από αυτόν καθορίζεται και η τιμή Severity που παίρνει τιμές:
    - Low
    - Medium
    - High
    - Critical
- Αναφορές από πρότυπα ή άρθρα σχετικά με την απειλή.
- Index patters, που περιγράφουν ποιοι πράκτορες μπορούν να συλλέξουν τα απαραίτητα δεδομένα για την αναγνώριση της απειλής. Για παράδειγμά για την αναγνώριση απειλών σε συστήματα Linux, στο σύστημα πρέπει να είναι εγκατεστημένος ο πράκτορας Auditbeat.
- Author, που είναι το όνομα του συγγραφέα του συγκεκριμένου κανόνα.
- Query. Είναι οι εντολές με τις οποίες αναζητούνται και συσχετίζονται συγκεκριμένα πεδία από αυτά που στέλνουν οι πράκτορες. Συνήθως οι γλώσσες που χρησιμοποιούνται για να ορίσουν τα Query είναι η EQL (Event Query Language) και η KQL (Kibana Query Language).
- Schedule. Ρύθμιση που καθορίζει ανά ποια χρονικά διαστήματα τρέχει ο κανόνας το Query του.



Σχήμα 4.8. Παράδειγμα κανόνα συσχέτισης Elastic [21] [29]

- Exceptions. Η λειτουργία αυτή τοποθετεί ορισμένους κανόνες στις εξαιρέσεις ώστε να μην καλούνται Alerts

Για να επιτραπεί η χρήση των λειτουργιών αυτών ο χρήστης πρέπει προηγουμένως να δημιουργήσει ένα API Key από το μενού Management-> Stack Management -> Security -> API keys.

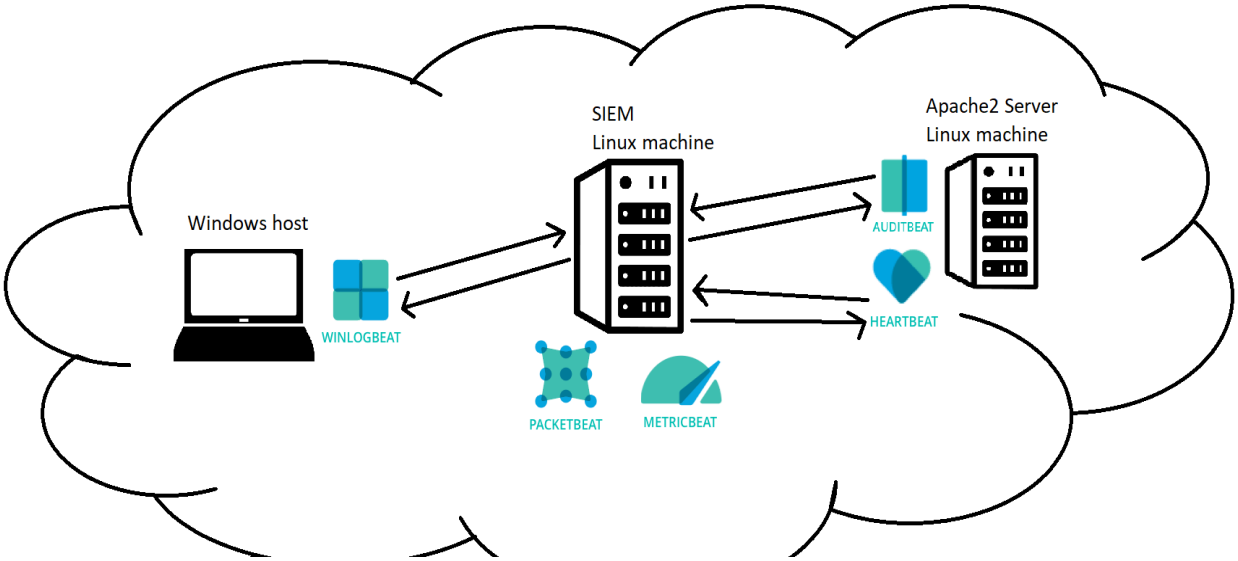
Επίσης, καθώς το X-Pack είναι ανοιχτού κώδικα, η Elastic προσφέρει δωρεάν επίσης, ένα σύνολο από κανόνες για γνωστές απειλές, καταγεγραμμένες είτε σε CVE's [27] είτε σε άλλους γνωστούς οργανισμούς όπως το MITRE ATT&CK [29]. Αυτοί οι κανόνες μπορούν να προστεθούν αυτόματα στους κανόνες ενός ELK SIEM που χρησιμοποιεί το X-Pack.

Μετά την προσθήκη του X-Pack στο παρόν SIEM ο πίνακας λειτουργιών διαμορφώνεται ως εξής:

Βασική λειτουργία SIEM	
Συλλογή δεδομένων	**ΟΧΙ**
Κανονικοποίηση δεδομένων	**ΝΑΙ**
Αποθήκευση δεδομένων	**ΝΑΙ**
Ανάλυση και οπτικοποίηση δεδομένων	**ΝΑΙ**
Alerting	**ΝΑΙ**
Συσχέτιση συμβάντων	**ΝΑΙ**

4.4 Εγκατάσταση Beats και συλλογή δεδομένων

Σε αυτό το σημείο, η μόνη λειτουργία που λείπει από το σύστημα SIEM είναι η συλλογή των δεδομένων. Για να δημιουργηθεί μια πιο ολοκληρωμένη εικόνα σχετικά με τη συλλογή δεδομένων αλλά και με τις περιπτώσεις χρήσης στο Κεφάλαιο 5, θα χρησιμοποιηθεί η αρχιτεκτονική της εικόνας 4.9.



Εικόνα 4.9. Αρχιτεκτονική συστήματος της Π.Α

Όπως φαίνεται και στην εικόνα το δίκτυο αποτελείται από τρείς σταθμούς.

- Ο πρώτος είναι το ο σταθμός που περιέχει το σύστημα SIEM και χρησιμοποιεί ως λειτουργικό σύστημα Linux Ubuntu 20.04 LTS, στο οποίο θα είναι εγκατεστημένοι οι πράκτορες Metricbeat και Packetbeat που συλλέγουν δεδομένα σχετικά με την επίδοση του συστήματος και πληροφορίες δικτύου

- Ο δεύτερος είναι ένας σταθμός χρήστη με λειτουργικό σύστημα Windows 10, με εγκατεστημένο το κατάλληλο πράκτορα Winlogbeat για τη καταγραφή δεδομένων σχετικά με συμβάντα σε λειτουργικά συστήματα Windows.

- Τέλος ο τρίτος σταθμός προσομοιώνει έναν Apache2 web server. Χρησιμοποιεί ως λειτουργικό σύστημα Linux Ubuntu 20.04 LTS και θα έχει εγκατεστημένους τους πράκτορες Auditbeat και Heartbeat για την επίβλεψη των λειτουργειών του λειτουργικού συστήματος και της διαθεσιμότητας του server αντίστοιχα.

Για να στείλουμε δεδομένα από κάθε Beat agent στο SIEM πρέπει κατά την εγκατάσταση του να προσδιορίσουμε τη διεύθυνση IP της Elasticsearch και να οριστούν είτε τα κατάλληλα username & password είτε να χρησιμοποιηθεί το API κλειδί.

Η πληροφορίες που χρειάζονται κατά την εγκατάσταση ενός Beat στο σύστημα αποθηκεύονται στο αρχείο ρυθμίσεων για κάθε beat, συνήθως αυτό το αρχείο έχει την ονομασία beat_name.yml [21].

Επίσης το αρχείο αυτό περιέχει πληροφορίες σχετικά με το ποια δεδομένα θα σταλούν από το σύστημα που επιβλέπουν προς την Elasticsearch ή το Logstash αντίστοιχα.

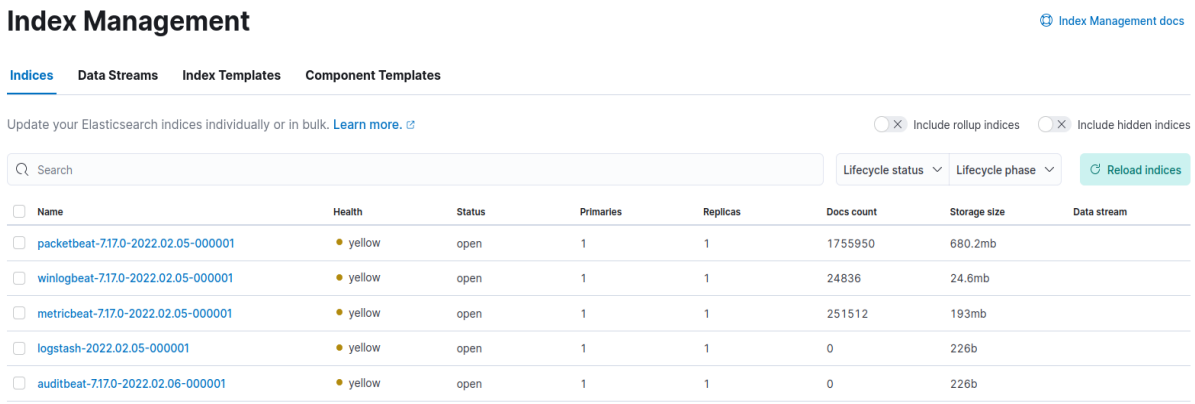
Τέλος μόλις ολοκληροθούν οι απαραίτητες ρυθμίσεις ο χρήστης χρειάζεται να τρέξει την εντολή setup για να δημιουργήσει τα κατάλληλα index στην Elasticsearch του SIEM, και μετά να εκκινήσει το Beat ως υπηρεσία.

```
$ ./bin/auditbeat setup

$service start auditbeat
```

Μετά τις εντολές αυτές θα πρέπει να φαίνονται τα Index στη διεπαφή χρήστη του Kibana.

Η πρόσβαση στη διεπαφή γίνεται από το μενού Dashboard->Stack Management-> Index Management.



Εικόνα 4.10. Επαλήθευση λειτουργίας των Beats [21]

Με τη χρήση των index που δημιουργήθηκαν μπορούν να δημιουργηθούν διάφορα Dashboards και γραφήματα ανάλυσης δεδομένων, ενώ ο πίνακας βασικών λειτουργειών διαμορφώνεται έτσι

Βασική λειτουργία SIEM	
Συλλογή δεδομένων	**ΝΑΙ**
Κανονικοποίηση δεδομένων	**ΝΑΙ**
Αποθήκευση δεδομένων	**ΝΑΙ**
Ανάλυση και οπτικοποίηση δεδομένων	**ΝΑΙ**
Alerting	**ΝΑΙ**
Συσχέτιση συμβάντων	**ΝΑΙ**

Κεφάλαιο 5ο: Δοκιμή και αξιολόγηση του υλοποιημένου συστήματος SIEM

5.1 Εισαγωγή

Πλέον το σύστημα είναι έτοιμο και μπορεί να λάβει δεδομένα από τα Beats που έχουν εγκατασταθεί, και να τα προωθήσει στο Elasticsearch. Αυτό που λείπει είναι οι κανόνες συσχέτισης δεδομένων και συμβάντων.

Σε αυτό το κεφάλαιο θα εξεταστούν περιπτώσεις κακόβουλων ενεργειών [29] και η δημιουργία των αντίστοιχων κανόνων.

Συγκεκριμένα θα εξεταστούν οι παρακάτω περιπτώσεις :

- Προσπάθεια κατεβάσματος απομακρυσμένου αρχείου σε Windows με τη χρήση Powershell.
- Προσπάθεια τροποποίησης των δυαδικών αρχείων του OpenSSH
- Προσπάθεια προσθήκης εξαίρεσης στο Windows Defender

5.2 Δημιουργία κανόνων

Για τη συγγραφή κανόνων σχετικών με συμβάντα, η Elastic χρησιμοποιεί τη γλώσσα EQL (Elastic Query Language) [21].

Για τη δημιουργία ενός νέου κανόνα ο χρήστης μπορεί να χρησιμοποιήσει τη διεπαφή χρήστη Kibana, Dashboard -> Security -> Rules -> Create new rule.

5.2.1 Δημιουργία κανόνα για κατέβασμα αρχείου μέσω Powershell

Το συγκεκριμένο είδος επίθεσης συμβαίνει όταν ένα κακόβουλο πρόγραμμα σχεδιασμένο για λειτουργικό σύστημα Windows χρησιμοποιεί το PowerShell για να κατεβάσει περισσότερα κακόβουλα εκτελέσιμα αρχεία.

Ο κανόνας δημιουργήθηκε με τη γλώσσα EQL, και έχει τη παρακάτω μορφή.

```
sequence by host.id, process.entity_id with maxspan=30s
  [network where process.name : ("powershell.exe", "pwsh.exe", "powershell_ise.exe") and network.protocol == "dns" and
   not dns.question.name : ("localhost", "*.microsoft.com", "*.azureedge.net", "*.powershellgallery.com", "*.windowsupdate.com", "metadata.google.internal") and
```

```
  not user.domain : "NT AUTHORITY"]
  [file where process.name : "powershell.exe" and event.type == "creation" and file.extension : ("exe", "dll", "ps1", "bat") and
  not file.name : "__PSScriptPolicy*.ps1"]
```

Ο κανόνας συσχετίζει την εκτέλεση του PowerShell με τη δημιουργία ενός νέου εκτελέσιμου σε Windows αρχείου, δηλαδή με καταλήξεις (exe, dll, ps1, bat)

5.2.2 Δημιουργία κανόνα για τροποποίηση των δυαδικών αρχείων του OpenSSH

Το συγκεκριμένη είδος επίθεσης προσπαθεί να γράψει πάνω σε σημεία των δυαδικών αρχείων του ssh με σκοπό να παρακάμψει πιθανές λειτουργίες αυθεντικοποίησης.

Χρησιμοποιώντας την EQL ο κανόνας έχει την εξής μορφή

```
event.category:file and event.type:change and
process.name:* and
(file.path:(/usr/sbin/sshd or /usr/bin/ssh) or file.name:libkeyutils.so)
```

Ο συγκεκριμένος κανόνας περιμένει να συμβεί γεγονός change που δείχνει την τροποποίηση κάποιου αρχείου, ενώ συσχετίζει την αλλαγή με το όνομα του αρχείου που τροποποιείται.

5.2.3 Δημιουργία κανόνα για προσθήκη εξαίρεσης στο Windows Defender

Η συγκεκριμένη επίθεση προσπαθεί να προσθέσει κακόβουλα δυαδικά αρχεία στις εξαιρέσεις του Windows Defender μέσω του powershell, ώστε να μην αναγνωριστούν ως πιθανές απειλές από το λειτουργικό σύστημα

Χρησιμοποιώντας την EQL ο κανόνας έχει την εξής μορφή.

```
process where event.type == "start" and
(process.name : ("powershell.exe", "pwsh.exe", "powershell_ise.exe") or process.pe.original_file_name in ("powershell.exe", "pwsh.dll", "powershell_ise.exe")) and
 process.args : ("*Add-MpPreference*", "*Set-MpPreference*") and
 process.args : ("*-Exclusion*")
```

Ο συγκεκριμένος κανόνας συσχετίζει την εκκίνηση μιας διαδικασία με την ονομασία της (PowerShell) και την προσθήκη εξαίρεσης στο τοίχος προστασίας των Windows.

5.3 Προσομοίωση κακόβουλων ενεργειών

Για την προσομοίωση των κακόβουλων ενεργειών χρησιμοποιήθηκαν πολύ απλές τεχνικές για να ελεγχθεί ο μηχανισμός συσχέτισης και Alerting. Οι αναφερθέντες επιθέσεις συνήθως είναι πολύ πιο περίπλοκες χρησιμοποιώντας διάφορους μηχανισμούς καμουφλαρίσματος.

5.3.1 Προσομοίωση κατεβάσματος αρχείου μέσω PowerShell

Για τη προσομοίωση αυτής της επίθεσης αρκεί η δημιουργία ενός απλού PowerShell αρχείου που κατεβάζει ένα αρχείο.

Για παράδειγμα:

Test_case.ps1

```
Invoke-WebRequest "https://download.filezilla-project.org/client/FileZilla_3.57.0_win64_sponsored2-setup.exe" -OutFile "potentialMalware"
```

5.3.2 Προσομοίωση τροποποίησης δυαδικού αρχείο του OpenSSH

Η προσομοίωση αυτής της επίθεσης είναι επίσης σχετικά απλή στην υλοποίηση, το μόνο που απαιτείται είναι η εγγραφή τυχαίων bytes στο δυαδικό αρχείο του ssh.

Αυτό μπορεί να γίνει με το επόμενο bash script

Test_case.sh

```
#!/bin/bash
echo echo "0x011111F1321321AD135" >> /usr/bin/ssh
```

Προφανώς η εκτέλεση αυτού του script δεν θα παρακάμψει τον μηχανισμό της αυθεντικοποίησης του ssh. Η εκτέλεση του πιθανότατα να κάνει την εντολή ssh μη λειτουργική. Για να παρακαμφθεί ο μηχανισμός αυθεντικοποίησης θα πρέπει να γραφτούν συγκεκριμένα δεδομένα στο δυαδικό αρχείο και για να γίνει αυτό θα πρέπει να γίνει ανάλυση του δυαδικού αρχείου.

5.3.3 Προσομοίωση προσθήκης εξαίρεσης στο Windows Defender

Για τη προσομοίωσης αυτής της επίθεσης απαιτείται ένα απλό PowerShell ps1 εκτελέσιμο που να προσθέτει μια εξαίρεση στο τοίχος προστασίας Windows Defender.

Ένα παράδειγμα είναι :

Test_case.ps1

```
echo Adding exception to Windows Defender
MpPreference -ExclusionPath “C:\it144267\potential_malware.exe”
```

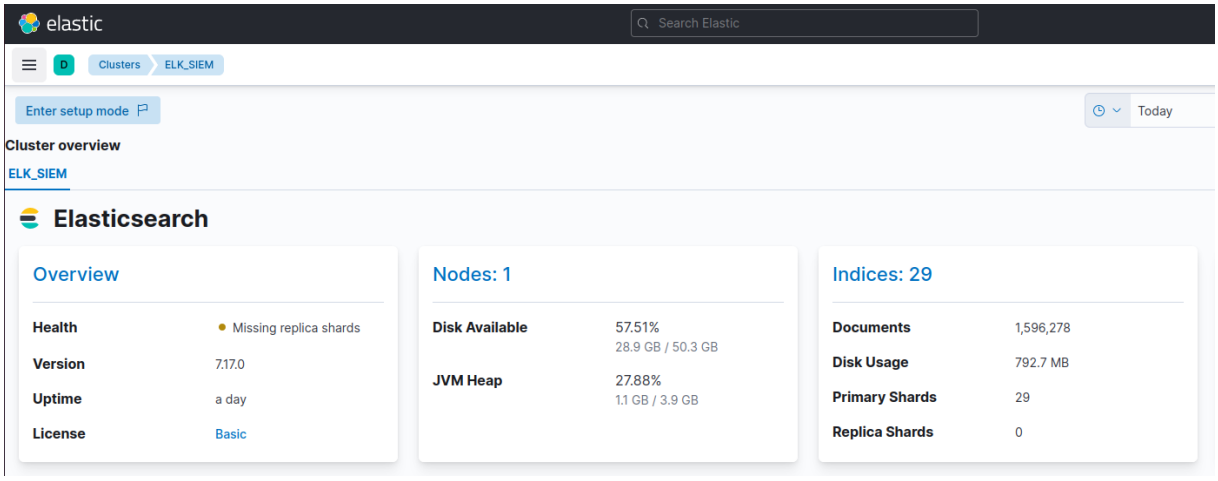
Το εκτελέσιμο ps1 θα προσθέσει εξαίρεση για το αρχείο potential_malware.exe στους κανόνες του windows defender έχοντας ως συνέπεια να μη ανιχνευθεί ένα πιθανό κακόβουλο αρχείο.

5.4 Αξιολόγηση

Σε αυτή την ενότητα θα αξιολογηθεί το σύστημα SIEM που υλοποιήθηκε ως προς τις λειτουργίες του και την αποδοτικότητα του.

Αρχικά το MetricBeat που εγκαταστάθηκε με σκοπό την επίβλεψη των πόρων, του συστήματος SIEM λειτουργεί και είναι διαθέσιμο, στη διεπαφή του Kibana, Dashboard -> Management -> Stack Monitoring.

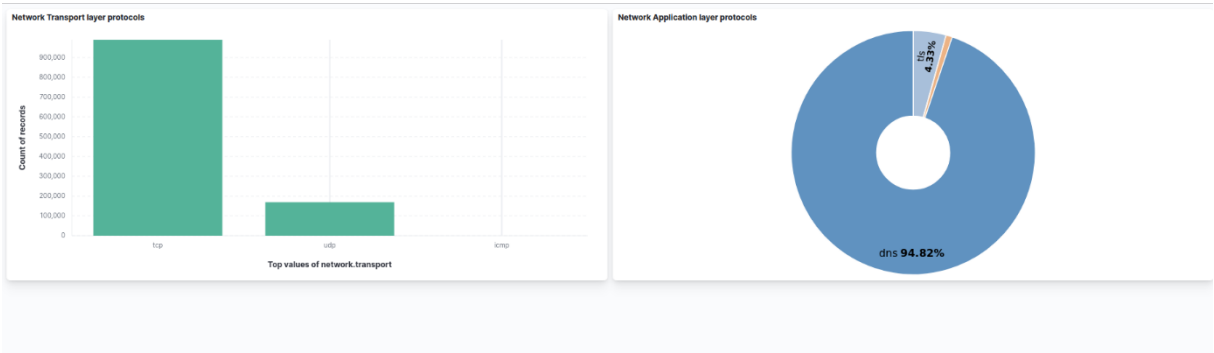
Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα.



Εικόνα 5.1.Χρήση Metrics για την επίβλεψη των πόρων του συστήματος [21]

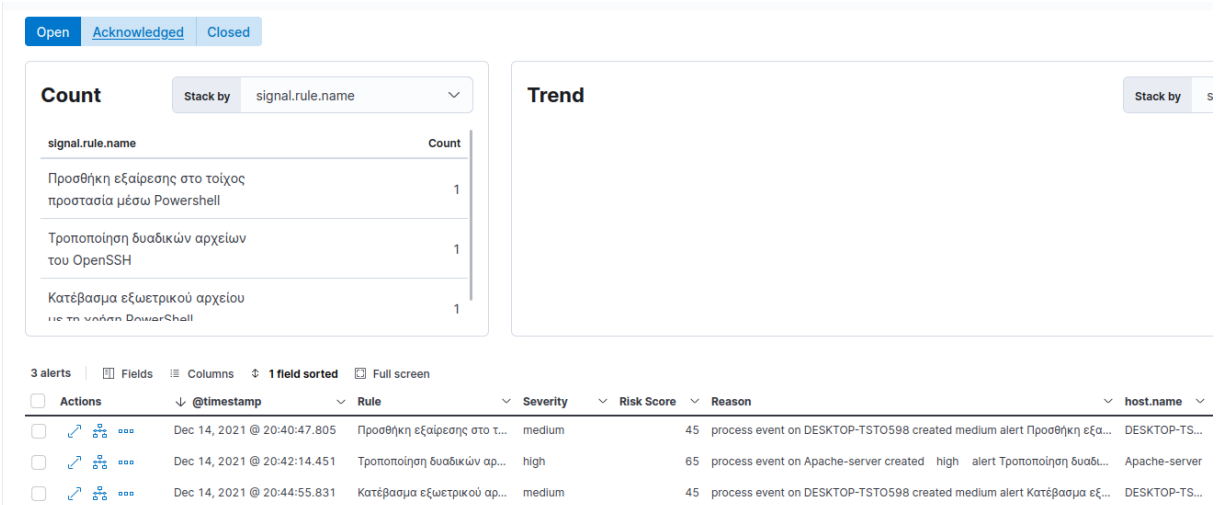
Μετά την εκτέλεση των κακόβουλων αρχείων στα συστήματα Windows και τον Apache Server, το SIEM θα πρέπει να έχει αναγνωρίσει τις απειλές και οι οποίες θα πρέπει να εμφανίζονται χαρακτηριστικά στην διεπαφή των Alerts.

Επίσης με τη χρήση του Packetbeat που εγκαταστάθηκε στο σύστημα SIEM είναι δυνατή η δημιουργία γραφημάτων για ανάλυση δεδομένων.



Εικόνα 5.2. Δημιουργία γραφημάτων για ανάλυση δεδομένων δικτύου [21]

Ανοίγοντας τη κατάλληλη διεπαφή του Kibana επιβεβαιώνεται πως και αυτή λειτουργία του SIEM δουλεύει και πως το SIEM εντόπισε τις παραπάνω κακόβουλες ενέργειες



Εικόνα 5.3. Εμφάνιση όλων των συμβάντων στη διεπαφή Alerts [21]

Από την καταγραφή γεγονότων μπορεί να βγει το συμπέρασμα πως η συλλογή και η κανονικοποίηση των δεδομένων και τον συμβάντων είναι επιτυχής και οι μηχανισμοί του SIEM λειτουργούν όπως θα έπρεπε.

Σε γενικές γραμμές η στοίβα ELK φαίνεται πως μπορεί να χρησιμοποιηθεί ως μια καλή βάση για τη υλοποίηση ενός συστήματος SIEM.

Παρόλα αυτά το SIEM που υλοποιήθηκε αν και είναι λειτουργικό ή προσθήκη κανόνων και όλων των επιπρόσθετων λειτουργιών ασφαλείας είναι μια δύσκολη διαδικασία που απαιτεί χρόνο, ψάξιμο και εξοικείωση με όλες τις τεχνολογίες στης στοίβας.

Στο επόμενο κεφάλαιο αξιολογούνται κάποιοι γνωστοί μηχανισμοί ανοιχτού κώδικα που είτε χρησιμοποιούνται ως SIEM είτε ως κομμάτια ενός SIEM.

Κεφάλαιο 6ο: Καταγραφή και αξιολόγηση συστημάτων και τεχνολογιών SIEM ανοιχτού κώδικα

6.1 Εισαγωγή

Στο χώρο της ασφάλειας, υπάρχουν πολλές λύσεις και τεχνολογίες που δίνουν λύσεις σε προβλήματα και απειλές από κακόβουλες ενέργειες. Όπως φάνηκε και στα προηγούμενα κεφάλαια μια λύση που χρησιμοποιείται ευρέως είναι τα συστήματα SIEM. Ακόμα και μεταξύ των συστημάτων SIEM υπάρχει μια μεγάλη πληθώρα από επιλογές για έναν χρήστη, κάποιες είναι ανοιχτού κώδικα και διανέμονται δωρεάν ενώ κάποιες άλλες είναι κλειστού κώδικα και διανέμονται επί πληρωμή. Σε αυτό το κεφάλαιο θα καταγραφούν και θα αξιολογηθούν κάποια συστήματα που μπορούν να χρησιμοποιηθούν ως SIEM ανοιχτού κώδικα, τα οποία μπορούν να εκτελέσουν περισσότερες από μια βασικές λειτουργίες που πρέπει να εκτελεί ένα σύστημα SIEM.

Η πραγματικότητα είναι πως συνήθως τα συστήματα SIEM που είναι κλειστού κώδικα και διανέμονται επι πληρωμή, έχουν περισσότερες λειτουργίες και είναι πιο εύκολα στην εγκατάσταση τους, αλλά αυτό δεν είναι κανόνας.

Επίσης συχνά, εταιρίες που διανέμουν συστήματα SIEM επί πληρωμή, διανέμουν και πιο απλουστευμένες εκδόσεις του ίδιου συστήματος, με λιγότερες λειτουργίες, δωρεάν.

6.2 Κριτήρια αξιολόγησης των συστημάτων SIEM

Τα κριτήρια με τα οποία αξιολογείται ένα σύστημα SIEM, δεν είναι πάντα ξεκάθαρα. Διαφορετικοί οργανισμοί, έχουν διαφορετικές ανάγκες, κάτι που οδηγεί σε διαφορετικά κριτήρια αξιολόγησης. Ένας μεγάλος οργανισμός με περίπλοκα συστήματα και λειτουργίες και διαχειρίζεται ευαίσθητα δεδομένα, δεν έχει τις ίδιες ανάγκες με έναν οργανισμό που το δίκτυο του αποτελείτε από μερικούς τερματικούς σταθμούς και λίγες συσκευές δικτύου. Επίσης ακόμα και στις απαραίτητες λειτουργίες του SIEM, μπορεί να υπάρξουν διαφορές στις ανάγκες.

Η διαδικασία προσδιορισμού των κριτηρίων αξιολόγησης και των αναγκών ασφάλειας ενός οργανισμού πολλές φορές δεν είναι απλή.

Συνήθως χωρίζεται σε βήματα. Κάποια από τα βασικότερα είναι:

Προσδιορισμός οικονομικών δυνατοτήτων του οργανισμού. Είναι σημαντικό ο οργανισμός πριν επιλέξει μια SIEM λύση, να αναλογιστεί τόσο το κόστος αγοράς, όσο και το κόστος ρύθμισης, αλλά και συντήρησης ενός εργαλείου SIEM. Η διαδικασία της συντήρησης ενός SIEM ανάλογα με τις λειτουργίες και τις δυνατότητες του μπορεί να γίνει μια ιδιαίτερα δύσκολη διαδικασία που απαιτεί εξειδικευμένο προσωπικό.

Καταγραφή και ανάλυση του συστήματος του οργανισμού. Ο βασικός σκοπός ενός SIEM είναι να επιβλέπει και να προστατεύει ένα υπολογιστικό σύστημα. Αυτό σημαίνει πως η δομή και η αρχιτεκτονική του συστήματος, παίζουν τεράστιο ρόλο στην επιλογή του κατάλληλο SIEM.

Προσδιορισμός αναγκών, πολιτικών και πρωτοκόλλων του οργανισμού. Σε αυτό το βήμα, προσδιορίζονται οι ανάγκες του οργανισμού, όχι μόνο από την πλευρά της ασφάλειας του συστήματος, αλλά και γενικότερα, καθώς το σύστημα SIEM θα πρέπει να λειτουργήσει ως μέρος του οργανισμού που έχει τις δικές του πολιτικές και λειτουργίες. Αυτό σημαίνει πως θα πρέπει να προσδιοριστούν τα κατάλληλα κριτήρια [1][4][8][10][13] ώστε το SIEM να ταιριάξει λειτουργικά μέσα στο ήδη υπάρχον σύστημα του οργανισμού.

Στο συγκεκριμένο κεφάλαιο η αξιολόγηση θα γίνει σε τρία (3) στάδια με βάση τρία (3) σύνολα παραμέτρων:

1. Βασικές λειτουργίες, που περιέχει τις λειτουργίες που πρέπει να υλοποιεί ένα σύστημα SIEM όπως αναλυθήκαν στο κεφάλαιο 2. Η αξιολόγηση γίνεται με βάση αν οι λειτουργίες αυτές υπάρχουν στο σύστημα υλοποιημένες χωρίς κάποια επιπλέον προσθήκη.
   Οι λειτουργίες αυτές καταγράφονται συνοπτικά στο παρακάτω πίνακα.

Αριθμός	Βασική Λειτουργία	Σύντομη Περιγραφή
1	Συλλογή δεδομένων	Δυνατότητα του SIEM να συλλέγει δεδομένα από διάφορες πήγες
2	Κανονικοποίηση δεδομένων	Δυνατότητα του SIEM να φιλτράρει και κανονικοποιεί τα δεδομένα που συλλέγει
3	Ανάλυση και οπτικοποίηση δεδομένων	Μηχανισμοί ανάλυσης των δεδομένων καθώς και οπτικοποίηση τους με κάποια διεπαφή χρήστη
4	Ασφαλής αποθήκευση δεδομένων	Ασφαλής αποθήκευση των δεδομένων που συλλέγονται για μελλοντική ανάλυση ή χρήση
5	Λειτουργία Alerting	Λειτουργία ενημέρωση σχετικά με ορισμένα συμβάντα. Η ενημέρωση γίνεται είτε στη διεπαφή χρήστη, είτε με e-mail
6	Συσχέτιση δεδομένων και συμβάντων	Συσχέτιση δεδομένων και γεγονότων μεταξύ τους

2. Επιπρόσθετες λειτουργίες, περιέχει πιο προχωρημένες λειτουργίες [1] που δεν περιλαμβάνονται σε όλα τα SIEM, συνήθως οι πρόσθετες λειτουργίες περιλαμβάνονται στα

συστήματα SIEM κλειστού κώδικα, για αυτό στη παρούσα αξιολόγηση περιέχονται επιπλέον λειτουργίες που είναι πιο πιθανό να συναντηθούν σε συστήματα SIEM ανοιχτού κώδικα.

Οι λειτουργίες αυτές καταγράφονται συνοπτικά στο παρακάτω πίνακα.

Αριθμός	Επιπρόσθετη λειτουργία	Σύντομη περιγραφή
1	Αλγόριθμοι μηχανικής μάθησης	Αλγόριθμοι που εκπαιδεύονται και μπορούν να εντοπίσουν άγνωστες απειλές που δεν υπάρχουν στους κανόνες.
2	Υποστήριξη πολιτικών και πρωτοκόλλων λειτουργίας	Υποστήριξη διάφορων πολιτικών δεδομένων ή πρωτοκόλλων της εταιρίας.
3	Ανάλυση συμπεριφοράς χρηστών	Αναλύει τη συμπεριφορά των χρηστών και των μηχανημάτων στο σύστημα που επιβλέπει
4	Δημιουργία assessments	Έχει τη δυνατότητα εξαγωγής των δεδομένων σε security assessments
5	Αυτόματη διαχείριση περιστατικών	Έχει τη δυνατότητα να κάνει τις προγραμματισμένες ενέργειες όταν αντιληφθεί συγκεκριμένες απειλές.

3. Γενικά χαρακτηριστικά του συστήματος SIEM. Τα χαρακτηριστικά αυτά είναι πιο γενικά και αξιολογούν κυρίως την εμπειρία του χρήστη. Ενώ στους προηγούμενους πίνακες οι λειτουργίες κατά την αξιολόγηση λαμβάνουν δυαδικές τιμές, σε αυτό το πίνακα η αξιολόγηση γίνεται με άριστα το 5.

Αριθμός	Χαρακτηριστικό	Σύντομη περιγραφή
1	Υποστήριξη λειτουργικών συστημάτων	Σε πόσα λειτουργικά συστήματα μπορεί να υλοποιηθεί το συγκεκριμένο σύστημα SIEM
2	Ευκολία εγκατάστασης και χρήσης	Ευκολία στην εγκατάσταση αλλά και στη χρήση του, για παράδειγμα διαθέσιμες διεπαφής χρήστη κτλ.
3	Documentation και καθοδήγηση	Το πόσο εύκολα είναι να βρεθούν οδηγίες χρήσης και άλλα έντυπα που βοηθούν στη χρήση ή περιγράφουν λύσεις σε πιθανά προβλήματα
4	Επεκτασιμότητα	Πόσο εύκολα είναι να επεκταθεί στην επίβλεψη περισσότερων μηχανημάτων ή να γίνει μέρος ενός μεγαλύτερου συστήματος

5	Ευκολία προσθήκης επιπλέον λειτουργιών	Πόσο εύκολο είναι να προστεθούν είτε άλλες λειτουργίες ανοιχτού κώδικα πάνω στο συγκεκριμένο σύστημα

6.3 Καταγραφή συστημάτων SIEM ανοιχτού κώδικα

Σε αυτή την ενότητα θα καταγραφούν κάποια από τα πιο γνωστά συστήματα και τεχνολογίες ανοιχτού κώδικα που χρησιμοποιούνται είτε ως βάση συστημάτων SIEM είτε ως ολοκληρωμένα SIEM από διάφορους οργανισμούς και θα αξιολογηθούν στην επόμενη ενότητα.

Οι μηχανισμοί αυτοί είναι:

- AllientVault OSSIM. Το OSSIM (Open Source Security Information Management) [31] είναι ένα από τα πιο διαδεδομένα συστήματα SIEM. Είναι ανοιχτού κώδικα και διανέμεται δωρεάν από την εταιρία AT & T Cybersecurity. Το OSSIM, διανέμεται ως ξεχωριστό λειτουργικό σύστημα το οποίο είναι βασισμένο στο λειτουργικό σύστημα Linux Debian.

Κάποια από τα βασικότερα εργαλεία που χρησιμοποιεί το OSSIM είναι [31]:

- Spade: εντοπισμός προβλημάτων δικτύου
- Snort: σύστημα εντοπισμού παραβίασης δικτύου
- Spade: εντοπισμός προβλημάτων δικτύου
- Snort: σύστημα εντοπισμού παραβίασης δικτύου
- Acid: εργαλείο εμφάνισης logs
- OpenNMS: Εργαλείο ελέγχου διαθεσιμότητας υπηρεσίας
- Mrtg: εργαλείο για γραφήματα
- Mysql & PostgreSQL: Βάσεις αποθήκευσης δεδομένων
- Nessus: Εργαλείο δημιουργίας vulnerability assessment
- Nmap: Εργαλείο χαρτογράφησης και ανάλυσης δικτύου

- Wazuh . Το Wazuh [23] είναι είναι ένα εργαλείο που βασίζεται σε τρείς (3) βασικές υπηρεσίες [23]:

  - Τους Wazuh πράκτορες (agents), που απαιτούν εγκατάσταση στο σύστημα που επιβλέπουν και στέλνουν συνέχεια πληροφορίες σχετικά με συμβάντα στον Wazuh server. Οι πράκτορες του Wazuh είναι βασισμένοι στο OSSEC για να αντιλαμβάνονται συμβάντα ασφαλείας.

  - Wazuh server. Είναι ο server που δέχεται δεδομένα από τους Wazuh agents. Ο server ελέγχει και αναλύει τα δεδομένα και χρησιμοποιεί τους κανόνες του για να τα κατατάξει σε απειλές και μη.

- Elastic stack. Το Wazuh χρησιμοποιεί τη στοίβα ELK για να υλοποιήσει τις λειτουργίες της αναζήτησης, αποθήκευσης και οπτικοποίησης των δεδομένων καθώς χρησιμοποιεί το Kibana ως διεπαφή χρήστη.

- Apache Metron. Το Apache metron [24] είναι ένα σύστημα που αξιοποιεί διάφορες open source τεχνολογίες, για να προσφέρει ένα κεντροποιημένο σύστημα ασφάλειας και ανάλυσης που συχνά χρησιμοποιείται και ως SIEM.

- SIEMonster. Το SIEMonster [25] είναι μηχανισμός που αποτελείτε από στοιχεία της στοίβας ELK (Logstash και Filebeat), καθώς και το OSSEC που χρησιμοποιεί για τον εντοπισμό συμβάντων. Υποστηρίζει εγκατάσταση σε Cloud και σε Docker containers.

- OSSEC. Το OSSEC [22] είναι HIDS (Host-based Intrusion System), και αποτελεί τη βάση για πολλά SIEM εργαλεία. Είναι ένα σύστημα που προσφέρει αναγνώριση απειλών και ανίχνευση επιθέσεων σε πραγματικό χρόνο. Επίσης προσφέρει ορισμένες επιπλέον τεχνολογίες όπως εντοπισμός κακόβουλου λογισμικού.

- ELK Stack [21][30]. Η στοίβα υπηρεσιών ELK αποτελεί και αυτή βάση για πολλά συστήματα SIEM καθώς έχει πολλές δυνατότητες στην αποθήκευση, κανονικοποίηση και διαχείριση των δεδομένων. Αποτελείται από τις υπηρεσίες Elasticsearch, Logstash και Kibana ενώ για να λάβει δεδομένα χρησιμοποιεί τα Beats που είναι πράκτορες (agents) της στοίβας.

6.4 Αξιολόγηση συστημάτων SIEM ανοιχτού κώδικα

Η αξιολόγηση των συστημάτων και των τεχνολογιών θα γίνει σε 3 στάδια, ανάλογα με τους πίνακες της ενότητας 6.2. Η πρώτη γραμμή περιλαμβάνει τους αριθμούς των λειτουργιών ή χαρακτηριστικών, ενώ η πρώτη στήλη τα ονόματα των συστημάτων που αξιολογούνται.

Προφανώς η αξιολόγηση γίνεται με της προσωπικές εκτιμήσεις του συγγραφέα αυτής της Π.Α, και δεν σημαίνει ότι είναι απολύτως σωστές, ούτε έχει σκοπό να προβάλει ή να δυσφημήσει κάποια υπηρεσία η εργαλείο που θα αξιολογηθεί.

Η αξιολόγηση θα γίνει με τον εξής τρόπο.

Για τους πίνακες που παίρνουν δυαδικές τιμές, δηλαδή των λειτουργειών που εκτελούν τα συστήματα, ο αριθμός των υπηρεσιών που υλοποιούνται θα διαιρείται με τον συνολικό αριθμό των λειτουργιών, και το κλάσμα θα πολλαπλασιάζεται με το 10.

Για το πίνακα 3, θα αθροίζονται οι αριθμοί της κάθε γραμμής που περιέχουν τιμές από 1-5, και θα το αποτέλεσμα θα πολλαπλασιάζεται με το 2.

Η τελική αξιολόγηση θα περιέχει τη μέση τιμή από τους 3 πίνακες.

Για το πρώτο πίνακα έχουμε.

	1	2	3	4	5	6	Σύνολο
AllienVault OSSIM	1	1	1	1	1	1	10
Wazuh	1	1	1	1	1	1	10
Apache Metron	1	1	0	1	1	1	8.5
SIEMonster	1	1	1	1	1	1	10
OSSEC	1	1	0	0	1	1	6.5
ELK Stack	1	1	1	1	0	0	6.5

Τα αποτελέσματα της πρώτης αξιολόγησης είναι αναμενόμενα καθώς οι τέσσερεις πρώτες υπηρεσίες χρησιμοποιούν πολλές άλλες υπηρεσίες ανοιχτού κώδικα σαν βάση ενώ αντίθετα οι δύο τελευταίες αποτελούν βάση για άλλες πολλές υπηρεσίες

Για τον δεύτερο πίνακα με τις επιπρόσθετες λειτουργίες έχουμε.

	1	2	3	4	5	Σύνολο
AllienVault OSSIM	0	1	0	1	0	3.5
Wazuh	0	1	0	0	1	3.5
Apache Metron	1	1	1	0	0	5
SIEMonster	0	1	0	1	1	5
OSSEC	0	0	0	0	1	2
ELK Stack	0	1	0	0	0	2

Τέλος για το πίνακα με τα χαρακτηριστικά έχουμε

	1	2	3	4	5	M.O
AllienVault OSSIM	1	4	3	2	2	2.6 | 5.2
Wazuh	3	3	4	2	2	2.8 | 5.6
Apache Metron	2	4	3	3	4	3.6 | 7.2

SIEMonster	2	4	3	1	2	2.4 | 4.8
OSSEC	4	3	3	4	4	3.6 | 7.2
ELK Stack	5	3	4	5	4	4.2 | 8.4

Βγάζοντας το μέσο όρο από τους τρείς πίνακες προκύπτουν οι τελικές βαθμολογίες για το κάθε σύστημα.

	Τελική βαθμολογία
AllienVault OSSIM	6.2
Wazuh	6.3
Apache Metron	6.9
SIEMonster	6.6
OSSEC	5.3
ELK Stack	5.8

Από την αξιολόγηση προκύπτει πως τα περισσότερα υλοποιημένα SIEM είναι πιο γρήγορα, και εύκολα στην εγκατάσταση και στη χρήση καθώς διαθέτουν όλες τις βασικές ιδιότητες. Το μειονέκτημα τους όμως είναι η επεκτασιμότητα και η τροποποίηση τους. Είναι το σύστημα που είναι και τίποτα περισσότερο ή λιγότερο.

Σε αντίθεση υπηρεσίες όπως το OSSEC ή η στοίβα ELK, αν και από μόνες τους προσφέρουν περιορισμένες υπηρεσίες σε αντίθεση με τα έτοιμα συστήματα SIEM, έχουν τη δυνατότητα να επεκταθούν και να γίνουν μέρος ενός μεγαλύτερου συστήματος. Θα πρέπει να ληφθεί υπόψη πως η επέκταση και τροποποίηση αν και είναι δυνατή, είναι μια διαδικασία που απαιτεί ιδιαίτερη τεχνογνωσία.

Η επιλογή εξαρτάται από τον κάθε οργανισμό, τις ανάγκες του και το σύστημα που διαθέτει.

Κεφάλαιο 7ο: Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Κατά την εκπόνηση της συγκεκριμένης Π.Α μελετήθηκαν εκτενώς τα συστήματα SIEM, οι λειτουργίες και οι αρχιτεκτονικές τους. Επίσης μελετήθηκε η στοίβα υπηρεσιών ELK που αποτελεί βάση για πολλούς μηχανισμού SIEM που χρησιμοποιούνται ευρέως. Το τελευταίο κομμάτι της Π.Α περιλάμβανε την δημιουργία ενός λειτουργικού μηχανισμού SIEM με τη χρήση της στοίβας ELK και επιπρόσθετων υπηρεσιών. Το σύστημα αυτό αξιολογήθηκε με βάση ορισμένες περιπτώσεις χρήσης.

Χρονικά, η διαδικασία της υλοποίησης του συστήματος SIEM ήταν αυτή που κράτησε περισσότερο. Είναι μια δύσκολη και περίπλοκή διαδικασία, αλλά ταυτόχρονα ενδιαφέρουσα που προσφέρει πολλές γνώσεις και δεξιότητες. Αν και στο υλοποιημένο σύστημα SIEM χρησιμοποιήθηκαν συγκεκριμένα παραδείγματα κακόβουλων ενεργειών και ένα μικρό πλήθος σταθμών η διαδικασία της δημιουργίας κανόνων και η σωστή εγκατάσταση πρακτόρων στα συστήματα ήταν ιδιαίτερα χρονοβόρα.

Αυτό δείχνει πως σε πραγματικές συνθήκες με μεγαλύτερα δίκτυα και πολύ περισσότερες πιθανές απειλές η εγκατάσταση και η ρύθμιση ενός συστήματος SIEM είναι μια ιδιαίτερα περίπλοκη διαδικασία που απαιτεί εξειδίκευση πάνω στο αντικείμενο.

Τα πρότυπα που υπάρχουν σχετικά με τις πιθανές απειλές και ευπάθειες συμβάλουν σημαντικά στη μείωση της περιπλοκότητας της διαδικασίας αυτής καθώς περιγράφουν επαρκώς τις απειλές αυτές.

7.2 Μελλοντικές επεκτάσεις

Αν και το SIEM που υλοποιήθηκε, είχε όλες τις βασικές λειτουργίες ενός SIEM, μελλοντικά θα μπορούσε να δεχθεί ενδιαφέρουσες επεκτάσεις που θα βελτίωναν τη λειτουργικότητα του και την εφαρμοσημότητα του.

7.2.1 Υλοποίηση SIEM σε Docker containers

Μια καλή επέκταση θα ήταν η υλοποίηση του συστήματος SIEM μέσα σε ένα Docker container [20]. Το Docker είναι λογισμικό προσομοίωσης επιπέδου λειτουργικού συστήματος. Τα containers προσομοιώνουν ένα ξεχωριστό λειτουργικό σύστημα. Το πλεονέκτημα με αυτή την υλοποίηση είναι η ευκολία στην εγκατάσταση του SIEM, αφού τα containers είναι ανεξάρτητα από το λειτουργικό σύστημα στο οποίο τρέχουν. Για παράδειγμα αν το συγκεκριμένο SIEM είχε υλοποιηθεί σε ένα Linux container, θα μπορούσε να μεταφερθεί σε έναν υπολογιστή ή server με λειτουργικό σύστημα Windows που θα έχει εγκατεστημένο Docker, και να τρέξει κατευθείαν εκεί.

Η αρχιτεκτονική παρουσιάζεται στην εικόνα 7.1

7.2.2 Προσθήκη προχωρημένου Alerting με τη χρήση του Elastalert

Αν και το σύστημα διαθέτει Alerts που εμφανίζονται στη διεπαφή χρήστη όταν καταγράφονται συμβάντα, κάποιος πιο προχωρημένος μηχανισμός Alerting, όπως για παράδειγμα με e-mail ή με SMS στο κινητό τηλέφωνο θα ήταν πιο αποδοτικός.

Αυτή η λειτουργία μπορεί να υλοποιηθεί από την υπηρεσία ανοιχτού κώδικα ElastAlert [32], που είναι μηχανισμός Alerting συμβατός με τη λειτουργία της στοίβας ELK.

Ο μηχανισμός αυτός είναι τρέχει με τη χρήση Python, και δίνει τη δυνατότητα στο χρήστη, να προσδιορίσει περιπτώσεις και τρόπους Alerting.

Η αλήθεια είναι ότι ο μηχανισμός αναγνώρισης συμβάντων που χρησιμοποιεί η υπηρεσία ElastAlert είναι πολύ πιο απλοϊκή και με λιγότερες δυνατότητες από αυτή που χρησιμοποιήθηκε στο SIEM της Π.Ε με τη χρήση του X-Pack. Για αυτό το λόγο θα μπορούσε να χρησιμοποιηθεί μόνο ο μηχανισμός Alerting της ElastAlert. Αυτό μπορεί να γίνει δημιουργώντας ειδικά index με τη χρήση του μηχανισμού Alerting που ήδη υπάρχει και με τη χρήση αυτών των index θα μπορούσαν να οριστούν προϋποθέσεις για Alerts μέσω της ElastAlert.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Papers

[1] Igor Kotenko and Andrey Chechulin, Common Framework for Attack Modeling and Security Evaluation in SIEM Systems 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing

[2] Oskars Podzins, Why SIEM is Irreplaceable in a Secure IT Environment?, Riga, Latvia, 2019

[3] Mike Ostrowski, From SIEM to SOC: Crossing the Cybersecurity Chasm., United States, RSA Conference, 2018

[4] S. Sandeep Sekharan and Kamalanathan Kandasamy, Profiling SIEM Tools and Correlation Engines for Security Analytics, IEEE WiSPNET 2017 conference

[5] Kai-Oliver Detken, Thomas Rix, Carsten Kleiner, Bastian Hellmann, Leonard Renners, SIEM Approach for a Higher Level of IT Security in Enterprise Networks, The 8thIEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 2015, Warsaw, Poland

[6] NIST SP800-92, "Guide to Computer Security Log Management", 2006

[7] Igor Anastasov, Danco Davcev, SIEM implementation for global and distributed environments, Skopje, 2014

[8] Kai-Oliver Detken, Marcel Jahnke, Carsten Kleiner , Marius Rohde1 DECOIT GmbH, Combining Network Access Control (NAC) and SIEM Functionality based on Open Source, 2017, Bucharest, Romania

[9] Moukafih Nabil, Sabir Soukaina, Abdelmajid Lakbabi and Orhanou Ghizlane, SIEM Selection Criteria for an efficient contextual security, 2017

[10] Hassan Mokalled, Rosario Catelli, Valentina Casola, Daniele Debertol, Ermete Meda, Rodolfo Zunino, The Applicability of a SIEM Solution: Requirements and Evaluation, 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises

[11] Rafał Leszczyna, Michał R. Wróbel, Evaluation of Open Source SIEM for Situation Awareness Platform in the Smart Grid Environment, 2015

[12] K. Agrawal and H. Makwana, "A study on critical capabilities for security information and event management,", 2015.

[13] Kelly M. Kavanagh and Mark Nicolett, "Magic quadrant for security information and event management" Published: 10 Aug. 2016

[14] Ibrahim Yahya Mohammed AL-Mahbashi, Dr. M. B. Potdar, Mr. Prashant Chauhan, Network Security Enhancement through Effective Log Analysis Using ELK, Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC)

[15] NIST SP800-128, "Guide for Security-Focused Configuration Management of Information Systems", 2006

[16] NIST SP800-171, “Enhanced Security Requirements for Protecting Controlled Unclassified Information”, 2021

[17] Farrukh Ahmed, Urooj Jahangir, Hamad Rahim, Kamran Ali, Dur-e-Shawar Agha, Centralized Log Management Using Elasticsearch, Logstash and Kibana, 2020 International Conference on Information Science and Communication Technology

[18] Sheffi Gupta, Rinkle Rani, A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases

[19] Vlad-Andrei Zamfir, Mihai Carabas, Costin Carabas, Nicolae Tapus, Systems monitoring and big data analysis using the Elasticsearch system, 2019

[20] Lei Chen, Ming Xian, Jian Liu, Huimei Wang, Docker Container Log Collection and Analysis System Based on ELK, 2020

Websites

[21] Elasticsearch, Logstash, Kibana, Beats, Elastic [Online] Available at: https://www.elastic.co

[22] OSSEC, https://www.ossec.net/

[23] Wazuh, The Open Source Security Platform, https://wazuh.com/

[24] Apache Metron, https://metron.apache.org/

[25] SIEMonster, https://siemonster.com/

[26] OASIS, Sharing threat intelligence just got a lot easier!, https://oasis-open.github.io/cti-documentation/

[27] MITRE, CVE, https://cve.mitre.org/

[28] OVAL (Open Vulnerability and Assessment Language), https://oval.mitre.org/

[29] MITRE ATT&CK https://attack.mitre.org/

[30] Github elastic repository, https://github.com/elastic

[31] OSSIM github repository https://github.com/alienfault/ossim

[32] ElastAlert - Easy & Flexible Alerting With Elasticsearchhttps://elastalert.readthedocs.io/en/latest/

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ ΡΥΘΜΙΣΕΩΝ SIEM

Αρχείο elasticsearch.yml

```
# ======================== Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: ELK_SIEM
#cluster.initial_master_nodes: ["master"]
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: master
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ---------------------------------- Paths ------------------------------------
```

```
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.1.16
#
# 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.seed_hosts: []
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module documentation.
# dev mode
discovery.type: single-node
#
# ---------------------------------- Various -----------------------------------
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
#
# ---------------------------------- Security ----------------------------------
#
#                            *** WARNING ***
#
# Elasticsearch security features are not enabled by default.
# These features are free, but require configuration changes to enable them.
# This means that users don't have to provide credentials and can get full access
# to the cluster. Network connections are also not encrypted.
#
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.
```

```
# Refer to the following documentation for instructions.

#

# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-stack-security.html

#

#enable x-pack security


xpack.security.enabled: true

xpack.security.authc.api_key.enabled: true
```

Αρχείο kibana.yml

```
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "192.168.1.16"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
server.publicBaseUrl: "localhost"

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: "http://192.168.1.16:9200"

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
kibana.index: ".kibana"
```

```
# The default application to load.
#kibana.defaultAppId: "home"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "kibana_system"
#elasticsearch.password: "pass"

# Kibana can also authenticate to Elasticsearch via "service account tokens".
# If may use this token instead of a username/password.
# elasticsearch.serviceAccountToken: "my_token"

# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# Optional settings that provide the paths to the PEM-format SSL certificate and key files.
# These files are used to verify the identity of Kibana to Elasticsearch and are required when
# xpack.security.http.ssl.client_authentication in Elasticsearch is set to required.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key

# Optional setting that enables you to specify a path to the PEM file for the certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This value
# must be a positive integer.
elasticsearch.requestTimeout: 30000

# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist configuration.
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to disable.
#elasticsearch.shardTimeout: 30000

# Logs queries sent to Elasticsearch. Requires logging.verbose set to true.
#elasticsearch.logQueries: false
```

```
# Specifies the path where Kibana creates the process ID file.
#pid.file: /run/kibana/kibana.pid

# Enables you to specify a file where Kibana stores log output.
#logging.dest: stdout

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English - en , by default , Chinese - zh-CN .
#i18n.locale: "en"
#
xpack.encryptedSavedObjects.encryptionKey: 'F3Ca6a6i1bjr9LrgJbUhVSxHwipRs4mm'
```

i