ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Σχολή Ψηφιακής Τεχνολογίας

Τμήμα Πληροφορικής και Τηλεματικής

Πρόγραμμα Μεταπτυχιακών Σπουδών Δ’ Κύκλου

Τηλεπικοινωνιακά Δίκτυα και Υπηρεσίες Τηλεματικής

Διπλωματική Εργασία

“Αυτοματοποιημένος έλεγχος ασφάλειας εσωτερικού δικτύου με χρήση NMS”

Κοτρωνούλας Νικόλαος

Αθήνα, 2017

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Σχολή Ψηφιακής Τεχνολογίας

Τμήμα Πληροφορικής και Τηλεματικής

Πρόγραμμα Μεταπτυχιακών Σπουδών Δ’ Κύκλου

Τηλεπικοινωνιακά Δίκτυα και Υπηρεσίες Τηλεματικής

Τριμελής Εξεταστική Επιτροπή

Δαλάκας Βασίλειος
ΕΔΙΠ, Τμήμα Πληροφορικής και Τηλεματικής,Χαροκόπειο Πανεπιστήμιο

Ριζομυλιώτης Παναγιώτης
Επίκουρος Καθηγητής, Πανεπιστήμιο Αιγαίου

Βαμβακάρη Μαλβίνα
Αναπληρώτρια Καθηγήτρια, Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Ο Κοτρωνούλας Νικόλαος δηλώνω υπεύθυνα ότι:

1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Στην οικογένεια μου

Πρόλογος

Η παρούσα διπλωματική εργασία υλοποιήθηκε στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών του τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο Αθηνών από τον Οκτώβρη του 2016 έως τον Φεβρουάριο του 2017.

Στα πλαίσια της διπλωματικής μου εργασίας ήρθα σε επαφή με ένα μεγάλο σύνολο νέων τεχνολογίων καθώς επίσης και με ανθρώπους που μου παρείχαν ηθική και τεχνική συμπαράσταση τους οποίους θα ήθελα να ευχαριστήσω για την ιδιαίτερη συμβολή τους.

Ειδικότερα θα ήθελα να ευχαριστήσω τον καθηγητή μου Δρ. Βασίλειο Δαλάκα, μέλος ΕΔΙΠ του τμήματος Πληροφορικής και Τηλεματικής του Χαροκόπειου Πανεπιστήμιου ο οποίος με εμπιστεύτηκε και με καθοδήγησε καθ’ όλη της διάρκεια της προσπάθειας μου καθώς επίσης και τον Ανάργυρο Τσαδήμα μέλος ΕΤΕΠ του Τμήματος Πληροφορικής και Τηλεματικής του Χαροκόπειου Πανεπιστήμιου.

Περιεχόμενα

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

NMS	Network Monitoring System
SNMP	Simple Network Management Protocol
VPNs	Virtual Private Networks
LAN	Local Area Network
WAN	Wide Area Network
IDS	Intrusion Detection Systems
IPS	Intrusion Prevention System
API	Application programming interface
DB IDO	Database Icinga Data Output
NSCA	Nagios Service Check Acceptor

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο αυτοματοποιημένος έλεγχος ασφάλειας ενός εσωτερικού δικτύου (intranet) με χρήση συστήματος παρακολούθησης δικτύου (Network Monitoring System). Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η παρουσίαση ενός συστήματος παρακολούθησης δικτύου, καθώς και ο τρόπος εγκατάστασης και παραμετροποίησής του με κατάλληλο τρόπο, ώστε να μπορεί να πραγματοποιεί ελέγχους ασφάλειας σε πιθανές παραβιάσεις με απότοκο στόχο την άμεση και αποτελεσματική ανίχνευση των επιθέσεων αυτών.

Προκειμένου να μπορεί να εφαρμοστεί ο έλεγχος ασφάλειας, η εγκατάσταση και παραμετροποίηση του συστήματος παρακολούθησης δικτύου απαιτεί τη δημιουργία και παραμετροποίηση ενός συγκεκριμένου logfile το οποίο όταν εφαρμόζεται στο σύστημα παρακολούθησης δικτύου μπορεί να ανιχνεύει πιθανές παραβιάσεις. Ειδικότερα, ο κύριος στόχος της εργασίας αυτής είναι να μπορούν να εντοπιστούν από τα log αρχεία του server πιθανές παραβιάσεις (κενά ασφαλείας) ή οποιαδήποτε ύποπτη δραστηριότητα και να γίνεται αναφορά του συμβάντος στον διαχειριστή μέσω του συστήματος παρακολούθησης δικτύου. Με τον τρόπο αυτό εκτιμάται πως μπορεί να απλοποιηθεί η διαδικασία αυτοματοποιημένης ανίχνευσης πιθανών παραβιάσεων και να επέλθει βελτίωση της απόδοσης του συστήματος.

Όσον αφορά την μέθοδο που ακολουθήθηκε, έγινε διερεύνηση ποια λογισμικά είχαν τη δυνατότητα να ελέγχουν το αρχείο καταγραφής αλλά και να συνδυάζονται με το υπάρχον κέντρο πληροφορικής και δικτύων στο Icinga. Στη συνέχεια, έγινε εγκατάσταση του συγκεκριμένου σε ένα σύστημα Debian έκδοσης 8, που είναι και το λειτουργικό σύστημα που υποστηρίζουν οι διακομιστές του κέντρου πληροφορικής. Αφού έγινε η εγκατάσταση διερευνήθηκαν οι πιθανές παράμετροι που εξηγούνται στα συμπεράσματα και έγινε προσπάθεια ενσωμάτωσης του στο Icinga.

Λέξεις κλειδιά:

Network Monitoring System, check_log_file, Icinga web 2, nmap, port scanning

Summary

The aim of this thesis is to offer an overview of a Network Monitoring System (NMS) and showcase appropriate methods for system installation and set up, so that the NMS performs automated safety checks/scans as part of a private network (intranet) that lead to timely and adequate detection for possible intrusions.

Setting up the NMS to allow such safety checks/scans requires programming and setting up of a logfile that, when applied to the NMS, can detect possible attacks. Specifically, the goal is to use the server's logfiles to allow for possible intrusions or suspicious activity ('safety gaps') to be identified and reported to the user through the NMS. We anticipate that this approach can simplify and improve automated detection of possible intrusions.

From a methodological standpoint, we examined what software was able to check the log and be combined with the existing computer center and network in Icinga. We install it on a Debian system 8, which is the operating system that supports the computer center servers. After the installation the possible parameters were investigated and explained. At the conclusions we could see an attempt to integrate it at Icinga.

Keywords:
Network Monitoring System, check_log_file, Icinga web 2, nmap, port scanning

Κεφαλαιο 1

1.1 Εισαγωγή

Τα συστήματα παρακολούθησης δικτύων (Network Monitoring Systems) διαφέρουν από τα συστήματα ανίχνευσης εισβολών (IDSs) ή τα λεγόμενα συστήματα αποτροπής εισβολών (IPSs). Ενώ τα συστήματα αυτά ανιχνεύουν εισβολές και προλαμβάνουν εχθρικές ή μη εξουσιοδοτημένες δραστηριότητες από μη εξουσιοδοτημένους χρήστες, τα συστήματα παρακολούθησης δικτύων (NMS) μας επιτρέπουν να γνωρίζουμε σε ποια κατάσταση λειτουργίας βρίσκεται το δίκτυο μας κατά τη διάρκεια των συνηθισμένων εργασιών. Η παρακολούθηση του δικτύου επιτυγχάνεται με τη χρήση διαφόρων λογισμικών ή ένα συνδυασμό λύσεων hardware plug-and-play και συσκευών λογισμικού. [2]

1.2 Η έννοια του συστήματος παρακολούθησης δικτύου

Με τον όρο παρακολούθηση δικτύου (network monitoring) ενός συστήματος πληροφορικής εννοούμε την συνεχή παρακολούθηση ενός δικτύου υπολογιστών σχετικά με το πόσο αργό είναι ή όσον αφορά την έλλειψη ορισμένων στοιχείων. Ένα network monitoring system παρακολουθεί το δίκτυο για ενδεχόμενα προβλήματα που προκαλούνται από υπερφόρτωση, εφαρμογές που έχουν υποστεί κάποια ξαφνική διακοπή, web servers ή προβλήματα που αφορούν τις συνδέσεις του δικτύου και άλλες συσκευές γενικότερα. [1]

Σχεδόν κάθε είδους δικτύου μπορεί να παρακολουθείται χωρίς να έχει σημασία αν είναι ασύρματο ή ενσύρματο, εταιρικό LAN, VPN ή παροχέας υπηρεσιών WAN. Μπορούμε να παρακολουθούμε συσκευές σε διαφορετικά λειτουργικά συστήματα με ένα πλήθος λειτουργιών, που κυμαίνονται από BlackBerries και κινητά τηλέφωνα, σε servers, routers και switches. Τα συστήματα αυτά βοηθούν στο να εντοπιστούν συγκεκριμένες δραστηριότητες και μετρήσεις απόδοσης και παράγουν αποτελέσματα που επιτρέπουν στον τρόπο αντιμετώπισης τους. [2]

1.3 Η λειτουργία του Network monitoring system

Ο τρόπος λειτουργίας των NMS (Network monitoring system), αφορά τον έλεγχο σχετικά με την δραστηριότητα και την υγεία των εσωτερικών συστημάτων του δικτύου αποστέλλοντας

ένα σήμα, το οποίο ονομάζεται ping, για τις διάφορες θύρες του συστήματος. Το σύστημα ελέγχου χρησιμοποιεί μια μεγάλη ποικιλία από διαστήματα ελέγχου, τα οποία ουσιαστικά είναι ο χρόνος μεταξύ των pings. Τα NMS έχουν τη δυνατότητα να ελέγχουν κάθε είδους πρωτοκόλλων δικτύου και ιδιαίτερα τα πρωτόκολλα διαδικτύου. Για παράδειγμα, οι υπηρεσίες παρακολούθησης ιστοσελίδων μπορούν να ελέγχουν τις σελίδες HTTP, HTTPS, SNMP, FTP, SMTP, POP3, IMAP, DNS, SSH, Telnet, SSL και TCP. Έτσι, όταν πρόκειται για web servers, ένα πρόγραμμα παρακολούθησης του δικτύου μπορεί να στείλει ένα αίτημα HTTP σε έναν web server για να καθορίσει την κατάστασή του. Αλλά όταν πρόκειται για έναν e-mail server, το λογισμικό παρακολούθησης στέλνει περιοδικά ένα μήνυμα ελέγχου μέσω SMTP, το οποίο ανακτάται από το IMAP ή POP3. Με τον τρόπο αυτό, μπορεί να μιμηθεί τη διαδρομή ενός τυπικού μηνύματος και να ελέγχεται η υγεία του δικτύου και του server μέσα απο το οποίο περνά.

Όταν ένα εργαλείο παρακολούθησης δικτύου εντοπίσει κάποιο πρόβλημα σε ένα σύστημα, μέσω ενός αποτυχημένου αιτήματος κατάστασης για παράδειγμα και δεν υπάρχει η δυνατότητα δημιουργίας σύνδεσης, τότε αυτό οδηγεί στο λεγόμενο time out. Σε περιπτώσεις όπου υπάρχει μια αποτυχία αιτήματος κατάστασης, το σύστημα παρακολούθησης του δικτύου θα παράγει μια ενέργεια. Οι ενέργειες αυτές ποικίλλουν, καθώς μπορούν να ενημερώσουν τον διαχειριστή του δικτύου με μια ειδοποίηση, ένα SMS κειμένου, ένα μήνυμα τηλεειδοποίησης ή ένα email. [1]

1.4 Βασικές τεχνικές παρακολούθησης

Ορισμένες απο τις βασικές τεχνικές που είναι διαθέσιμες για την παρακολούθηση του δικτύου οι οποίες χρησιμοποιούνται για τη συλλογή των δεδομένων παρακολούθησης από το δίκτυο αναφέρονται στη συνέχεια και είναι οι εξής:

- **Ping**

  Είναι ένα εργαλείο διαχείρισης του δικτύου που χρησιμοποιείται για τον έλεγχο της προσβασιμότητας και της διαθεσιμότητας του κεντρικού υπολογιστή σε ένα δίκτυο IP. Τα δεδομένα από τα αποτελέσματα αυτά καθορίζουν εάν το πλήθος του δικτύου είναι ενεργό ή όχι. Επιπλέον, μετράει το χρόνο μετάδοσης και την απώλεια πακέτων κατά την επικοινωνία με έναν κεντρικό υπολογιστή.

- **SNMP (Simple Network Management Protocol)**

Το SNMP είναι ένα πρωτόκολλο διαχείρισης δικτύου που χρησιμοποιείται για την ανταλλαγή πληροφοριών μεταξύ των κόμβων σε ένα δίκτυο όπου περιλαμβάνει το λογισμικό παρακολούθησης του δικτύου. Είναι το πιο ευρέως χρησιμοποιημένο πρωτόκολλο για τη διαχείριση και την παρακολούθηση των δικτύων και περιλαμβάνει τα παρακάτω στοιχεία:

Διαχείριση της συσκευής: Ο κόμβος του δικτύου που υποστηρίζει το SNMP και την πρόσβαση σε συγκεκριμένες πληροφορίες.

Agent: Λογισμικό που αποτελεί μέρος της υπό παρακολούθηση συσκευής. Είναι ένας ενδιάμεσος που έχει πρόσβαση στο MIB (διαχείριση πληροφοριών βάσεων δεδομένων) της συσκευής και επιτρέπει στα συστήματα NMS να διαβάσουν και να γράψουν στο MIB.

Network Management System (NMS): Είναι μια εφαρμογή σε ένα σύστημα που παρακολουθεί και ελέγχει τις διαχειριζόμενες συσκευές μέσω του ενδιάμεσου χρησιμοποιώντας τις εντολές SNMP.

- **Syslog**

  Το Syslog (δεν πρέπει να συγχέεται με τα Windows Eventlog), είναι ένα σύστημα καταγραφής μηνυμάτων που επιτρέπει σε μια συσκευή να στέλνει ειδοποιήσεις συμβάντων σε δίκτυα IP. Οι πληροφορίες από αυτά τα μηνύματα μπορούν να χρησιμοποιούνται για τη διαχείριση του συστήματος, καθώς και τον έλεγχο της ασφάλειας.

- **Leveraging the power of scripts**

  Σε δίκτυα όπου το NMS δεν είναι διαθέσιμο για την παρακολούθηση τους, ή το υπάρχον NMS δεν υποστηρίζει συγκεκριμένες λειτουργίες, οι διαχειριστές του δικτύου μπορούν να κάνουν χρήση σεναρίων (scripts). Τα σενάρια χρησιμοποιούν κοινές εντολές, όπως την ping, netstat, lynx, snmpwalk, κλπ, οι οποίες υποστηρίζονται από τα περισσότερα στοιχεία του δικτύου για να εκτελέσουν μια ενέργεια, όπως η συλλογή πληροφοριών από τα στοιχεία, κάνοντας αλλαγές σε ρυθμίσεις της συσκευής, ή να εκτελέσουν μια προγραμματισμένη εργασία. Τα σενάρια Bash, Perl, κλπ είναι κοινά εργαλεία scripting που χρησιμοποιούνται από τους διαχειριστές του δικτύου. [3]

1.5 Είδη δικτύων και έλεγχος του NMS

Τα δίκτυα στα οποία γίνεται η χρήση του NMS είναι τα ασύρματα ή ενσύρματα δίκτυα, τα εταιρικά τοπικά δίκτυα (LAN), τα εικονικά ιδιωτικά δίκτυα (VPNs) και τα δίκτυα ευρείας

περιοχής ενός παρόχου υπηρεσιών (WAN). Η χρήση του NMS μας διευκολύνει όσον αφορά τα σύνθετα περιβάλλοντα, εκδίδοντας αναφορές όπου οι διαχειριστές κάνουν χρήση για να επιβεβαιώσουν τις ρυθμίσεις και τις πολιτικές συμμόρφωσης, να επισημάνουν πιθανή εξοικονόμησης κόστους από την εύρεση περιττών πόρων, για παράδειγμα, μπορούν να επιλύουν προβλήματα αποδοτικότητας, βοηθούν στον προσδιορισμό της παραγωγικότητας των εργαζομένων, επισημαίνουν την υπερφόρτωση του εξοπλισμού που αυτό ενδεχομένως να ρίξει το δίκτυο, προσδιορίζουν τις αδυναμίες των δικτύων ευρείας περιοχής και άλλα σημεία συμφόρησης, Measure latency, ή την καθυστερημένη μεταφορά δεδομένων και εντοπίζουν ανωμαλίες στην εσωτερική κίνηση που θα μπορούσαν να υποδείξουν μια απειλή για την ασφάλεια. [2]

Όπως αναφέραμε και πιο πάνω, ένα NMS δεν είναι ένα σύστημα ανίχνευσης εισβολής (IDS) ή ένα σύστημα αποτροπής εισβολών (IPS). Το NMS μπορεί να ανιχνεύσει ενοχλητικές ενέργειες, αλλά δεν είναι αυτή η αποστολή του.
Όσον αφορά τον έλεγχο ένα NMS ελέγχει συγκεκριμένα:

- Τον χώρο στο δίσκο, τον επεξεργαστή και την χρήση της μνήμης.
- Τα αρχεία καταγραφής (για τυχόν λάθη ή άλλο θέμα).
- Την αξιοποίηση του δικτύου και το εύρος ζώνης.
- Σημαντικές υπηρεσίες και διαδικασίες.
- Την εσωτερική ή εξωτερική διαθεσιμότητα ιστοσελίδων.

Ένα καλό σύστημα παρακολούθησης θα εξετάσει μια συσκευή για ένα συγκεκριμένο σύνολο στατιστικών στοιχείων, θα διατηρήσει αυτά τα δεδομένα και θα υποβάλει μια έκθεση στον κατάλληλο διαχειριστή αν αυτά τα στατιστικά στοιχεία υπερβαίνουν το αποδεκτό όριο αν π.χ. ο δίσκος είναι 90% πλήρης, θα ενημερώσει τον διαχειριστή μέσω ενός e-mail, έτσι ώστε να μπορεί να διορθώσει την κατάσταση. **[4]**

1.6 Το πρωτόκολλο SNMP

Το SNMP είναι ένα πρωτόκολλο του επιπέδου εφαρμογών του οποίο διευκολύνει την ανταλλαγή πληροφοριών διαχείρισης μεταξύ των συσκευών του δικτύου. Είναι μέρος του TCP/IP και επιτρέπει στους διαχειριστές να παρακολουθούν την απόδοσή του και να επιλύουν τα προβλήματα που εμφανίζονται. Το SNMP έχει τρία βασικά στοιχεία: διαχειριζόμενες

συσκευές, πράκτορες (agents) και συστήματα διαχείρισης δικτύου (Network Management Systems- NMS).

Μια διαχειριζόμενη συσκευή είναι ένας κόμβος του δικτύου ο οποίος περιέχει ένα SNMP πράκτορα και βρίσκεται μέσα στο διαχειριζόμενο δίκτυο. Οι συσκευές αυτές συλλέγουν και αποθηκεύουν πληροφορίες και τις διαθέτουν στο σύστημα διαχείρισης του δικτύου με χρήση του SNMP. Τέτοιες συσκευές είναι οι δρομολογητές, οι γέφυρες και οι διακόπτες. Ένας πράκτορας έχει γνώση των τοπικών πληροφοριών διαχείρισης και τις μετατρέπει σε μορφή που να είναι συμβατή με το SNMP. Στη συνέχεια ένα σύστημα διαχείρισης δικτύου εκτελεί εφαρμογές οι οποίες παρακολουθούν και ελέγχουν τις διαχειριζόμενες συσκευές. Το σύστημα διαχείρισης του δικτύου προσφέρει τον κύριο όγκο των πόρων επεξεργασίας που απαιτούνται για τη διαχείριση.

Οι διαχειριζόμενες συσκευές ελέγχονται και παρακολουθούνται με τέσσερις βασικές SNMP εντολές: read, write, trap, traversal operations (διαδικασίες εξέτασης).

- Η εντολή read χρησιμοποιείται από το NMS για την παρακολούθηση των συσκευών. Το NMS εξετάζει διάφορες μεταβλητές που διατηρούνται από τις διαχειριζόμενες συσκευές.
- Η εντολή write χρησιμοποιείται για τον έλεγχο των διαχειριζόμενων συσκευών και αλλάζει τις τιμές των μεταβλητών που αποθηκεύονται σε αυτές.
- Η εντολή trap χρησιμοποιείται από τις διαχειριζόμενες συσκευές για την ασύγχρονη ενημέρωση του NMS. Όταν συμβαίνουν κάποια συγκεκριμένα γεγονότα, η συσκευή στέλνει ένα trap στο NMS.

Το πρωτόκολλο που χρησιμοποιείται ευρύτατα για τη διαχείριση σε TCP/IP δίκτυα είναι το Simple Network Management Protocol (SNMP). Πιο εξελιγμένες εκδόσεις του SNMP αποτελούν η SNMPv1 η SNMPv2 και η SNMPv3.[5]

Κεφάλαιο 2

2.1 Λογισμικό ανοιχτού κώδικα

Οι εφαρμογές που παρέχονται μέσω του συστήματος Linux και των λογισμικών ανοιχτού κώδικα είναι πάρα πολλές. Εκτελούν την διαχείριση αιτημάτων των πελατών/χρηστών με μεγάλη αποτελεσματικότητα. Ο χρήστης έχει την δυνατότητα να διαμορφώσει τον πηγαίο κώδικα βάσει των απαιτήσεών του χωρίς να επιβαρύνεται με οποιαδήποτε επιπλέον χρέωση.

Επίσης είναι το ίδιο αξιόπιστα με τα αντίστοιχα εμπορικά πακέτα με επιπλέον ανταγωνιστικά χαρακτηριστικά και είναι ευρέος διαδεδομένα καθώς χρησιμοποιούνται απο πολλές επιχειρήσεις ανά τον κόσμο. Στη συνέχεια ακολουθεί μια σύντομη περιγραφή ορισμένων εκ των δημοφιλέστερων λειτουργικών όσον αφορά την παρακολούθηση των συσκευών, των υπηρεσιών, των θυρών, των πρωτοκόλλων και την ανάλυση της κυκλοφορίας του δικτύου.[6]

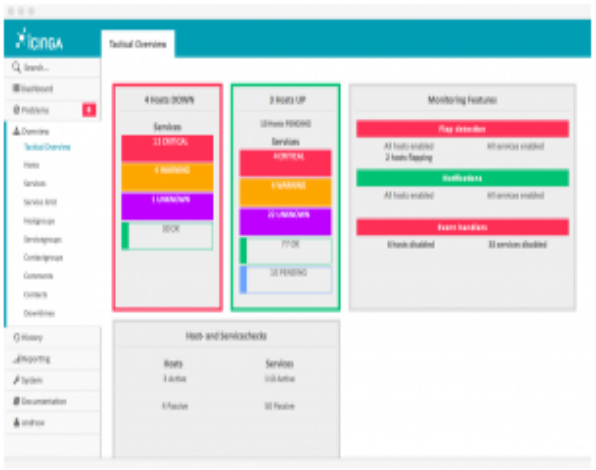
GFI LanGuard Χρησιμοποιείται για την σάρωση τόσο μικρών όσο και μεγάλων δικτύων, ως προς την αναζήτηση τρωτών σημείων του λογισμικού και για εφαρμογές χωρίς άδεια. Οι πληροφορίες προέρχονται απο περισσότερες απο 60.000 συσκευές που τρέχουν σε Windows, Mac OS ή Linux και εμφανίζονται σε μια κεντρική οθόνη, έτσι είμαστε σε θέση να δούμε την κατάσταση του συνόλου του δικτύου μας ανά πάσα στιγμή και από οποιαδήποτε τοποθεσία.	

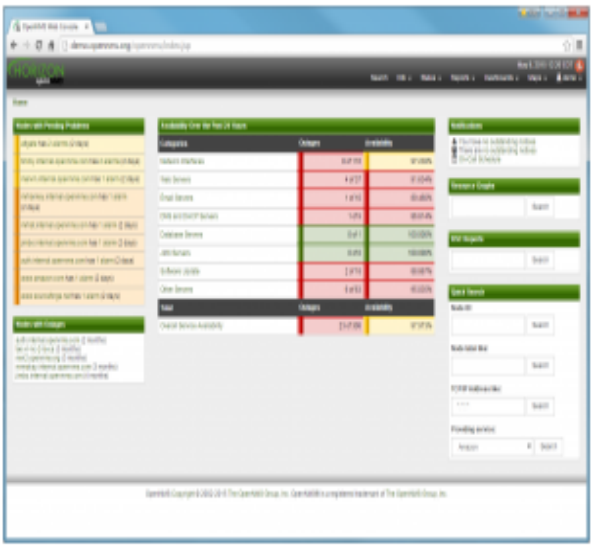
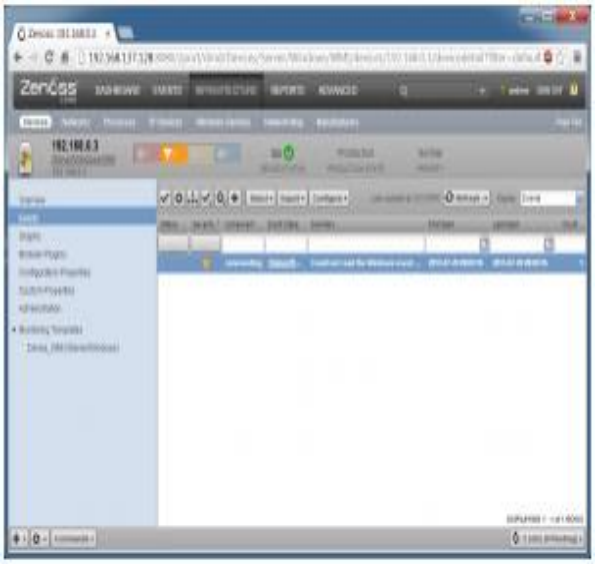
Nagios

Αποτελεί ισχυρό εργαλείο παρακολούθησης του δικτύου που μας βοηθά στο να εξασφαλιστεί ότι τα κρίσιμα συστήματα, οι εφαρμογές και οι υπηρεσίες μας θα είναι πάντα σε λειτουργία. Παρέχει δυνατότητες, όπως προειδοποιήσεις, χειρισμούς και υποβολή εκθέσεων. Το Nagios Core είναι η καρδιά της εφαρμογής που περιέχει τον πυρήνα παρακολούθησης και ένα βασικό web UI. Στην κορυφή του πυρήνα, μπορούνε να εφαρμοστούν τα plugins που θα μας επιτρέψουν να παρακολουθούμε τις υπηρεσίες, τις εφαρμογές και τις μετρήσεις με οπτικοποίηση των δεδομένων όπως γραφήματα, καθώς επίσης υποστηρίζει και βάση δεδομένων MySQL.

Icinga web 2

Το Icigna είναι μια Linux based εφαρμογή παρακολούθησης πλήρως ανοικτού κώδικα που ελέγχει τη διαθεσιμότητα των πόρων του δικτύου και ειδοποιεί αμέσως τους χρήστες όταν κάτι δεν λειτουργεί σωστά. Παρέχει δεδομένα επιχειρηματικής ευφυΐας για την σε βάθος ανάλυση και μια ισχυρή διασύνδεση γραμμής εντολών.



OpenNMS Το OpenNMS είναι μια ανοιχτού κώδικα εφαρμογή διαχείρισης της ποιότητας του δικτύου που προσφέρει αυτοματοποιημένη ανακάλυψη, διαχείριση συμβάντων και ειδοποιήσεις, μέτρηση των επιδόσεων και χαρακτηριστικά αξιοπιστίας των υπηρεσιών. Περιλαμβάνει μια εφαρμογή-πελάτη για το iPhone, iPad ή iPod Touch, δίνοντάς μας τη δυνατότητα να βλέπουμε τους κόμβους, ειδοποιήσεις και ένα περιβάλλον παρακολούθησης.	
Zenoss Core Ένα άλλο πρόγραμμα ανοιχτού κώδικα είναι το Zennos core το οποίο δίνει στους διαχειριστές δικτύου μια ολοκληρωμένη, λύση για την παρακολούθηση και διαχείριση όλων των εφαρμογών, servers, αποθήκευσης, δικτύωσης εξαρτημάτων, εργαλεία εικονικά, καθώς και άλλα στοιχεία της υποδομής των επιχειρήσεων. Είναι γραμμένο σε Python 90%, Java 10% και εκδόθηκε σύμφωνα με τους όρους της GNU General Public License έκδοση 2 και είναι ελεύθερο λογισμικό.	

Πίνακας 1: Δημοφιλέστερα συστήματα παρακολούθησης

2.2 Επιλογή λογισμικού

Στην παρούσα διπλωματική εργασία επιλέξαμε να χρησιμοποιήσουμε το λογισμικό icinga web 2. Αυτό που ξεχώρισε το icinga web 2 από τα υπόλοιπα συστήματα ανοιχτού κώδικα είναι το γεγονός ότι αποτελεί το τελευταίο λειτουργικό που κυκλοφόρησε στην αγορά εργασίας (Φεβρουάριος 2015) καθώς παρουσιάζει τις περισσότερες δυνατότητες σε σύγκριση με τα

υπόλοιπα. Η συγκεκριμένη διεπαφή διαθέτει εντελώς νέο σχεδιασμό και πολλές βελτιώσεις φιλικές προς τον χρήστη συγκριτικά με τα υπόλοιπα λειτουργικά γεγονός που ικανοποιεί έναντι των υπολοίπων.

Το Icinga web 2 είναι συμβατό με όλα τα υπάρχοντα plugins, διεπαφές χρήστη (π.χ.. Classic UI, Icinga Web) και addons. Έχει σχεδιαστεί για να είναι πιο εύκολο στην εγκατάσταση, και είναι αρκετά πιο γρήγορο και ισχυρό. Όπως διαπιστώνουμε παρακάτω παρέχει:

- Απλή εγκατάσταση: Σε αντίθεση με τους προκατόχους του, το Icinga 2 έρχεται με IDO, Livestatus, performance data and Graphite writers, σε συμβατότητα με τα χαρακτηριστικά των σχετικών βιβλιοθηκών τους.
- Εύκολη και αποτελεσματική add-on ενσωμάτωση: Το Icinga 2 διαθέτει πολλαπλά backends, για οποιοδήποτε addon και μπορεί να ενσωματωθεί εύκολα. Έτσι, μπορουμε να διαθέτουμε σε πραγματικό χρόνο γραφικές παράστασης παρακολούθησης. Επίσης, αποσκοπεί στη μείωση φορτίου του συστήματος, όπου τα δεδομένα που δεν είναι γραμμένα διαγράφονται στο σκληρό δίσκο πάνω από ένα script όπως συμβαίνει στο Icinga 1 και Nagios.
- Προσανατολισμένη Απόδοση: Το Icinga 2 είναι φτιαγμένο για να είναι γρήγορο. Χάρη στον πολυνηματικό σχεδιασμό του, μπορεί να τρέξει χιλιάδες ελέγχους κάθε δευτερόλεπτο χωρίς καμία ένδειξη ενόχλησης στην CPU του. Αυτό συμβαίνει ακριβώς για να είμαστε σίγουροι, ότι έχουμε ενσωματωμένα συμπλέγματα ελέγχου που παράγουν δεδομένα απόδοσης για το Icinga2 για διάφορα περιστατικά ή για μία ομάδα χρηστών. [9]

2.3 Icinga web 2

Το Icinga 2 ξεκίνησε τον Φεβρουάριο του 2015 ως επέκταση του Icinga το οποίο αποτελούσε εφαρμογή παρακολούθησης του συστήματος Nagios. Αποτελεί επέκταση του icinga η οποία ξαναγράφτηκε για να παρέχει στους χρήστες της ένα σύγχρονο περιβάλλον εργασίας και υποστήριξης με πολλαπλές βάσεις δεδομένων. Υποστηρίζει γραφήματα, παρέχοντας στους διαχειριστές γραφικά απόδοσης σε πραγματικό χρόνο. Οι διαχειριστές μπορούν να δουν τα φίλτρα, και να ιεραρχήσουν τα προβλήματα, για την παρακολούθηση των οποίων έχουν ήδη ληφθεί μέτρα. Μια νέα άποψη διεπαφής , επιτρέπει στους διαχειριστές να βλέπουνε hosts και services σε μία σελίδα. Είναι γραμμένο σε c++ και τρέχει σε Apache web

Server. Οι βάσεις δεδομένων που υποστηρίζει είναι MySql, PostgreSql, Oracle, IDODB. Επίσης παρέχει API (application programming interface - διεπαφή προγραμματισμού εφαρμογών) που επιτρέπει στους διαχειριστές να ενσωματώσουν πολλές επεκτάσεις χωρίς πολύπλοκες τροποποίηση του πυρήνα Icinga.

Το Icinga 2 έχει την δυνατότητα να εγκατασταθεί σε windows, Linux, Mac OS X, και άλλα συστήματα που είναι Unix-like. Κατάλληλα προκατασκευασμένα πακέτα είναι διαθέσιμα για Debian/Ubuntu, Fedora/RHEL/CentOS 6, OpenSUSE/SLES και ArchLinux. Παρέχει επίσης την δυνατότητα χρήσης σε κινητές συσκευές οι οποίες είναι διαθέσιμες σε iOS, Android, BlackBerry Tablet OS και webOS καθώς υπάρχει η εφαρμογή που επιτρέπει την πρόσβαση από εξουσιοδοτημένους χρήστες μέσω κινητών συσκευών τους. Εκδόθηκε σύμφωνα με τους όρους της GNU General Public License έκδοση 2 και αποτελεί ελεύθερο λογισμικό όπως επίσης είναι και ίσως το μόνο που είναι διαθέσιμο σε 21 γλώσσες εκτός των Αγγλικών. Τέλος, άξιο αναφοράς χρίζει το γεγονός ότι παρέχει στο διαδίκτυο πληθώρα πληροφοριών σχετικά με την εγκατάσταση και την παραμετροποίηση του. [7]

Σε αντίθεση με τις προηγούμενες εκδόσεις, η διεπαφή αυτή έχει μια λεπτομερή εικόνα για την υπηρεσία υποδοχής και για έλεγχους που δεν εκτελούνται στο χρόνο. Έχει βελτιωμένη την υπηρεσία ενσωμάτωσης υπηρεσίες καταλόγου Active directory και σε άλλους διακοσμητές LDAP. Το icinga web 2 υποστηρίζει την φόρτωση χρηστών, ομάδες χρηστών καθώς και ομάδες των μελών αυτών όπου έλεγχος φόρτωσης-ταυτότητας γίνεται από το active directry. Επιπλέον, σημαντική πρόοδος είναι η δημιουργία των υπηρεσιών hosts και servers. Μια μεγάλη ευκαιρία είναι η δημιουργία host συστημάτων και υπηρεσιών εξυπηρέτησης αντί να προσδιορίζονται μέσω των επιβεβαιωμένων αρχείων του Icigna κάθε χρήστης τώρα μπορεί να τα δημιουργήσει μέσω της διεπαφής του δικτύου και ακόμα να τα κοινοποιήσει σε άλλους. Η διαμόρφωση μας επιτρέπει να προσθέσουμε ενέργειες μόνο σε ορισμένα hosts και servers αλλά υποστηρίζει μακροεντολές για hosts, service name για προσαρμοσμένες μεταβλητές.

Τέλος η νέα δικτυακή διεπαφή παρέχει ένα πολύ βασικό API (Application programming interface- Διεπαφή προγραμματισμού εφαρμογών) για τον προγραμματισμό αφαίρεσης host και εξυπηρέτηση σε νεκρούς χρόνους. Βασικός έλεγχος ταυτότητας για την πρόσβαση είναι επίσης ένα νέο χαρακτηριστικό που διευκολύνει τη χρήση του API. Για το μέλλον, μπορούμε να

περιμένουμε περισσότερες δράσεις API και τη δημιουργία ενοτήτων για την ενσωμάτωση διαφόρων δημοφιλών εργαλείων για την ενίσχυση της στοίβας DevOps.[8]

2.4 Εικονικές μηχανές και λειτουργικά συστήματα

Όσον αφορά την εγκατάσταση και παραμετροποίηση του icinga web 2, αρχικά έγινε χρήση ενός λογισμικού το οποίο θα μας επέτρεπε την δημιουργία ενός εικονικού περιβάλλοντος (λειτουργικού συστήματος) ίδιου με το περιβάλλον του server του Χαροκοπείου, δηλαδή μια εικονική μηχανή. Τα δημοφιλέστερα λογισμικά γι' αυτό τον σκοπό είναι το Virtualbox (https://www.virtualbox.org/) της Oracle καθώς και το VMware player (http://www.vmware.com/) της VMware Inc. Και τα δύο λογισμικά διατίθενται δωρεάν, οι διαφορές μεταξύ τους είναι ελάχιστες και πολλές κριτικές αναφέρουν το VirtualBox ως πιο γρήγορο στην απόδοση του.

Επιλέξαμε να χρησιμοποιήσουμε το VirtualBox το οποίο μπορείς κανείς να το βρεί και να το εγκαταστήσει από την ιστοσελίδα https://www.virtualbox.org/wiki/Downloads. Μέσω αυτού του προγράμματος δημιουργήσαμε ένα εικονικό μηχάνημα (virtual machine). Σε αυτό το εικονικό μηχάνημα εγκαταστήσαμε το λειτουργικό σύστημα Debian8 (64 bit) την έκδοση Jessie, η οποία είναι η τελευταία και θα υποστηρίζεται για τα επόμενα 5 χρόνια την οποία και κατεβάσαμε από την ιστοσελίδα https://www.debian.org/releases/stable/debian-installer/. Την τελευταία έκδοση του Icinga web 2 μπορεί να την βρει κάποιος στην επίσημη ιστοσελίδα του, https://www.icinga.org/download/. Στη συνέχεια στην ενότητα που ακολουθεί πρόκειται να παρουσιαστούν αναλυτικά τα βήματα που ακολουθήθηκαν και οι εντολές που εκτελέστηκαν με σκοπό την εγκατάσταση της εφαρμογής και των προαπαιτούμενων της.

2.5 Διαδικασία εγκατάστασης του Icinga web 2

Αρχικά εγκαθιστούμε το icinga 2 το οποίο για να γίνει σωστά θα πρέπει να επιλέξουμε τα επίσημα "package repositories" τα οποία θα ταιριάζουνε με το λειτουργικό μας σύστημα και τα οποία τα επιλέγουμε από το http://packages.icinga.org/debian/. Στη συνέχεια, προσθέτουμε το κατάλληλο repository package για το λειτουργικό σύστημα μας το οποίο όπως είπαμε είναι το debian 8 Jessie. Πριν από αυτό όμως, έχουμε σιγουρευτεί ότι ανοίξαμε το τερματικό του root η αλλιώς κάναμε su- στο απλό τερματικό του debian και γράφουμε τις παρακάτω εντολές:

- wget -O - https://debmon.org/debmon/repo.key 2>/dev/null | apt-key add -
- echo 'deb http://debmon.org/debmon debmon-jessie main' >/etc/apt/sources.list.d/debmon.list
- apt-get update

Στην συνέχεια κάνουμε εγκατάσταση του icinga 2 με την ακόλουθη εντολή:

- apt-get install icinga2

Για τον έλεγχο των εξωτερικών υπηρεσιών στο icinga 2 χρειάζεται να τρέξουμε ορισμένα plugins τα οποία μπορούν να χρησιμοποιηθούν από το icinga 2, με την ακόλουθη εντολή:

- apt-get install nagios-plugins

Στη συνέχεια γίνεται εκκίνηση της λειτουργίας του icinga 2 με την ακόλουθη εντολή:

- /etc/init.d/icinga2 start

Το επόμενο βήμα αφορά την εγκατάσταση του icinga web 2. Το DB IDO (Database Icinga Data Output) διαμορφώνει την εξαγωγή όλων των ρυθμίσεων και την κατάσταση των πληροφοριών σε μια βάση δεδομένων και υποστηρίζεται μόνο από τη MySQL και τη PostgreSQL. Εμείς επιλέξαμε να χρησιμοποιήσουμε τη MySQL. Για το λόγο αυτό κάνουμε εγκατάσταση τη MySQL με την παρακάτω εντολή (κατά την εγκατάσταση θα μας εμφανίσει κάποια παράθυρα και θα χρειαστεί να βάλουμε ένα κωδικό για να δημιουργήσουμε τον root λογαριασμό):

- apt-get install mysql-server mysql-client

Έπειτα χρειάζεται να εγκαταστήσουμε το IDO module της MySQL οπότε τρέχουμε την εντολή:

- apt-get install icinga2-ido-mysql

Θα εμφανιστούν τέσσερα νέα παράθυρα όπου το πρώτο μας ρωτάει αν θέλουμε να ενεργοποιήσουμε τη λειτουργία Ido-mysql και επιλέγουμε *NO*, διότι θα ενεργοποιηθεί αμέσως μετά. Στο δεύτερο παράθυρο μας ρωτάει για την επιβεβαίωση της MySQL όπου και επιβεβαιώνουμε επιλέγοντας *YES* καθώς στο τρίτο παράθυρο μας ζητάει κωδικό πρόσβασης που ορίσαμε για την MySQL και στο τέταρτο παράθυρο την επιβεβαίωση του. Μόλις

ολοκληρωθεί η ρύθμιση, μπορούμε να εντάξουμε το αρχείο Ido-mysql.conf. Θα πρέπει να έχουμε σημειώσει τις διαφορετικές ρυθμίσεις διότι θα μας χρησιμεύσει για τη διαμόρφωσή του web interface.

Εγκατάσταση Ido-MySQL

```
cat /etc/icinga2/features-available/ido-mysql.conf
user= ”icinga”
password= ”******”
host= ”localhost”
database= “icinga1”
```

Μετά την ενεργοποίηση της Ido-MySQL κάνουμε επανεκκίνηση το Icinga 2 με την εντολή:

- service icinga2 restart

και συνεχίζουμε με την εγκατάσταση του web server με την εντολή:

- apt-get install apache2

Επειδή web διεπαφές και άλλα addons του icinga είναι σε θέση να στείλουν εντολές στο icinga 2 μέσω του External Command Pipe εμείς το ενεργοποιούμε με την εντολή:

- icinga2 feature enable command

Στη συνέχεια κάνουμε restart το icinga 2 όπως θα μας ζητηθεί και συνεχίζουμε με την εγκατάσταση του icinga web 2 με την εντολή:

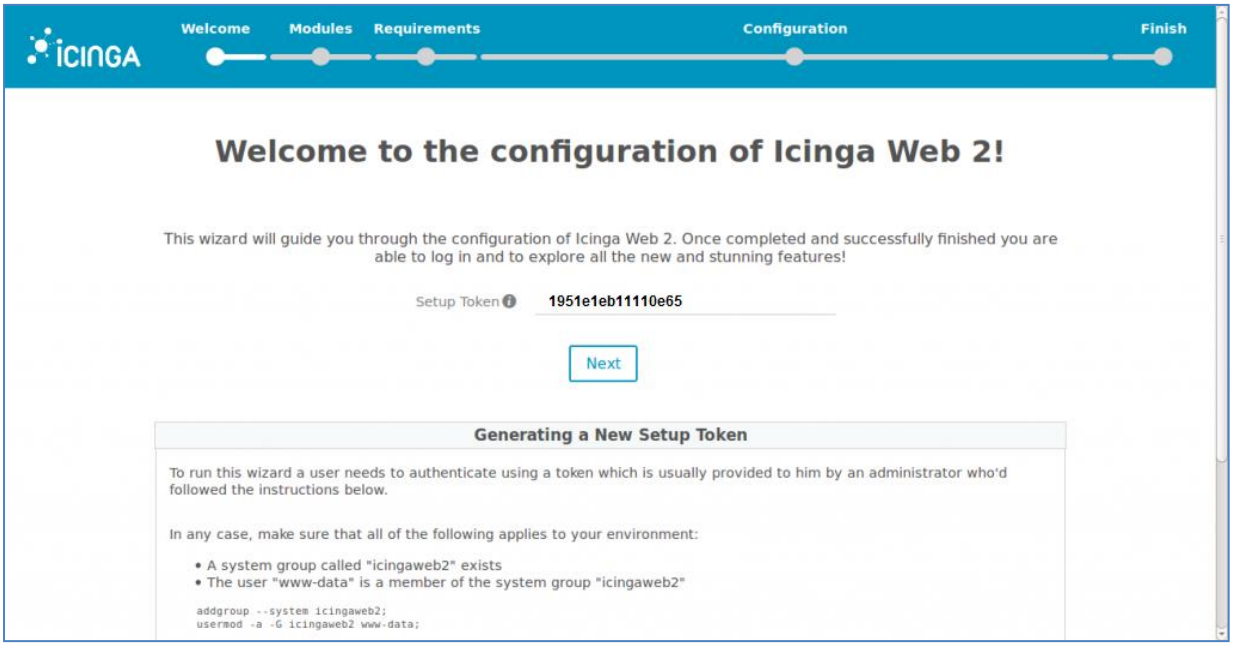
- apt-get install icingaweb2

Σχετικά με την εγκατάσταση του icinga web 2 έχουμε την δυνατότητα να την κάνουμε με δύο τρόπους μέσω web setup και μέσω source στην περίπτωση μας την κάναμε με web setup οπότε τα βήματα που θα ακολουθηθούν είναι για τον τρόπο αυτό.

Ξεκινώντας χρειαζόμαστε ένα token πιστοποίησης το οποίο το παίρνουνε με την εντολή:

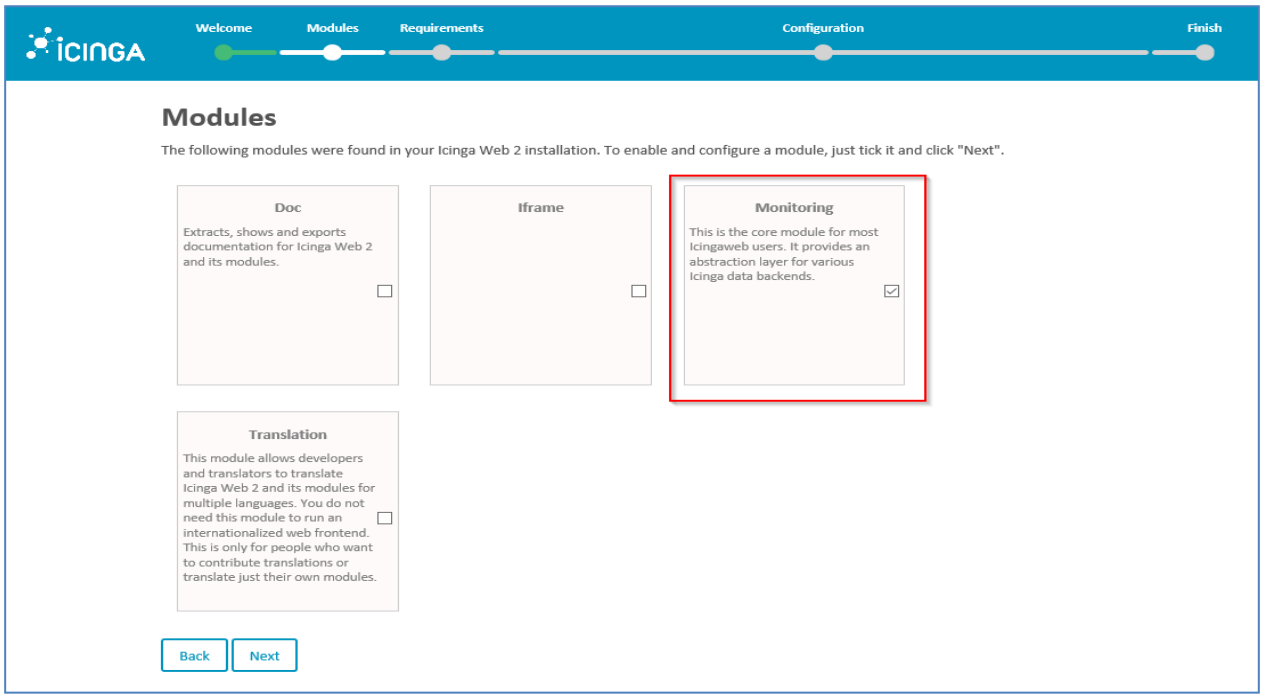
- icingacli setup token create

Στη συνέχεια ανοίγουμε τον browser μας και πηγαίνουμε στη διεύθυνση localhost/icingaweb2/setup. Πληκτρολογούμε τον κωδικό token όπως φαίνεται στην εικόνα και πατάμε next.[10]



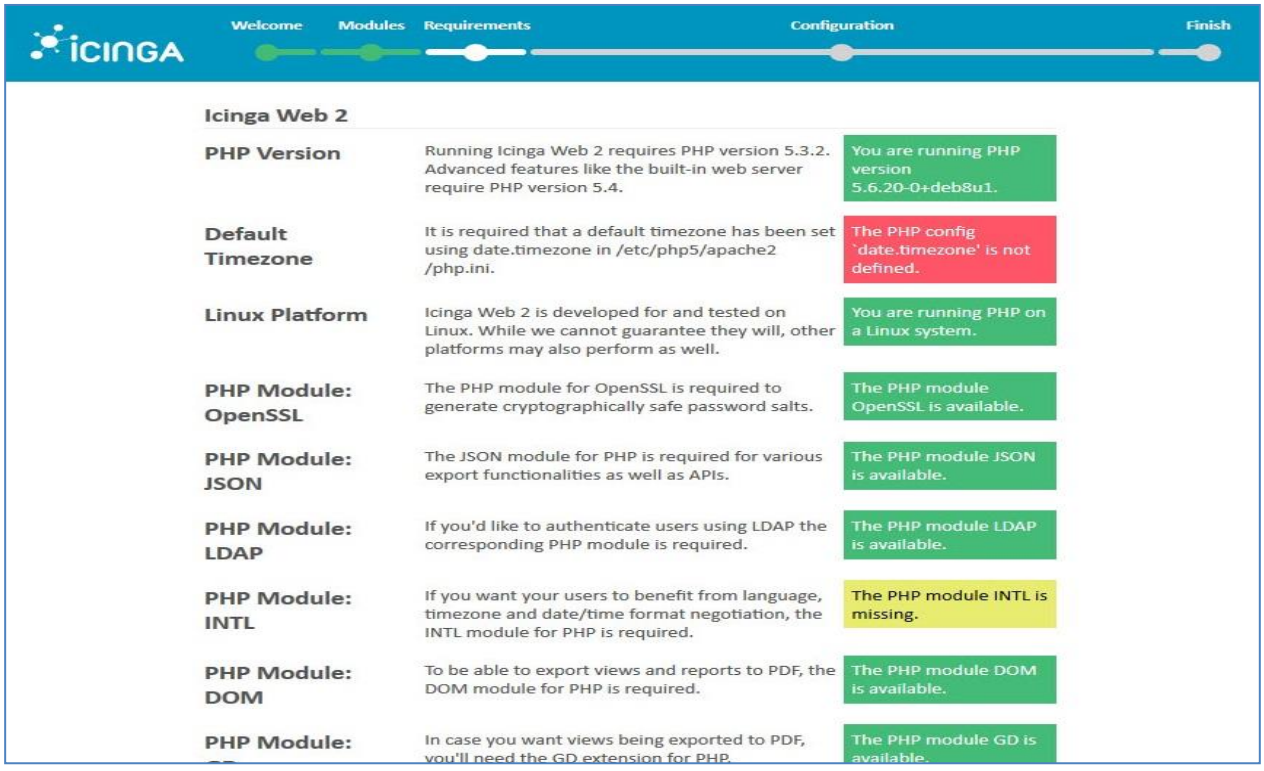
Εικόνα 1: Αρχικό περιβάλλον διεπαφής

Στην επόμενη σελίδα επιλέγουμε τα modules που θέλουμε στη δική μας περίπτωση το monitoring και πατάμε next.



Εικόνα 2: Επιλογή modules

Τώρα είναι το κύριο τμήμα για να ρυθμίσουμε όλες τις απαιτήσεις πριν από τη μετάβαση στο επόμενο βήμα.

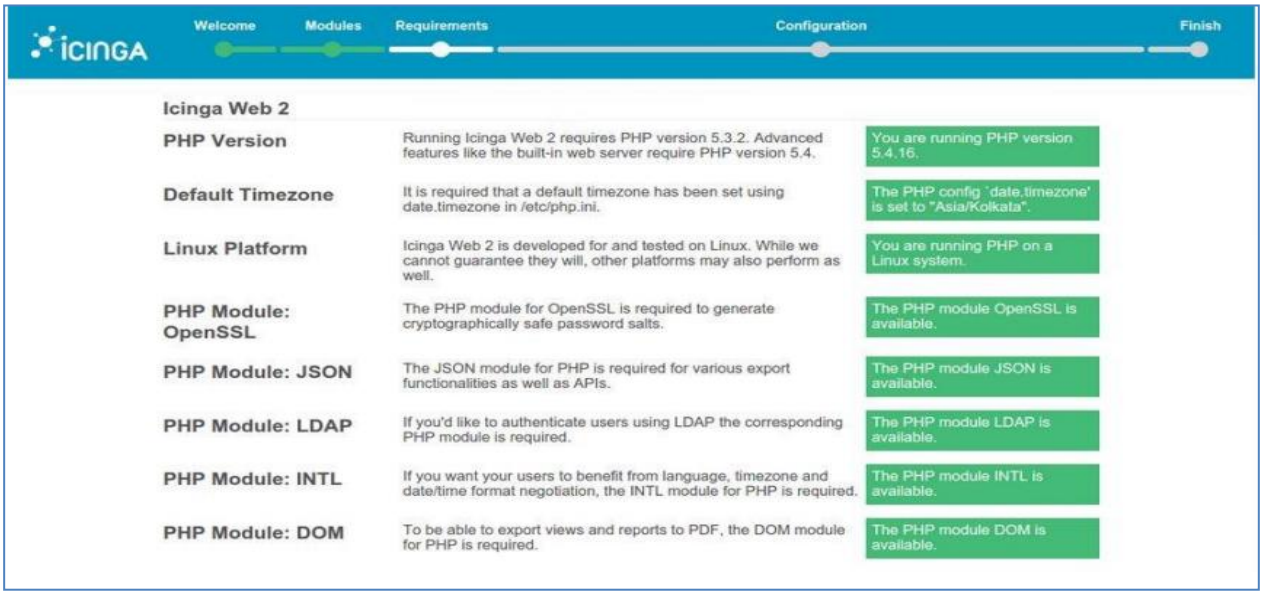


Εικόνα 3: Ρυθμίσεις λογισμικού

Όσον αφορά την διόρθωση του default timzone error που μας εμφανίζει χρειάζεται να ανοίξουμε το αρχείο που μας υποδεικνύει και να αλλάξουμε την date.timezone σε date.timezone ='Europe/Athens', αποθηκεύουμε και τρέχουμε την ακόλουθη εντολή ώστε να σβήσουμε τα warnings:

- • apt-get install php5-json php5-gd php5-imagick php5-pgsql php5-intl.

Κάνουμε refresh την σελίδα μας για να πάρει τις αλλαγές που κάναμε και να μας τις εμφανίσει με πράσινο χρώμα και πατάμε next.

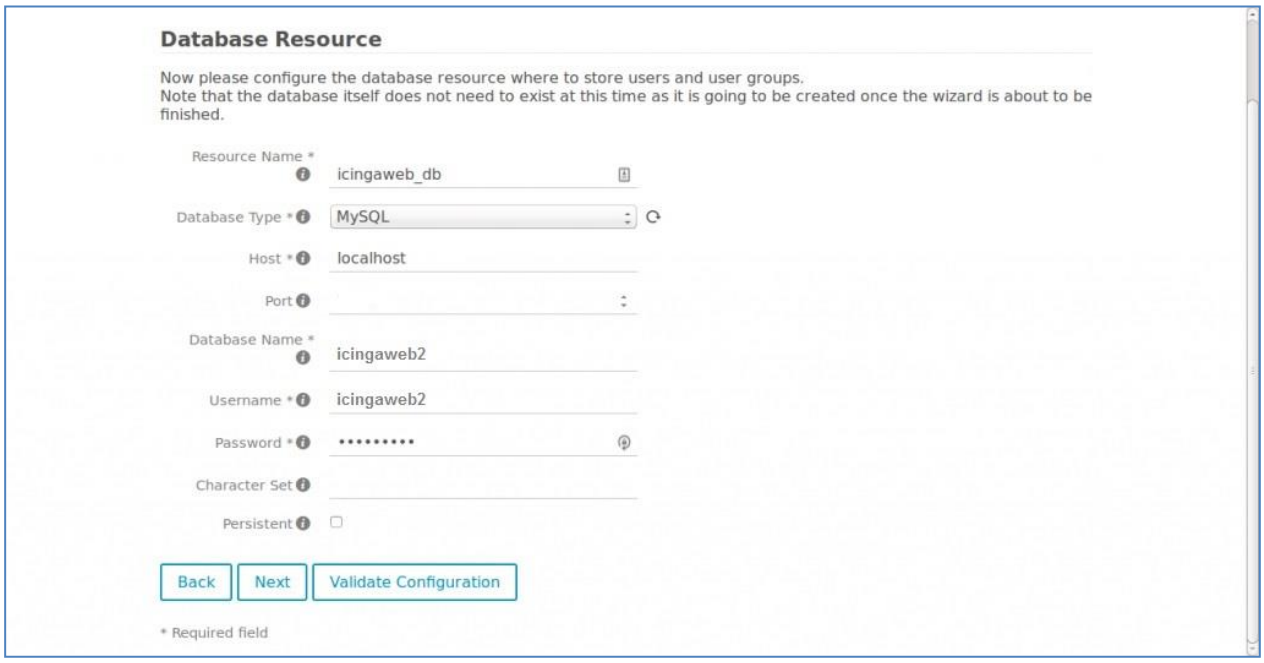


Εικόνα 4: Οι ρυθμίσεις του συστήματος μετά τις αλλαγές

Στην επόμενη οθόνη θα πρέπει να επιλέξουμε το authentication type ως database.

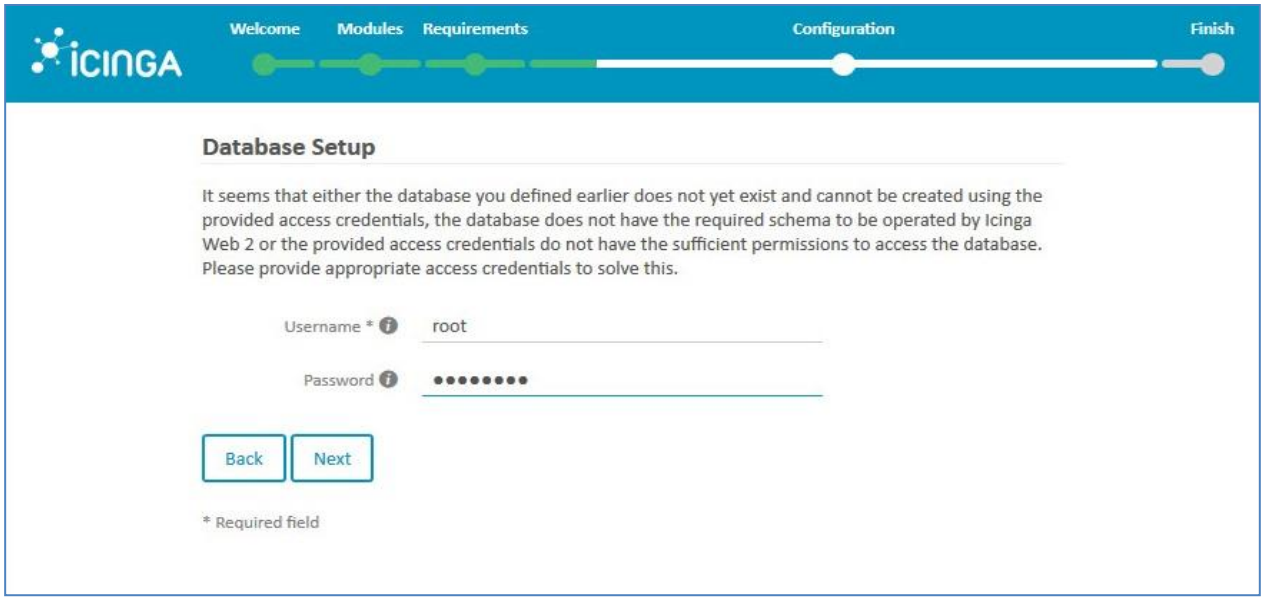
Εικόνα 5: Ορισμός του authentication type

Συνεχίζουμε στο Database Resource όπου και δημιουργούμε μια καινούργια βάση δεδομένων την οποία θα κάνουμε validate και αφού μας εμφανίσει ότι έγινε, πατάμε next.



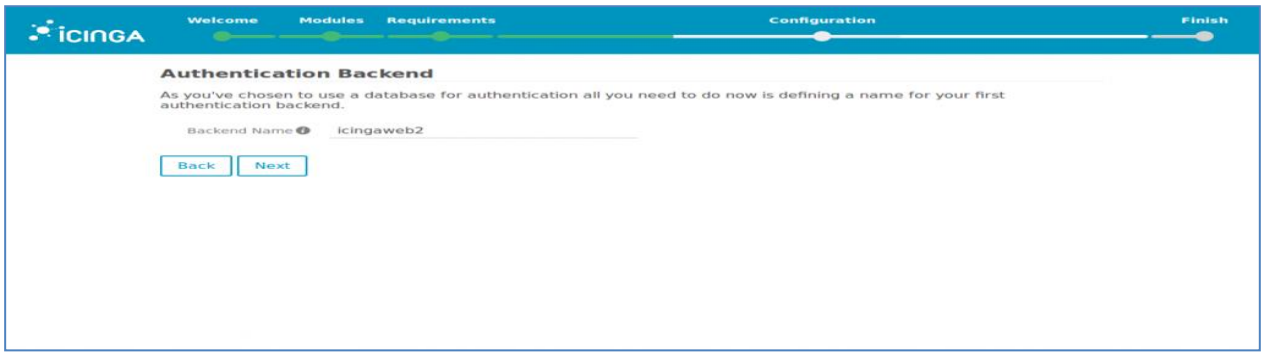
Εικόνα 6: Ορισμός του Database Resource

Η επόμενη οθόνη database setup μας καλεί να δηλώσουμε το λογαριασμό root για το mysql ώστε να δημιουργηθεί η καινούργια database που βάλαμε προηγουμένως, πληκτρολογώντας το username και password μας.



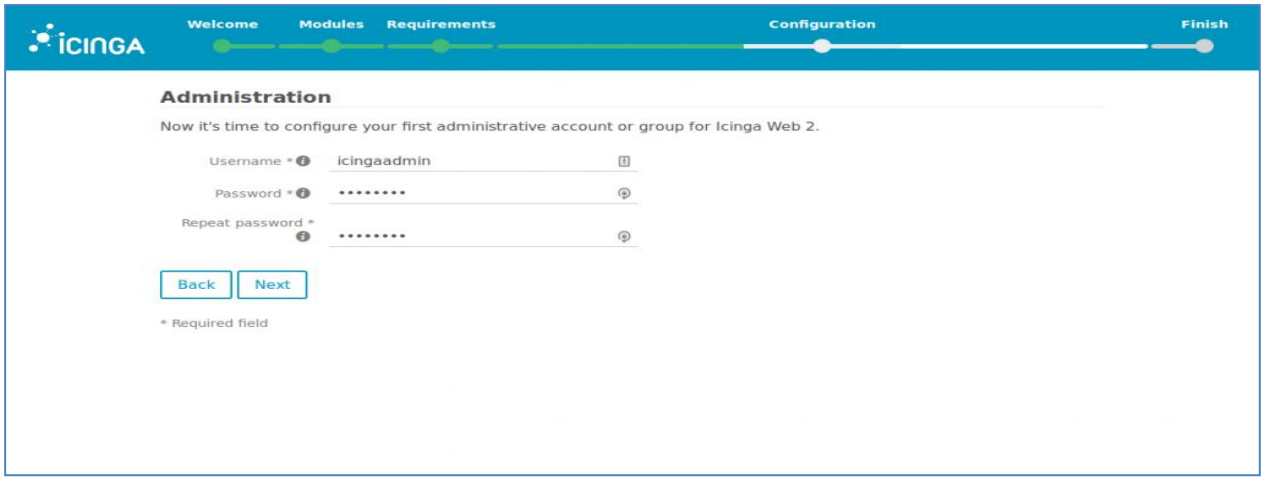
Εικόνα 7: Δημιουργία λογαριασμού root

Στη συνέχεια στην οθόνη authentication backend δηλώνουμε το όνομα της database μας.



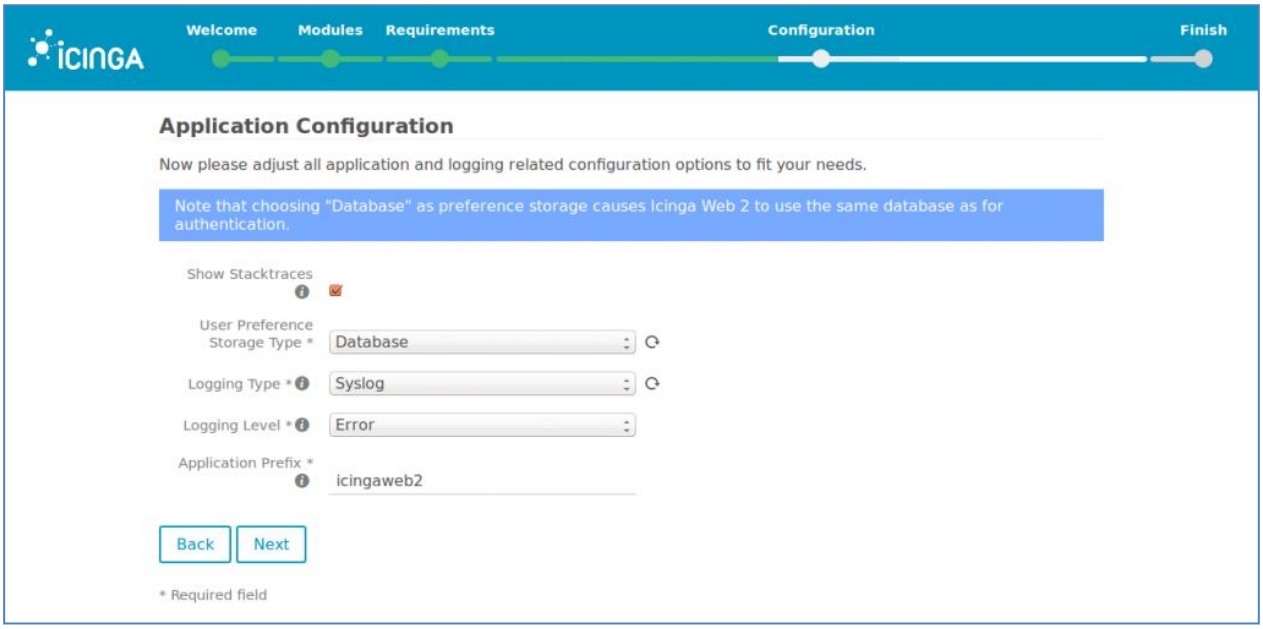
Εικόνα 8: Δήλωση ονόματος της Database

Στην επόμενη οθόνη administration, δημιουργούμε τον administrator για το icinga web 2, δηλώνοντας το username και το password που θα έχει.



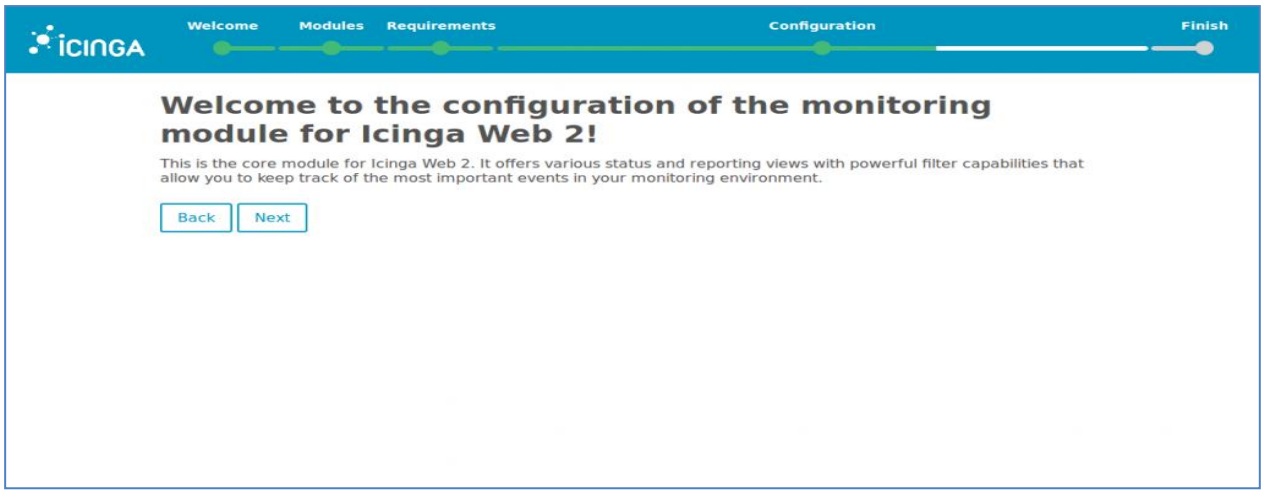
Εικόνα 9: Δημιουργία administrator

Στην οθόνη που ακολουθεί θα πρέπει να ρυθμίσουμε τις παραμέτρους της εφαρμογής όπως φαίνονται στην εικόνα.

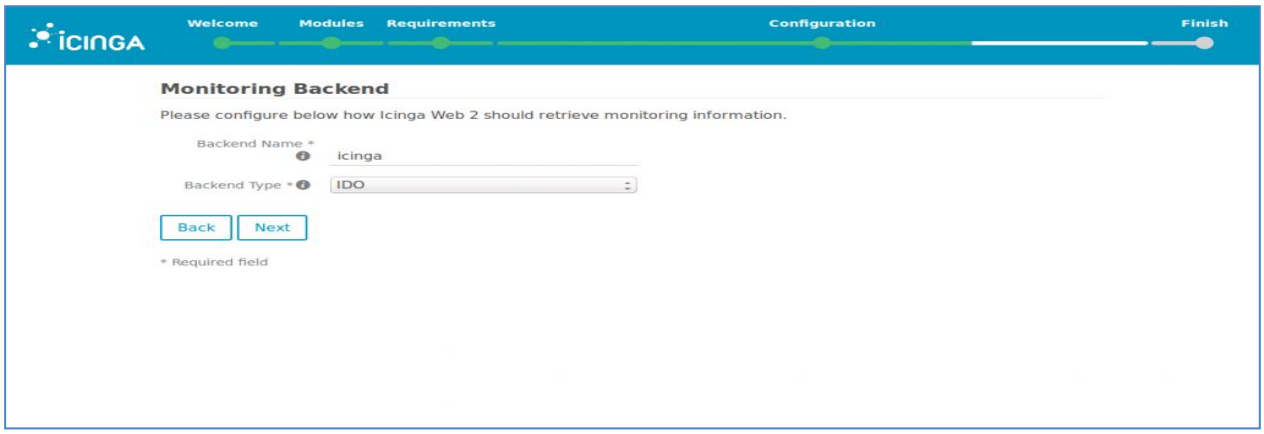


Εικόνα 10: Ρύθμιση παραμέτρων

Βεβαιωνόμαστε ότι η ρύθμιση του icinga web 2 έχει γίνει με επιτυχία και επιλέγουμε από που θα παίρνει τα δεδομένα.

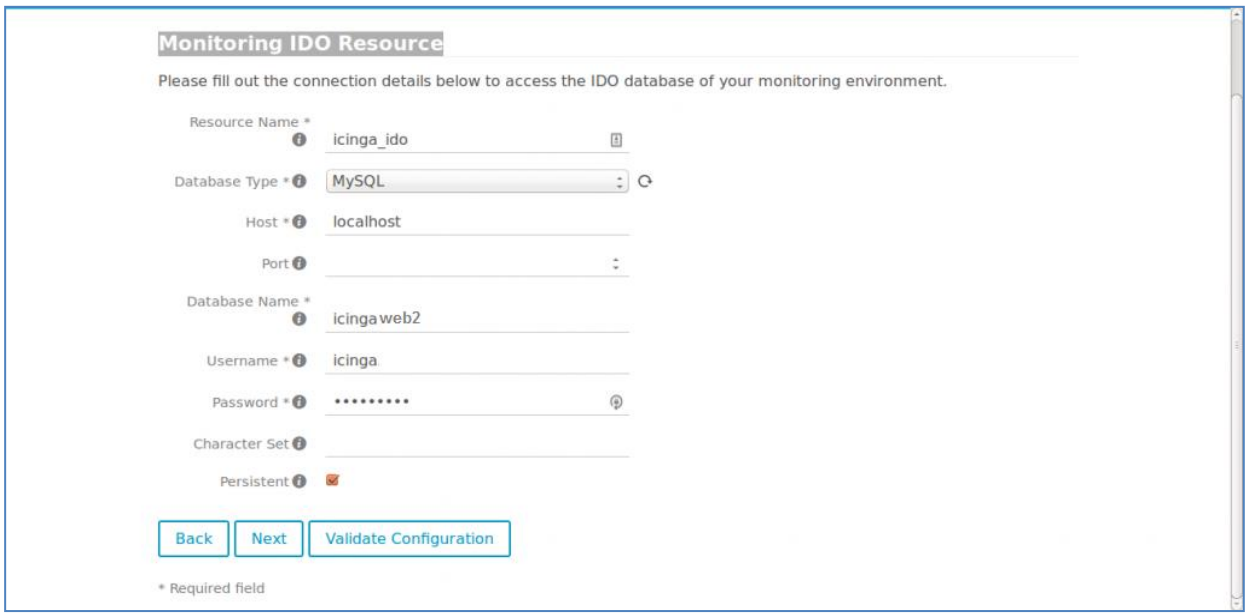


Εικόνα 11: Επιτυχής επιβεβαίωση ρυθμίσεων



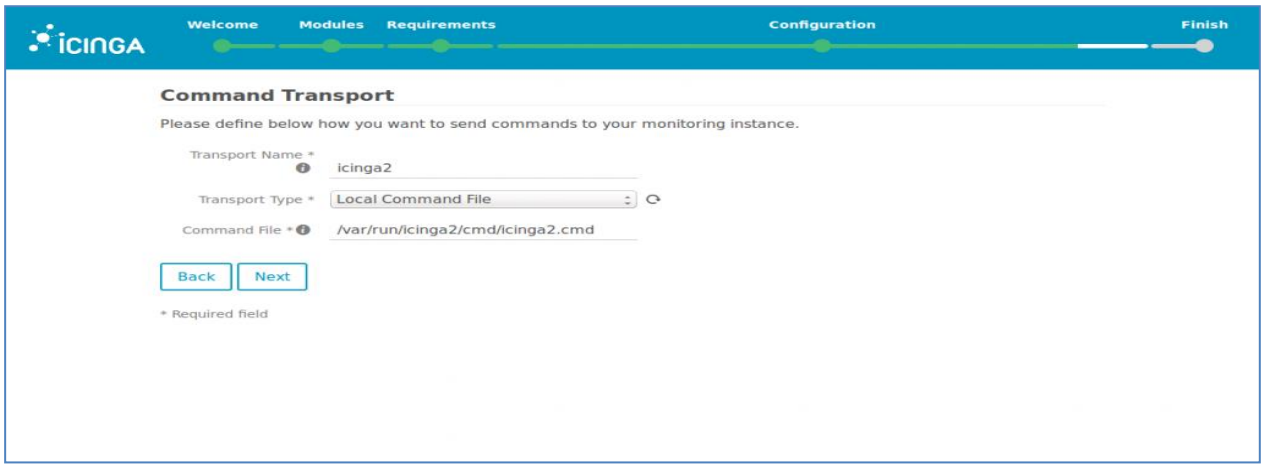
Εικόνα 12: Δήλωση επιλογής δεδομένων

Στη συνέχεια δηλώνουμε την ido mysql database που φτιάξαμε προηγουμένως και κάνουμε validate.



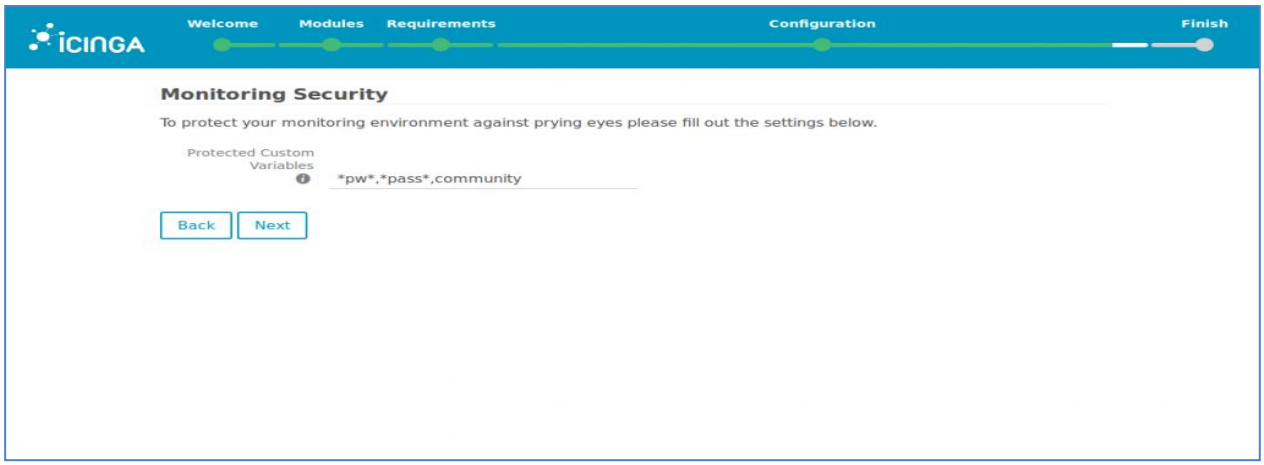
Εικόνα 13: Δήλωση της ido mysql database

Στην επόμενη οθόνη που μας εμφανίζει, αφήνουμε ίδιο το Command Transport,



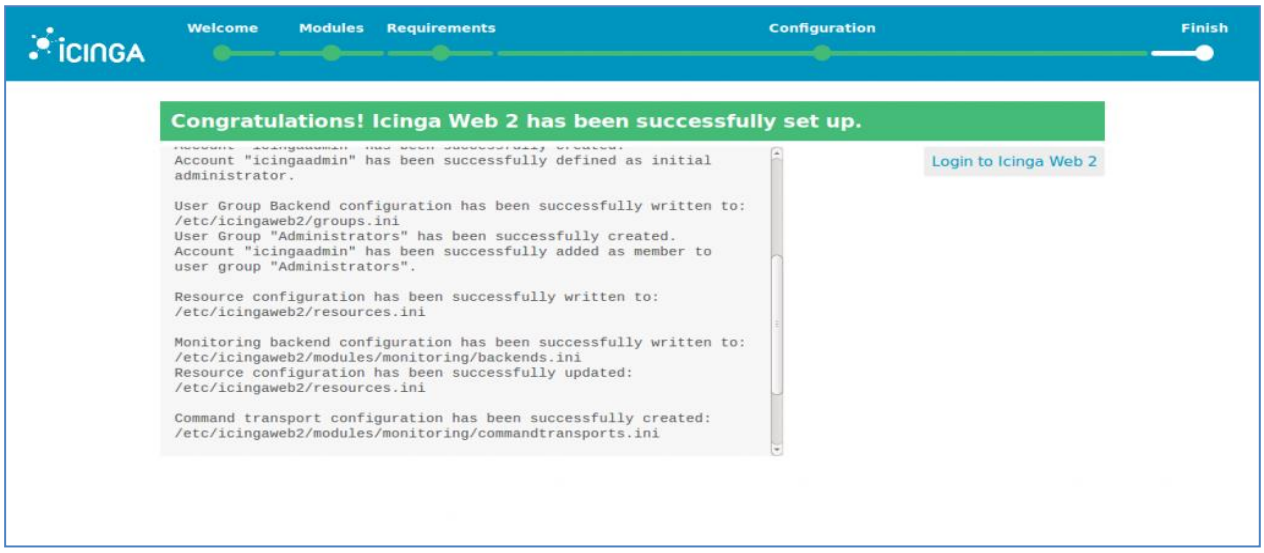
Εικόνα 14: Οθόνη Command Transport

Όπως επίσης το ίδιο κάνουμε και στην οθόνη Monitoring Security.



Εικόνα 15: Οθόνη Monitoring Security.

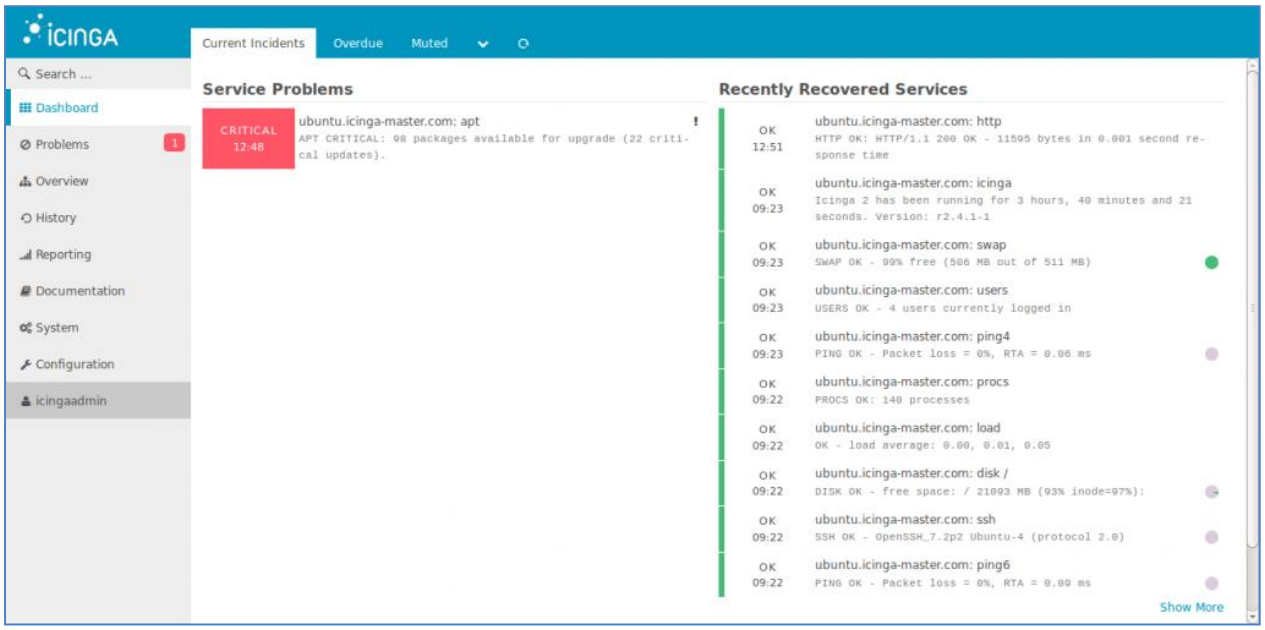
Τέλος, στην οθόνη που ακολουθεί μπορούμε να δούμε μια γενική επισκόπηση των ρυθμίσεων που έχουμε κάνει, ολοκληρώνεται η εγκατάσταση του icinga web 2 και συνδεόμαστε στην εφαρμογή.



Εικόνα 16: Γενική επισκόπηση των ρυθμίσεων και εγκατάσταση

Εικόνα 17: Αρχική οθόνη εισαγωγής στη εφαρμογή

Εδώ έχουμε την δυνατότητα να δούμε τις ειδοποιήσεις της υπηρεσίας του κύριου Icinga server μας ή μπορούμε απλά να περιηγηθούμε στο URL http: // IP / icingaweb2 / ώστε να έχουμε πρόσβαση στο web interface. Μπορούμε να προσθέσουμε οποιοδήποτε αριθμό κόμβων σε αυτό το σύστημα για την παρακολούθηση.



Εικόνα 18: Interface εφαρμογής

Κεφαλαιο 3

3.1 Τι είναι τα Check_log files

Ο όρος check_logfiles αναφέρεται σε ένα plugin για το Nagios το οποίο σαρώνει τα αρχεία καταγραφής (logfiles) προκειμένου να βρεί συγκεκριμένα πρότυπα, επίσης είναι σημαντικό να αναφερθεί ότι όλα τα Plugins για το Nagios παίζουν εξίσου το ίδιο και στο Icinga 2 και για αυτό το επιλέξαμε. Τα check_logfiles plugin έχουν σχεδιαστεί ώστε να λειτουργούν σε κρίσιμα περιβάλλοντα επομένως οι γραμμές καταγραφής δεν είναι αποδεκτό να λείπουν μετά από μια εναλλαγή του αρχείου καταγραφής. Όταν μια τέτοιου είδους εναλλαγή του αρχείου καταγραφής λαμβάνει χώρα, το check_logfiles ανιχνεύει και αναλύει τις γραμμές του αρχειοθετημένου αρχείου καταγραφής, ακόμα κι αν αυτό είναι συμπιεσμένο.

Συνήθως αυτό που γίνεται είναι να ανιχνεύει μόνο τις γραμμές ενός αρχείου καταγραφής που προστέθηκαν από την τελευταία εκτέλεση του plugin. Τα κύρια χαρακτηριστικά του είναι τα εξής:

- Μπορούν να δοθούν πολλαπλές κανονικές εκφράσεις.
- Οι εκφράσεις μπορούν να κατηγοριοποιηθούν ως προειδοποίηση (warning) ή κρίσιμο (critical) .
- Μπορεί να χειριστεί οποιαδήποτε στρατηγική εναλλαγή αρχείων καταγραφής.
- Hook scripts, είτε εξωτερικά σενάρια είτε ένα κομμάτι κώδικα perl στο αρχείο ρυθμίσεων μπορεί να εκτελεί ενέργειες όταν μια γραμμή αντιστοιχεί σε ένα πρότυπο. Για παράδειγμα, κάθε φορά που ένα κρίσιμο πρότυπο (critical pattern) έχει βρεθεί, ένα NSCA (Nagios Service Check Acceptor) μήνυμα αποστέλλεται στο διακομιστή Nagios.
- Είναι γραμμένο σε Perl, αλλά διανέμεται επίσης και ως Windows.exe **[12]**

3.1.1 Κίνητρο

Σε ένα κρίσιμο περιβάλλον τα συμβατικά plugins που σαρώνουν τα αρχεία καταγραφής (logfiles) δεν είναι επαρκή. Η έλλειψη ικανότητας στο χειρισμό της εναλλαγής των αρχείων καταγραφής και της ένταξής τους στην σάρωση επιτρέπει κενά στην παρακολούθηση. Τα check_logfiles γράφτηκαν ακριβώς επειδή οι ελλείψεις αυτές θα εμπόδιζαν το Nagios από το να αντικαταστήσει ένα ιδιόκτητο σύστημα παρακολούθησης. [13]

3.1.2 Χαρακτηριστικά

Ορισμένα κύρια χαρακτηριστικά που βοηθούν στην υλοποίηση παρουσιάζονται παρακάτω και είναι τα εξής:

- Ανίχνευση εναλλαγών - Συνήθως το βράδυ τα logfiles εναλλάσσονται και συμπιέζονται. Κάθε λειτουργικό σύστημα ή εταιρεία έχει το δικό του σύστημα ονοματοδοσίας. Εάν αυτή η εναλλαγή γίνεται μεταξύ δύο εκτελέσεων των check_logfiles τότε το εναλλασσόμενο αρχείο πρέπει να σαρωθεί για να αποφευχθούν τα κενά. Τα συνηθέστερα συστήματα εναλλαγής είναι προκαθορισμένα αλλά μπορούν να περιγράψουν οποιαδήποτε στρατηγική.
- Περισσότερα από ένα πρότυπα μπορούν να οριστούν τα οποία και πάλι μπορούν να χαρακτηριστούν ως πρότυπα προειδοποίησης και κρίσιμα πρότυπα.
- Ενέργειες - Συνήθως τα plugins του Nagios επιστρέφουν μόνο ένα κωδικό εξόδου και μια γραμμή κειμένου που περιγράφει το αποτέλεσμα του ελέγχου. Κάποιες φορές όμως χρειάζεται να εκτελέσουμε κώδικα κατά την σάρωση κάθε φορά που παίρνουμε ένα χτύπημα (hit). Τα check_logfiles μας επιτρέπουν να καλούμε τα σενάρια (scripts), είτε μετά από κάθε χτύπημα είτε στην αρχή είτε στο τέλος της εκτέλεσης.
- Εξαιρέσεις - Αν ένα πρότυπο ταιριάζει, η γραμμή αυτή θα μπορούσε να αποτελεί μια πολύ ειδική περίπτωση που δεν θα πρέπει να υπολογίζεται ως σφάλμα. Μπορούμε να ορίσουμε τα πρότυπα εξαιρέσεων τα οποία είναι πιο συγκεκριμένες εκδόσεις των κρίσιμων / προειδοποιητικών (critical / warning) προτύπων. Ένα τέτοιο ταίριασμα θα ακύρωνε μια προειδοποίηση.
- Κατώφλι - Μπορούμε να ορίσουμε τον αριθμό των γραμμών που θα ταιριάζουν οι οποίες είναι απαραίτητες για την ενεργοποίηση μιας προειδοποίησης.
- Πρωτόκολλο - Οι γραμμές που ταιριάζουν μπορούν να γραφτούν σε ένα αρχείο πρωτοκόλλου το όνομα του οποίου θα περιληφθεί στην έξοδο του plugin.
- Μακροεντολές - Οι ορισμοί των προτύπων και τα ονόματα των αρχείων καταγραφής είναι δυνατό να περιέχουν μακροεντολές, οι οποίες επιλύονται κατά το χρόνο της εκτέλεσης.
- Δεδομένα απόδοσης - Ο αριθμός των γραμμών σαρώνεται και ο αριθμός των warnings/ criticals είναι η έξοδος.
- Windows - Το plugin λειτουργεί με Unix, καθώς και με τα Windows (π.χ. με ActiveState Perl).

3.2 Εγκατάσταση

Όσον αφορά την εγκατάσταση του plugin πραγματοποιήθηκε με τον ακόλουθο τρόπο ακολουθώντας τις παρακάτω εντολές:

Για να το κατεβάσουμε συγκεκρίμενο plugin γράφουμε:

```
$ Wget http://labs.consol.de/download/shinken-nagios-plugins/check_logfiles-3.5.1.tar.gz
```

Με την παρακάτω εντολή ανοίγουμε το αρχείο

```
$ tar xzvf check_logfiles-3.5.1.tar.gz
```

και στη συνέχεια κάνουμε compile με την εντολή

```
$ make
```

Και για την εγκατάσταση γράφουμε την εντολή

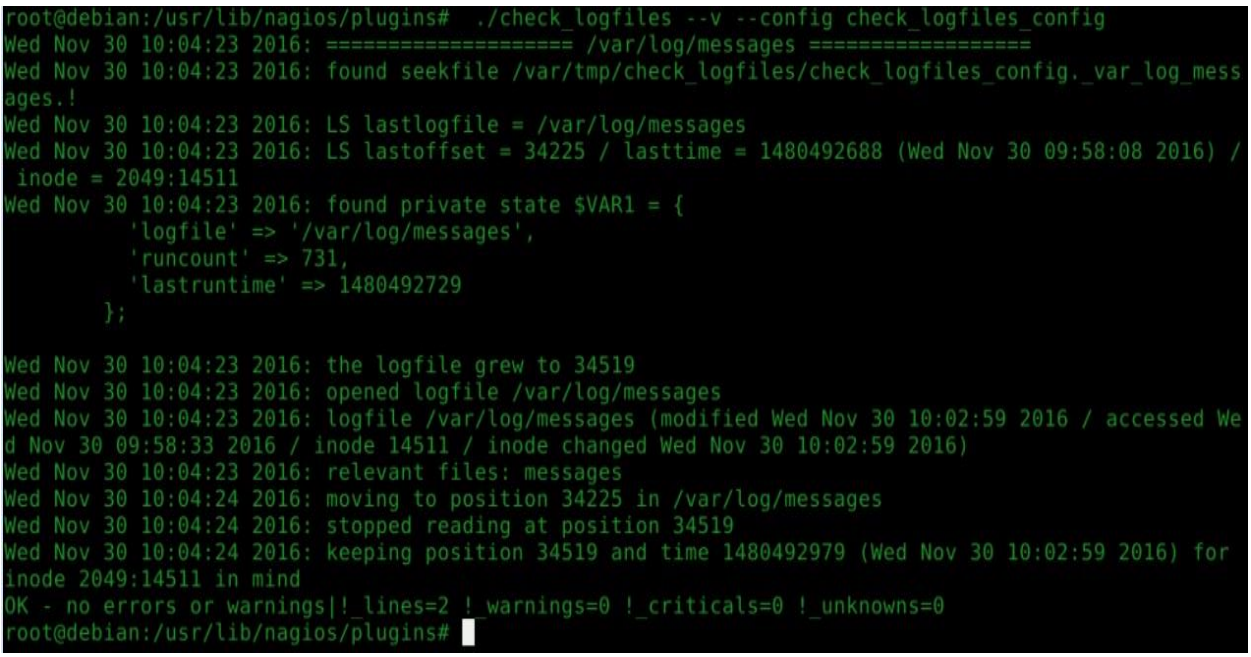
```
$ make install
```

3.2.1 Απλή λειτουργία plugin

Συνήθως, για να καλέσουμε το plugin το κάνουμε με την επιλογή --config, που παίρνει το όνομα ενός αρχείου ρυθμίσεων:

```
nagios$ check_logfiles --config
OK - no errors or warnings
```

Σε αυτή την πιο απλή του μορφή το check_logfiles μπορεί να πάρει όλες τις απαραίτητες παραμέτρους και τις επιλογές της γραμμής εντολών. Ωστόσο, δεν μπορούν όλα τα χαρακτηριστικά να χρησιμοποιηθούν σε αυτή την περίπτωση, όπως βλέπουμε στην εικόνα 19 παρακάτω.



Εικονα 19. Αποτέλεσμα της εκτέλεσης του check_log_file

Αρχικά τα check_logfiles σαρώνουν ένα αρχείο καταγραφής μέχρι να φτάσει το αρχείο στο τέλος του. Στη συνέχεια το αποτέλεσμα που θα εκδοθεί αποθηκεύεται στο λεγόμενο ζητούμενο (seekfile) αρχείο. Την επόμενη φορά που το check_logfile θα τρέξει, το αποτέλεσμα αυτό θα χρησιμοποιηθεί ως η αρχική θέση μέσα στο αρχείο καταγραφής. Εν τω μεταξύ σε περίπτωση που έχει συμβεί μια εναλλαγή, το υπόλοιπο εναλλασσόμενο αρχείο θα σαρωθεί επίσης. Για τις πιο απλές εφαρμογές είναι αρκετό να καλέσουμε το check_logfile με τις παραμέτρους της γραμμής εντολών. Οι πιο σύνθετες εργασίες σάρωσης μπορούν να περιγραφούν με ένα αρχείο config.

3.2.2 Παράμετροι της γραμμής εντολών

Όσον αφορά τη δημιουργία του plugin χρησιμοποιούνται ορισμένες παράμετροι στη γραμμή εντολών οι οποίες μας δίνουν την δυνατότητα να ορίσουμε ακριβώς τι θέλουμε να ελέγξουμε κατά την αναζήτηση, με ποιο τρόπο να γίνεται αυτό και ποια μορφή θα έχει το αποτέλεσμα που θα μας επιστρέψει κατά την εκτέλεση του το plugin. Στη συνέχεια παρουσιάζονται όλες οι διαθέσιμες παράμετροι με την επεξήγηση τους, ορισμένες εκ των οποίων χρησιμοποιήθηκαν στην δημιουργία του δικού μας plugin στα πλαίσια της παρούσας διπλωματικής εργασίας και εμφανίζονται με πράσινο χρώμα:

- **--tag=<identifier>** Θα εμφανιστεί στην έξοδο του plugin και χρησιμοποιείται για το διαχωρισμό των διαφόρων υπηρεσιών.

- --logfile=**<filename>** Αφορά το όνομα του αρχείου καταγραφής που θέλουμε να σαρώσουμε.
- **--rotation=<method>** Είναι η μέθοδος για το πως θα εναλλάσσονται τα αρχεία.
- --criticalpattern=**<regexp>** Έκφραση η οποία οδηγεί σε ένα κρίσιμο σφάλμα.
- --warningpattern=**<regexp>** Έκφραση η οποία οδηγεί σε μια προειδοποίηση.
- **--criticalexception=<regexp>/--warningexception=<regexp>** Αποτελούν εξαιρέσεις που δεν υπολογίζονται ως σφάλματα.
- --okpattern=**<regexp>** Πρότυπο το οποίο μηδενίζει τους μετρητές λάθους.
- --noprotocol Όλες γραμμές που ταιριάζουν είναι γραμμένες σε ένα αρχείο πρωτοκόλλου και με το όνομα αυτού του αρχείου θα εμφανίζεται στην έξοδο του plugin. Αυτή η επιλογή το απενεργοποιεί.
- **--syslogserver** Με αυτή την επιλογή περιορίζουμε το πρότυπο που ταιριάζει στις γραμμές που προέρχονται από τα check_logfiles που εκτελούνται στον server.
- **--syslogclient=<clientname>** Με αυτή την επιλογή περιορίζουμε το πρότυπο που ταιριάζει στις γραμμές που προέρχονται από τον host .
- --sticky[=<**lifetime>]** Σφάλματα τα οποία μεταδίδονται μέσω διαδοχικών εκτελέσεων.
- **--unstick** Επαναφέρει τα σφάλματα που μεταδίδονται μέσω διαδοχικών εκτελέσεων.
- --config Το όνομα ενός αρχείου ρυθμίσεων.
- --configdir Το όνομα ενός καταλόγου ρυθμίσεων. Τα config αρχεία που τελειώνουν σε .cfg ή .conf εισάγονται αναδρομικά.
- **--searches=<tag1,tag2,…>** Λίστα ετικετών των αναζητήσεων αυτών που πρόκειται να εκτελεστούν. Χρησιμοποιώντας την παράμετρο αυτή, δεν εκτελούνται όλες οι αναζητήσεις που αναφέρονται στο config αρχείο, αλλά μόνο εκείνες που έχουν επιλεχθεί.
- **--report=[short|long|html]** Αυτή η επιλογή ενεργοποιεί πολλές γραμμές εξόδου (Προεπιλογή: απενεργοποιημένο). Η ρύθμιση html δημιουργεί ένα πίνακα που εμφανίζει τα τελευταία χτυπήματα στην υπηρεσία προβολής στοιχείων.
- **--maxlength=[length]** Με την παράμετρο αυτή οι μεγάλες ουρές περικόπτονται (Προεπιλογή: απενεργοποιημένο). Ορισμένα προγράμματα (π.χ. TrueScan) δημιουργούν εγγραφές στο αρχείο καταγραφής συμβάντων τέτοιου μήκους, ώστε η έξοδος του plugin να είναι παραπάνω από 1024 χαρακτήρες. Το NSClient ++ τα απορρίπτει αυτά.

- **--winwarncrit** Με την παράμετρο αυτή τα μηνύματα στο αρχείο καταγραφής συμβάντων κατηγοριοποιούνται από τον τύπο WARNING/ERROR (Προεπιλογή: απενεργοποιημένο).
- **--rununique** Αυτή η παράμετρος εμποδίζει τα check_logfiles να ξεκινήσουν όταν ήδη ένα άλλο περιστατικό χρησιμοποιεί το ίδιο config file.
- **--timeout=<seconds>** Αυτή η παράμετρος προκαλεί διακοπή της τρέχουσας αναζήτησης μετά από ένα καθορισμένο αριθμό δευτερολέπτων. Η διακοπή γίνεται με ελεγχόμενο τρόπο, έτσι ώστε οι γραμμές που έχουν διαβαστεί μέχρι εκείνη την στιγμή να χρησιμοποιηθούν στον υπολογισμό του τελικού αποτελέσματος.
- **--warning=<Number>** Με τον τρόπο αυτό πολύπλοκα σενάρια χειρισμού, εφοδιάζονται με μια προειδοποιητική παράμετρο (φυσικά critical). Μέσα στο σενάριο η τιμή είναι προσπελάσιμη ως macro CL_WARNING.

3.2.3 Η μορφή ενός αρχείου διαμόρφωσης

Οι ορισμοί είναι γραμμένοι με Perl-σύνταξη. Υπάρχει μια διάκριση μεταξύ των καθολικών μεταβλητών που επηρεάζει τα check_logfiles στο σύνολό τους και τις μεταβλητές που σχετίζονται με τις ενιαίες αναζητήσεις. Μια "αναζήτηση" συνδυάζει το πού να ψάξουμε, τι να ψάξουμε, τι βάρος έχει ένα χτύπημα, ποια ενέργεια θα ενεργοποιείται στην περίπτωση ενός χτυπήματος, και ούτω καθεξής.

$seekfilesdir	Είναι ένας κατάλογος όπου τα αρχεία με τις πληροφορίες για την κατάσταση του θα αποθηκευτούν μετά από έναν έλεγχο των check_logfiles. Οι πληροφορίες κατάστασης βοηθούν τα check_logfiles να θυμούνται μέχρι ποια θέση έχει σαρωθεί το αρχείο καταγραφής κατά τη διάρκεια της τελευταίας εκτέλεσης. Με αυτό τον τρόπο θα διαβαστούν μόνο οι πρόσφατες γραμμές των αρχείων καταγραφής .	Η προεπιλογή είναι το / tmp ή ο κατάλογος που έχει καθοριστεί με την -με -seekfiles -dir του ./configure.
$protocolsdir	Είναι ένας κατάλογος όπου τα check_logfiles καταγράφουν τα αρχεία πρωτοκόλλου με τις	Η προεπιλογή είναι το / tmp ή ο κατάλογος

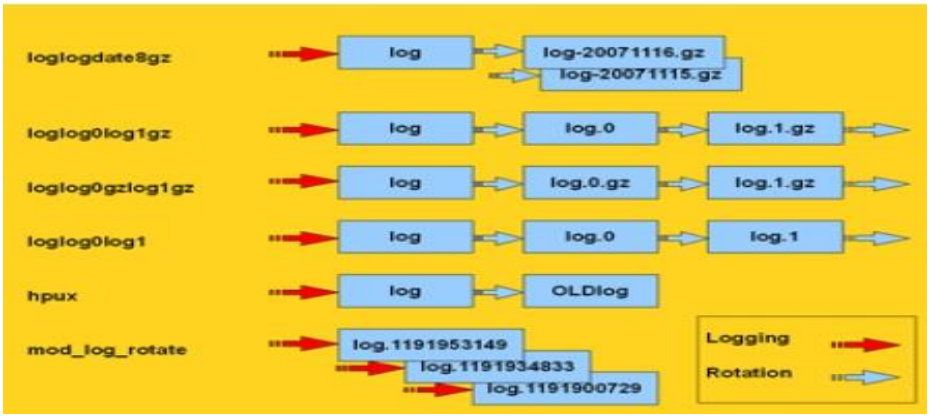
	γραμμές που ταιριάζουν.	που έχει καθοριστεί με την -με -protocol -dir of ./configure.
$protocolretention	Εμφανίζει την διάρκεια ζωής των αρχείων πρωτοκόλλου σε ημέρες. Μετά από αυτές τις ημέρες τα αρχεία διαγράφονται αυτόματα.	Η προεπιλογή είναι 7 ημέρες.
$scriptpath	Λίστα καταλόγων, όπου μπορούν να βρεθούν τα εκτελέσιμα σενάρια (χωρισμένα από: κάτω από το Unix και κάτω απο τα Windows).	Η προεπιλογή είναι / bin: / usr / bin: / sbin: / usr / sbin ή οι κατάλογοι που έχει καθοριστεί με την -με -trusted -path of ./configure.
$MACROS	Η hash με τους macro ορισμούς, ορίζεται από τον χρήστη.	
$prescript	Εξωτερικό σενάριο το οποίο θα εκτελεστεί κατά την εκκίνηση του check_logfiles. Η μακροεντολή $ CL_TAG παίρνει την τιμή "εκκίνηση". Οι παράμετροι $ prescriptparams, $ prescriptstdin και $ prescriptdelay μπορούν να χρησιμοποιηθούν σαν scriptparams, scriptstdin και scriptdelay.	
$postscript	Εξωτερικό σενάριο το οποίο θα εκτελεστεί πριν απο την λήξη του check_logfiles. Η μακροεντολή $ CL_TAG παίρνει την τιμή "περίληψη". Οι παράμετροι $ prescriptparams, $ prescriptstdin και $ prescriptdelay μπορούν να χρησιμοποιηθούν σαν scriptparams, scriptstdin και scriptdelay.	

$options	Λίστα με τις επιλογές που ελέγχουν την επιρροή των προ και μετά σεναρίων. Οι επιλογές είναι οι smartpostscript, supersmartpostscript, smartprescript και supersmartprescript. Με την option report= ” short |long|html” μπορούμε να προσαρμόσουμε την έξοδο του plugin. Με την report=”long|html” η έξοδος του plugin μπορεί ενδεχομένως να είναι πολύ μεγάλη. Από προεπιλογή είναι ορισμένο σε 4096 χαρακτήρες (η ποσότητα των δεδομένων που ένα unpatched Nagios είναι σε θέση να επεξεργαστεί). Η maxlength επιλογή μπορεί να χρησιμοποιηθεί για να αυξήσει αυτό το όριο, π.χ. maxlength = 8192. Η επιλογή seekfileerror καθορίζει το errorlevel, αν ένα seekfile δεν μπορεί να γραφτεί, π.χ. seekfileerror = άγνωστο (προεπιλογή: critical). Το ίδιο ισχύει και για protocolfileerror (προεπιλογή: ok). Συνήθως το τελευταίο μήνυμα σφάλματος θα εμφανιστεί στην πρώτη γραμμή της εξόδου. Με προεπισκόπηση = 5 μπορούμε να πούμε στο check_logfiles να μας εμφανίσει για παράδειγμα τα τελευταία 5 χτυπήματα. (Προεπιλογή είναι: προεπισκόπηση = 1).	
@searches	Τα στοιχεία (αναφορές hash) περιγράφουν την πραγματική εργασία των check_logfiles. Τα κλειδιά για αυτές τις αναφορές hash βρίσκονται στον επόμενο πίνακα.	

Πίνακας 2. Μεταβλητές αναζήτησης

3.2.4 Μονές αναζητήσεις

Οι μονές αναζητήσεις διευκρινίζονται περαιτέρω από τις ακόλουθες παραμέτρους:

Tag	Μοναδικό αναγνωριστικό.
logfile	Το όνομα του αρχείου καταγραφής, για να σαρωθεί.
archivedir	Το όνομα του καταλόγου όπου τα αρχεία θα μετακινηθούν μετά από μια εναλλαγή του αρχείου καταγραφής. Η προεπιλογή είναι ο κατάλογος στον οποίο βρίσκεται το αρχείο καταγραφής. 
rotation	Μία από τις προκαθορισμένες μεθόδους ή κανονική έκφραση, η οποία βοηθά στην αναγνώριση των εναλλασσόμενων αρχείων. Εάν αυτό λείπει, τα check_logfiles υποθέτουν ότι το αρχείο καταγραφής απλά θα αντικατασταθεί αντί να εναλλαχθεί.
Type	Ένα από τα "rotating", "simple", "virtual", "errpt", "ipmitool", "oraclealertlog" ή "eventlog" αν τα παράθυρα eventlog πρέπει να σαρωθούν.
criticalpatterns	Είναι μια έκφραση ή μια αναφορά σε μια σειρά από αυτές τις εκφράσεις. Εάν μία από αυτές τις εκφράσεις ταιριάζει με μια γραμμή στο αρχείο καταγραφής, αυτό θεωρείται ως ένα κρίσιμο σφάλμα. Αν η έκφραση ξεκινάει με "!", τότε το νόημα αντιστρέφεται. Αν δεν βρεθεί αντιστοιχία για αυτό το πρότυπο, μετράει ως ένα κρίσιμο σφάλμα.
criticalexceptions	Μία ή περισσότερες κανονικές εκφράσεις που ακυρώνουν προηγούμενο ταίριασμα του criticalpatterns.

warningpatterns	Αντιστοιχεί στα criticalpatterns, εκτός από το warning αντ' αυτού δημιουργείται ένα κρίσιμο σφάλμα.
warningexceptions	βλέπε παραπάνω
okpatterns	Είναι μια έκφραση ή μια αναφορά σε μια σειρά από τέτοιες εκφράσεις. Εάν μία από αυτές τις εκφράσεις ταιριάζει με μια γραμμή στο αρχείο καταγραφής, όλες τα προηγούμενα warnings και criticals απορρίπτονται.
Script	Αν ένα πρότυπο ταιριάζει, τότε αυτό το σενάριο θα εκτελεστεί. Θα πρέπει να βρίσκεται κάτω απο έναν από τους καταλόγους που ορίζονται στο $ scriptpath. Το σενάριο δέχεται αφθονία πληροφοριών σχετικά με τα χτύπημα μέσω των μεταβλητών περιβάλλοντος.
scriptparams	Δίνουμε τις παραμέτρους της γραμμής εντολών για το σενάριο και μπορούν να περιέχουν μακροεντολές. Αν το $script είναι μια αναφορά σε κώδικα, τότε το $scriptparams πρέπει να είναι ένας δείκτης μιας σειράς.
scriptstdin	Περιγράφεται εάν το σενάριο αναμένει είσοδο μέσω του stdin. Η συμβολοσειρά μπορεί επίσης να περιέχει μακροεντολές.
scriptdelay	Εφόσον το σενάριο έχει ολοκληρωθεί, τα check_logfiles περιμένουν για κάποια δευτερόλεπτα <delay> πριν συνεχίσουν την εκτέλεσή τους.
options	Είναι μια λίστα διαχωρισμένη με κόμματα με επιλογές όπου επιτρέπουν στο να τελειοποιηθεί η αναζήτηση. Κάθε επιλογή μπορεί να απενεργοποιηθεί εφόσον προηγείται στο όνομά της η λέξη 'no'. Οι επιλογές εξηγούνται λεπτομερώς στον επόμενο πίνακα.
template	Αντί για μια ετικέτα, η αναζήτηση μπορεί επίσης να αναγνωρίζεται από ένα όνομα προτύπου. Αν καλέσουμε την check_logfiles με το -tag option, τότε η αναζήτηση θα γίνει σαν να είχε οριστεί απο πριν με το όνομα της ετικέτας.

Πίνακας 3: Μεταβλητές μονών αναζητήσεων

3.2.5 Επιπλέον επιλογές

Στη συνέχεια παρουσιάζονται ορισμένες επιπλέον επιλογές που διατίθενται για χρήση κατά την αναζήτηση:

[no]script	Ελέγχει εάν ένα σενάριο μπορεί να εκτελεστεί.	προεπιλογή: off
[no]smartscript	Ελέγχει εάν ο κωδικός εξόδου και η έξοδος του script θα πρέπει να αντιμετωπιστεί ως ένα επιπλέον ταίριασμα.	προεπιλογή: off
[no]supersmartscript	Ελέγχει εάν ο κωδικός εξόδου και η έξοδος του script θα πρέπει να αντικαταστήσουν το ταίριασμα.	προεπιλογή: off
[no]protocol	Ελέγχει εάν οι γραμμές που ταιριάζουν είναι γραμμένες σε κάποιο αρχείο πρωτοκόλλου για μετέπειτα διερεύνηση.	προεπιλογή: on
[no]count	Ελέγχει εάν τα χτυπήματα προσμετρούνται και αποφασίζονται πέρα απο τον τελικό κωδικό εξόδου.	προεπιλογή: on
[no]syslogserver	Εάν οριστεί, μόνο οι γραμμές που βρίσκονται τοπικά λαμβάνονται υπόψη. Είναι σημαντικό αν τα check_logfiles τρέχουν σε έναν διακομιστή αρχείων καταγραφής συστήματος όπου πολλοί άλλοι hosts αναφέρουν κάποιο συμβάν επίσης.	προεπιλογή: off
[no]syslogclient=string	Είναι ένα είδους φίλτρο με το οποίο μόνο οι γράμμες που ταιριάζουν με το string θα εξεταστούν περαιτέρω.	προεπιλογή: off
[no]perfdata	Ελέγχει αν τα δεδομένα απόδοσης θα πρέπει να προστεθούν στην έξοδο.	προεπιλογή: on
[no]logfilenocry	Ελέγχει τον τρόπο αντίδρασης, εάν το αρχείο καταγραφής δεν υπάρχει. Εξ ορισμού αυτό αποτελεί	προεπιλογή: on

	ένα λόγο για άγνωστο σφάλμα. Αν έχει οριστεί η nologfilenocry, το αρχείο καταγραφής που λείπει θα συναινέσει.	
logfilemissing	Χρησιμοποιείται για να αλλάξει η UNKNOWN κατάσταση σε μια διαφορετική κατάσταση. Με το logfilemissing = critical έχουμε το check_file_existence -functionality.	προεπιλογή: unknown
[no]case	Ελέγχει αν οι κανονικές εκφράσεις είναι case-sensitive.	προεπιλογή: on
[no]sticky[=seconds]	Ελέγχει εάν ένα σφάλμα διαδίδεται μέσω διαδοχικών εκτελέσεων του check_logfiles. Μόλις βρεθεί ένα λάθος, το exitcode θα είναι μη μηδενικό μέχρι ένα okpattern να το επαναφέρει ή μέχρι να λήξει το σφάλμα μετά απο κάποια <δευτερόλεπτα>. Καλό είναι να μην χρησιμοποιούμε αυτή την επιλογή, μέχρι να είμαστε σίγουροι πως ξέρουμε τι ακριβώς κάνει.	προεπιλογή: off
[no]savethresholdcount	Ελέγχει εάν αποθηκεύεται ο μετρητής χτυπημάτων μεταξύ των εκτελέσεων. Αν ναι, ο αριθμός αυτός προστίθεται μέχρι να επιτευχθεί ένα όριο (criticalthreshold). Διαφορετικά, η εκτέλεση αρχίζει με resetted τους μετρητές.	προεπιλογή: on
[no]encoding=string	Το αρχείο καταγραφής είναι κωδικοποιημένο σε Unicode. (Π.χ. UCS-2)	προεπιλογή: off
[no]maxlength=number	Περικόπτει πολύ μεγάλες γραμμές στο <number> - Για χαρακτήρες	προεπιλογή: off
[no]winwarncrit	Χρησιμοποιείται αντί των προτύπων για να βρεί όλα τα events του τύπου WARNING / ERROR στο eventlog των Windows.	προεπιλογή: off

[no]criticalthreshold= number	Ο αριθμός που δηλώνει πόσες γραμμές πρέπει να ταιριάζουν με το πρότυπο μέχρι να θεωρηθεί ως κρίσιμο σφάλμα.	προεπιλογή: off
[no]warningthreshold= number	Ο αριθμός που δηλώνει πόσες γραμμές πρέπει να ταιριάζουν με το πρότυπο μέχρι να θεωρηθεί ως προειδοποίηση.	προεπιλογή: off
[no]allyoucaneat	Με αυτή την επιλογή το check_logfiles σαρώνει ολόκληρο το αρχείο καταγραφής κατά την αρχική του λειτουργία (όταν δεν υπάρχει seekfile).	προεπιλογή: off
[no]eventlogformat	Αυτή η επιλογή επιτρέπει να ξαναγραφτεί το κείμενο του μηνύματος ενός Windows event. Η σειρά αυτή μπορεί να εμπλουτιστεί με πρόσθετες πληροφορίες (EventID, Πηγή, ...)	προεπιλογή: off
[no]preferredlevel	Τα warningpattern/criticalpattern επιλέχθηκαν με τέτοιο τρόπο ώστε μια συγκεκριμένη γραμμή να ταιριάζει και με τα δυο (έτσι ώστε η έξοδος να μοιάζει με "1 error, 1 warning"), μπορούμε να χρησιμοποιήσουμε αυτήν την επιλογή ώστε να μετρήσουμε μόνο ένα από αυτά. (π.χ. με preferredlevel = critical η έξοδος θα είναι "1 error").	προεπιλογή: off
[no]randominode	Χρησιμοποιείται για πολύ ειδική περίπτωση, όπου η inode του αρχείου καταγραφής αλλάζει συνεχώς.	προεπιλογή: off
[no]savestate	Αυτή η επιλογή αναγκάζει τη δημιουργία ενός seekfile για αναζητήσεις εικονικού τύπου.	προεπιλογή: off
[no]capturegroups	Εάν ένα πρότυπο περιλαμβάνει παρενθέσεις για ομαδοποίηση, οι μεταβλητές $ 1, $ 2, ... αποθηκεύονται στις μακροεντολές CL_CAPTURE_ GROUP1, CL_CAPTURE_GROUP2, ... Ο αριθμός αυτών των μακροεντολών (η υψηλότερη τιμή της	προεπιλογή: off

	CL_CAPTURE_GROUPx) μπορεί να βρεθεί στο CL_CAPTURE_GROUPS.	

Πίνακας 4: Επιπλέον μεταβλητές αναζητήσεων

3.2.6 Προκαθορισμένες μακροεντολές

Στον πίνακα που ακολουθεί παρουσιάζονται οι προκαθορισμένες μακροεντολές που αφορούν τον έλεγχο οι οποίες μπορούν να χρησιμοποιηθούν κατά την αναζήτηση, καθώς και η επεξήγηση τους:

$CL_USERNAME$	Το όνομα του χρήστη που εκτελεί το check_logfiles
$CL_HOSTNAME$	Το hostname χωρίς domain
CL_DOMAIN	Το DNS-domain
CL_FQDN	Και τα δύο μαζί
$CL_IPADDRESS$	Η IP-adress
CL_DATE_YYYY	Το τρέχον έτος
CL_DATE_MM	Ο τρέχον μήνας (1..12)
CL_DATE_DD	Η ημέρα του μήνα
CL_DATE_HH	Η τρέχουσα ώρα (0..23)
CL_DATE_MI	Το τρέχον λεπτό
CL_DATE_SS	Το τρέχον δευτερόλεπτο
CL_DATE_CW	Η τρέχουσα εβδομάδα ημερολόγιο (ISO 8601: 1988)
$CL_SERVICEDESC$	Το όνομα του αρχείου config χωρίς επέκταση.
$CL_NSCA_SERVICEDESC$	Το ίδιο

$CL_NSCA_HOST_ADDRESS$	Η τοπική διεύθυνση 127.0.0.1
CL_NSCA_PORT	5667
$CL_NSCA_TO_SEC$	10
$CL_NSCA_CONFIG_FILE$	send_nsca.cfg
	Οι παρακάτω μακροεντολές αλλάζουν την αξία τους κατά το χρόνο εκτέλεσης.
CL_TAG	Η ετικέτα της τρέχουσας αναζήτησης ($ CL_tag $ είναι η ετικέτα σε μικρά γράμματα)
$CL_TEMPLATE$	Το όνομα του προτύπου που χρησιμοποιείται (εάν υπάρχει).
$CL_LOGFILE$	Το επόμενο αρχείο που θα σαρωθεί
$CL_SERVICEOUTPUT$	Η τελευταία γραμμή που ταιριάζει.
$CL_SERVICESTATEID$	Το επίπεδο σφάλματος ως αριθμός 0..3
$CL_SERVICESTATE$	Το επίπεδο σφάλματος ως λέξη (OK, WARNING, CRITICAL, UNKNOWN)
$CL_SERVICEPERFDATA$	Τα δεδομένα απόδοσης
$CL_PROTOCOLFILE$	Το αρχείο όπου όλες οι γραμμές που ταιριάζουν έχουν γραφτεί.

Πίνακας 5. Προκαθορισμένες μακροεντολές αναζήτησης

Αυτές οι μακροεντολές είναι επίσης διαθέσιμες σε σενάρια τα οποία καλούνται έξω από τα check_logfiles. Οι τιμές τους αποθηκεύονται σε μεταβλητές περιβάλλοντος, τα ονόματα των οποίων προέρχονται από τα ονόματα της μακροεντολής. Το προηγούμενο CL_ αντικαθίσταται από το check_logfiles_config. Μπορούμε επίσης να αποκτήσουμε πρόσβαση η οποία ορίζεται από το χρήστη με τις μακροεντολές.

3.2.7 Παραμετροποίηση

Στα πλαίσια της διπλωματικής εργασίας είναι η δημιουργία ένός plugin το οποίο περιλαμβάνει όσα αναφέρθηκαν και για την εκτέλεσή του εφαρμόστηκαν τα ακόλουθα βήματα:

Βημα 1ο: Αρχικά καθορίζουμε τον κατάλογο μέσα στον οποίο θέλουμε να δημιουργήσουμε και να εγκαταστήσουμε το plugin μας το οποίο ονομάσαμε check_logfile_config και είναι το αρχείο όπου θα κάνει τον έλεγχο που του έχουμε καθορίσει να γίνεται κατά την εκτέλεση του και το δημιουργούμε στον φάκελο /usr/lib/nagios/plugins/. Στην εικόνα 20 που ακολουθεί μπορούμε να δούμε το μονοπάτι στο οποίο είναι αποθηκευμένο το αρχείο μας καθώς επίσης τα ορίσματα και τις παραμέτρους που επιλέξαμε να χρησιμοποιήσουμε, τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

```
$scriptpath = '/usr/lib/nagios/plugins';
$MACROS = {
  MY_TAG => 'vassilis',
  MY_PATTERN => 'GET /icingaweb2/'
};

@searches = (
 {
  tag => '$MY_TAG$',
  logfile => '/var/log/apache2/access.log',
  sticky => '10',
  savethresholdcount => 'on',
  criticalthreshold => '2',
  allyoucaneat => 'on',
  report => 'long',
  criticalpatterns => [
    '$MY_PATTERN$',
    'Loop OFFLINE',
    'fctl:.*disappeared from fabric',
    '.*Lun.*disappeared.*'
  ],
  script => 'vassilis.sh',
  scriptparams => '$MY_PATTERN$',
  options => 'noprotocol,script,perfdata'
 },
);
```

Εικόνα 20. Το plugin Check_logfiles_config

Βημα 2ο: Στη συνέχεια του δίνουμε δικαιώματα για να μπορέσει το icinga2 να το διαβάσει. Αυτό γίνεται διότι ο nagios χρήστης δεν έχει δικαιώματα καθώς η check_logfiles διαβάζει το path /var/temp/ που είχε οριστεί εξ' αρχής, επόμενος προκειμένου να αποκτήσει τα δικαιώματα που χρειάζεται το αλλάζουμε απο root σε nagios, γράφοντας στην γραμμή εντολών στο path /usr/lib/nagios/plugins/ την εντολή:

```
Chown –R nagios: nagios check_logfiles_config
```

όπου το πρώτο nagios όπως βλέπουμε αφορά τον owner, ενώ το δεύτερο nagios αφορά το group στο οποίο ανήκει.

Την ίδια αλλαγή απο root σε nagios κάνουμε αντίστοιχα και στο path /etc/icinga2/conf.d γράφοντας την εντολή:

```
Chown –R nagios: nagios  .
```

Βημα 3ο: Στη συνέχεια, για να δούμε σε ποια group ανήκει το αρχείο έτσι ώστε να το εντάξουμε στην ομάδα χρηστών του adm για να έχει δικαιώματα το icinga2 στο αρχείο web προκειμένου να μπορεί να το διαβάζει, ορίζουμε την ομάδα γράφοντας την εντολή:

```
usermod –a –G adm nagios
```

και με την εντολή μας εμφανίζει το αποτέλεσμα της παραπάνω ενέργειας.

```
id nagios
```

Βήμα 4ο: Για να μπορέσουμε να τρέξουμε το plugin που φτιάξαμε, στο icinga web 2, χρειάζεται να φτιάξουμε και το κατάλληλο το command του με την εντολή που θα εκτελεί καθώς επίσης και τον server που θέλουμε να παρακολουθήσουμε. Για να γίνει αυτό, πάμε στον φάκελο /etc/icinga2/conf.d/ και ανοίγουμε το αρχείο commands.conf ώστε να φτιάξουμε τις εντολές που θέλουμε. Ξεκινάμε γράφοντας object CheckCommand και στη συνέχεια το

όνομα με το οποίο θέλουμε να την καλούμε το οποίο είναι “check_logfiles_config”. Μέσα σε αυτό κάνουμε import το plugin-check-command και δίνουμε στην μεταβλητή command το plugin που θέλουμε. Τέλος για να τρέξει το plugin πρέπει να δημιουργήσουμε τα arguments που απαιτούνται όπως φαίνονται στην παρακάτω εικόνα.

Εικόνα 21: Αρχείο εντολής check command

3.2.8 Δεδομένα απόδοσης

Ο αριθμός των γραμμών που σαρώθηκαν, καθώς και ο αριθμός του προτύπου το οποίο ταιριάζει (critical, warning και unknown) επισυνάπτονται στην έξοδο του plugin σε μορφή δεδομένων απόδοσης. Εάν θέλουμε έχουμε την δυνατότητα να το αλλάξουμε αυτό, χρησιμοποιώντας την επιλογή noperfdata.

```
check_logfiles --logfile /var/adm/messages \
   --criticalpattern 'Failed password' --tag ssh
CRITICAL - (4 errors) - May  9 11:33:12 localhost sshd[29742] Failed password for invalid
user8 ... |ssh_lines27 ssh_warnings=0 ssh_criticals=4 ssh_unknowns=0

check_logfiles --logfile /var/adm/messages \
   --criticalpattern 'Failed password' --tag ssh --noperfdata
CRITICAL - (2 errors) - May  9 11:58:48 localhost sshd[29813] Failed password for invalid
user8 ...
```

3.2.9 Σενάρια

Είναι πιθανόν να εκτελεστούν ορισμένα εξωτερικά σενάρια έξω από τα check_logfiles. Κάτι τέτοιο μπορεί να γίνει στη φάση εκκίνησης ($prescript), πριν από τη λήξη ($postscript) ή κάθε

φορά που ένα πρότυπο ταιριάζει με μια γραμμή, όπως φαίνεται στο παραπάνω παράδειγμα. Με την επιλογή "smartscript" η έξοδος και ο κώδικας εξόδου (exitcode) του σεναρίου αντιμετωπίζονται σαν να ταιριάζουν με το αρχείο καταγραφής (logfile) και περιλαμβάνονται στο συνολικό αποτέλεσμα. Η επιλογή "supersmartscript" αντικαθιστά την έξοδο και το exitcode του σεναρίου με εκείνα που ταιριάζουν.

Η Pre- και η Postscript που δηλώνονται ως supersmart scripts επηρεάζουν άμεσα τη διαδικασία του check_logfiles. Η επιλογή "supersmartprescript" προκαλεί την άμεση διακοπή του check_logfiles αν η prescript έχει έναν μη μηδενικό κωδικό εξόδου. Σε αυτήν την περίπτωση η έξοδος και το exitcode του check_logfiles αντιστοιχούν σε εκείνα του prescript. Με την επιλογή “supersmartpostscript” η έξοδος και το exitcode του check_logfiles μπορεί να προσδιοριστεί από το postscript. Έτσι, είναι πιθανή μια πιο ουσιαστική έξοδος.

3.3 Ένταξη στο Icinga

Στην περίπτωση που έχουμε μόνο μία υπηρεσία η οποία χρησιμοποιεί το check_logfile μπορούμε να προγραμματίσουμε το αρχείο ρυθμίσεων μας (config file) σε services.cfg / nrpe.cfg.

Βήμα 5ο: Το επόμενο βήμα μας είναι να φτιάξουμε τον host που θέλουμε να παρακολουθούμε. Έτσι λοιπόν δημιουργούμε ένα αρχείο και το ονομάζουμε webms.conf και μέσα σε αυτό γράψαμε τον host που θέλουμε.

```
object Host "webms"{
        address = "83.212.240.12"
        check_command = "hostalive"
        vars.snmp_address = "83.212.240.12"
        vars.snmp_community = "private"
        vars.snmp_v2 = "true"
        vars.snmp_v3 = "false"
        vars.os = "Linux"

}
```

Εικόνα 22: Το αρχείο webms.conf

Βήμα 6ο: Επόμενο βήμα είναι να βάλουμε σε λειτουργία το command που φτιάξαμε νωρίτερα σε λειτουργία. Για να γίνει αυτό ανοίγουμε το αρχείο services.conf και προσθέτουμε τα services που θέλαμε. Αρχικά δίνουμε το όνομα στο service που θα εκτελείτε και στη

συνέχεια κάνουμε import το generic-service και δηλώνουμε για ποιον host θα εκτελείτε αυτό το service. Στη συνέχεια δηλώνουμε ποια command θέλουμε να τρέξουμε και δίνουμε τιμές στα arguments.

```
apply Service "check" {
  import "generic-service"

  check_command = "check_logfiles_config"

  assign where host.vars.os == "Linux"

  vars.file = "/usr/lib/nagios/plugins/check_logfiles_config"
}
```

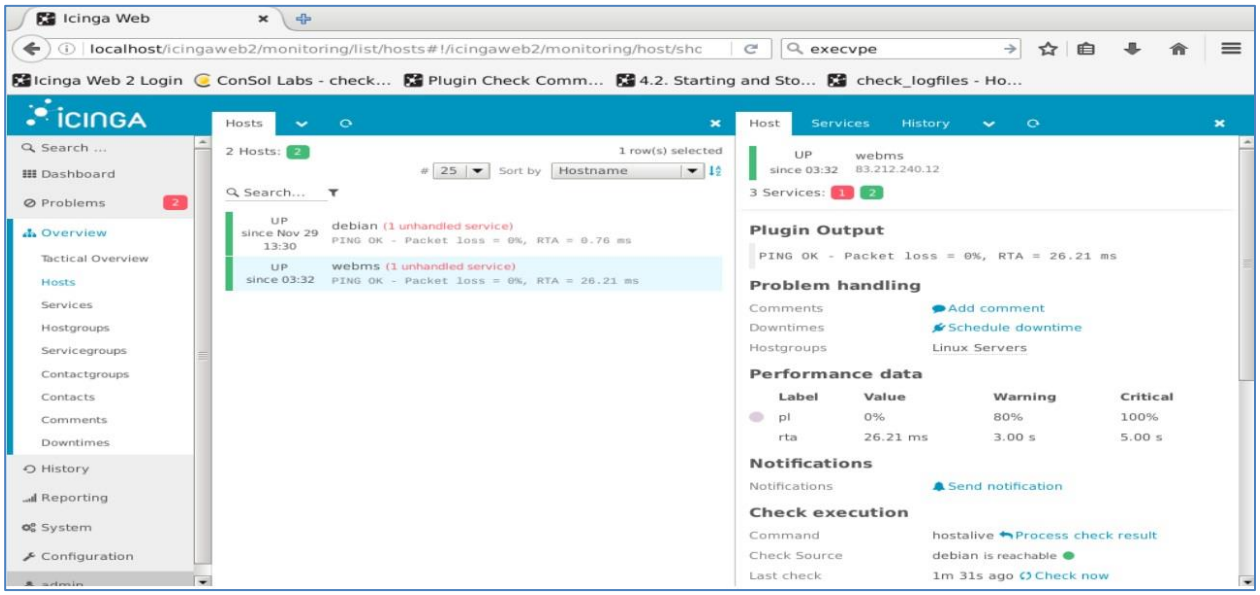
Εικόνα 23: Το αρχείο services.conf

Βήμα 7ο: Τέλος κάνουμε restart και reload το icinga 2 με τις εντολές:

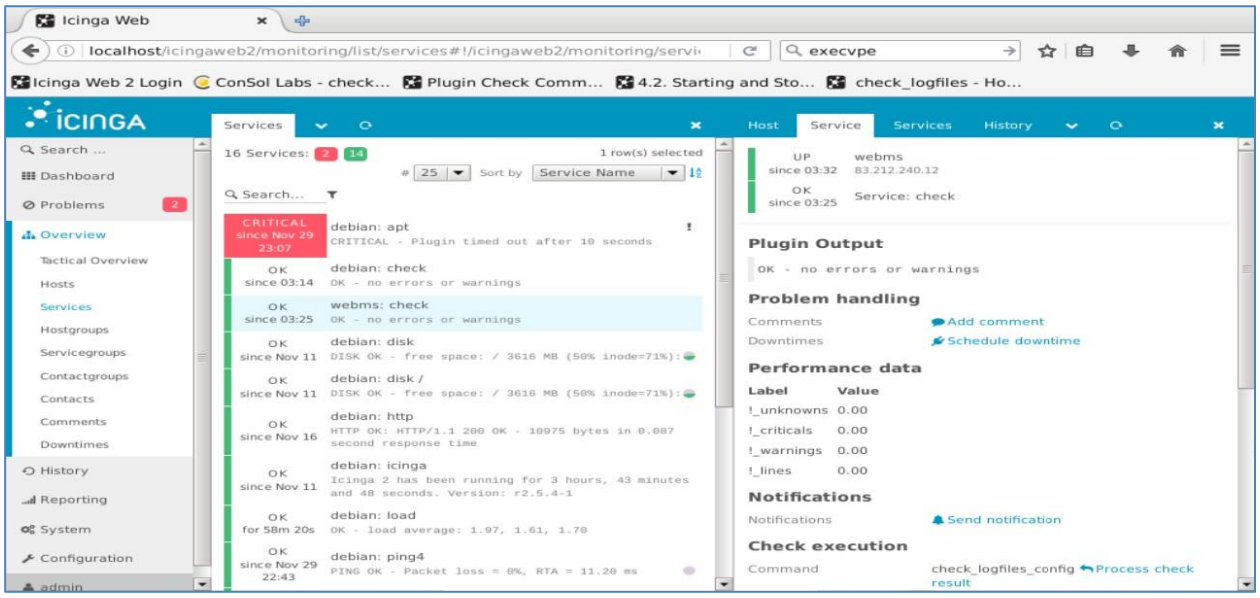
```
Service icinga2 restart

Service icinga2 reload
```

και ανοίγουμε το icinga web 2 για να δούμε τις αλλαγές μας όπως βλέπουμε στις παρακάτω εικόνες.



Εικόνα 24: Ο webms host στο περιβάλλον του Icinga web 2



Εικόνα 25: Οι webms services στο περιβάλλον του Icinga web 2

Κεφαλαιο 4

4.1 Port Scanning

Ο όρος Port Scanning αναφέρεται σε μία από τις πιο δημοφιλείς τεχνικές ανίχνευσης ανοικτών δικτυακών θυρών, που χρησιμοποιείται από τους εισβολείς με στόχο την άντληση πληροφοριών σε ένα απομακρυσμένο δίκτυο (remote network). Όλα τα μηχανήματα τα οποία βρίσκονται συνδεδεμένα σε ένα τοπικό δίκτυο (LAN – Local Area Network) ή στο διαδίκτυο, τρέχουν υπηρεσίες οι οποίες ακούνε σε κάποιες γνωστές ή μη θύρες (ports). Η σάρωση μίας θύρας (port scanning) βοηθά τον εισβολέα να βρει τις διαθέσιμες ελεύθερες, ώστε να αποκτήσει πρόσβαση στον υπολογιστή κάποιου χρήστη και να λειτουργήσει κακόβουλα.

Η τεχνική αυτή προσπαθεί να ανιχνεύσει τις αδυναμίες του χρήστη, ενώ η αποστολή μηνύματος ανίχνευσης σε μία θύρα, καθώς και η απάντηση στο απεσταλμένο μήνυμα καταδεικνύει την κατάσταση στην οποία βρίσκεται (ανοικτή ή κλειστή). Επιπλέον, συλλέγονται πληροφορίες για το είδος του λειτουργικού συστήματος το οποίο χρησιμοποιεί ο χρήστης και δεδομένα τα οποία πιθανόν να χρησιμοποιηθούν μελλοντικά.[14]

Το port-scanner στέλνει client αιτήματα σε μια server port ενός host προκειμένου να ανιχνέυσει αν η υποψήφια port είναι ενεργή. Ορισμένες από τις καταστάσεις που μπορεί να βρίσκεται μια θύρα είναι οι ακόλουθες:

- Open ή Accepted: Το υποψήφιο πρός σάρωση host στέλνει μια απάντηση υποδεικνύοντας οτι μια υπηρεσία "ακούει" για αιτήματα στη συγκεκριμένη port.
- Closed ή Denied ή Not Listening: Η απάντηση η οποία έρχεται από το host υποδεικνύει οτι οι συνδέσεις σε αυτή την port δεν είναι εφικτές.
- Filtered, Dropped ή Blocked: Δεν υπάρχει απάντηση από το host.

Στη συνέχεια παρουσιάζονται ορισμένες απο τις τεχνικές port scanning που χρησιμοποιούνται και είναι οι ακόλουθες:

- TCP Scanning: Αυτός ο τύπος σάρωσης συνδέεται με το target-port και ολοκληρώνει ένα πλήρες TCP three way handshake (SYN, SYN/ACK και ACK) και εντοπίζεται εύκολα από το target-system.

- SYN Scannning: Χαρακτηρίζεται και σαν half-scanning επειδή δεν πραγματοποιείται μια πλήρης TCP σύνδεση γιατί στέλνεται μόνο ένα πακέτο SYN στην target- θύρα.
- TCP FIN Scanning: Στέλνει ένα πακέτο FIN στην target θύρα και με βάση το RFC 793, το target-system θα πρέπει να στείλει πίσω ένα RST για όλες τις κλειστές θύρες. Αυτή η τεχνική συνήθως λειτουργεί σε TCP/IP stacks που βασίζονται στο UNIX.
- TCP Xmas Tree Scanning: Στέλνει ένα πακέτο FIN, URG και PUSH στην target θύρα και με βάση το RFC 793, το target-system θα πρέπει να στείλει πίσω ένα RST για όλες τις κλειστές θύρες.
- TCP Null Scanning: Απενεργοποιεί όλα τα Flags με βάση το RFC 793, το target-system θα πρέπει να στείλει πίσω ένα RST για όλες τις κλειστές θύρες.
- TCP ACK Scanning: Χρησιμοποιείται για την ανίχνευση firewall και των κανόνων του.
- TCP Windows Scanning: Εντοπίζει ανοιχτές θύρες, καθώς επίσης και θύρες με φίλτρα ή χωρίς φίλτρα σε μερικά συστήματα (όπως AIX και FreeBSD-το γνήσιο UNIX-) εξαιτίας ενός παράξενου τρόπου που αναφέρεται η αλλαγή μεγέθους στα TCP Windows.
- TCP RPC Scanning: Αφορά συγκεκριμένα συστήματα UNIX και χρησιμοποιείται για τον εντοπισμό και των Remote Procedure Call (RPC) ports όπως το σχετικό τους πρόγραμμα και αριθμό έκδοσης.
- UDP Scanning: Στέλνει ένα πακέτο UDP στην target-θύρα και εαν η target-θύρα αποκριθεί με ένα μήνυμα "ICMP port unreachable", τότε η θύρα είναι κλειστή. Αντιθέτως αν δεν ληφθεί το μήνυμα τότε οδηγούμαστε στο συμπέρασμα οτι η θύρα είναι ανοιχτή. [15]

4.2 Η έννοια του nmap

Το Nmap είχε σχεδιαστεί για να σαρώνει γρήγορα μεγάλα δίκτυα, αν και δουλεύει καλά ενάντια σε απλούς κεντρικούς υπολογιστές(hosts). Το nmap (Network Mapper) είναι ένα δωρεάν εργαλείο ανοικτού λογισμικού και ενώ χρησιμοποιείται συνήθως για τους ελέγχους ασφαλείας, πολλά συστήματα και οι διαχειριστές του δικτύου το θεωρούν χρήσιμο για εργασίες ρουτίνας, όπως η απογραφή του δικτύου, τη διαχείριση των προγραμμάτων αναβάθμισης των υπηρεσιών, και να παρακολουθεί τους hosts ή το χρόνο λειτουργίας των υπηρεσιών. Έχει ως βασικό στόχο την ανίχνευση δικτυακών συσκευών και συστημάτων και τον έλεγχό τους με διάφορους και διαφορετικούς τρόπους ως προς το λογισμικό που διαθέτουν,

τις παρεχόμενες υπηρεσίες και τις ανοιχτές πόρτες στις οποίες μπορούν να συνδεθούν απομακρυσμένα νόμιμοι αλλά και κακόβουλοι χρήστες.[16]

Για να πετυχαίνει το σκοπό του αποστέλλει στους στόχους, δηλαδή στα μηχανήματα που ελέγχει, ειδικά διαμορφωμένα δικτυακά πακέτα με πρωτότυπους τρόπους, κι έπειτα αναλύει τις απαντήσεις που λαμβάνει. Από την ανάλυση αυτή το nmap είναι σε θέση να εντοπίζει:

- τα ενεργά hosts από το σύνολο εκείνων που ελέγχει
- τα ανοικτά ports κάθε host
- το λειτουργικό σύστημα κάθε host
- την έκδοση κάθε λειτουργικού
- τις όποιες δικτυακές υπηρεσίες προσφέρει κάθε host
- τις εκδόσεις των υπηρεσιών αυτών
- τα όποια firewalls υπάρχουν στο δίκτυο.

Επιπρόσθετα, χάρη στη μηχανή scripting που ενσωματώνει αλλά και στην πληθώρα των έτοιμων scripts που διατίθενται ελεύθερα γι' αυτό, το nmap έχει τη δυνατότητα να εντοπίζει αδυναμίες (vulnerabilities), backdoors ή malware, να εξαπολύει επιθέσεις denial of service ή brute force, να εκμεταλλεύεται γνωστά bugs σε υπηρεσίες. Το nmap αρχικά διατίθετο μόνο για Linux. Σήμερα παρέχονται binaries για πολλά άλλα Unix-like OSes, για το OS X αλλά και για τα Windows. Αν και το nmap λειτουργεί από τη γραμμή εντολών συχνά συνοδεύεται κι από το Zenmap, ένα εύχρηστο front-end για το περιβάλλον γραφικών. [17]

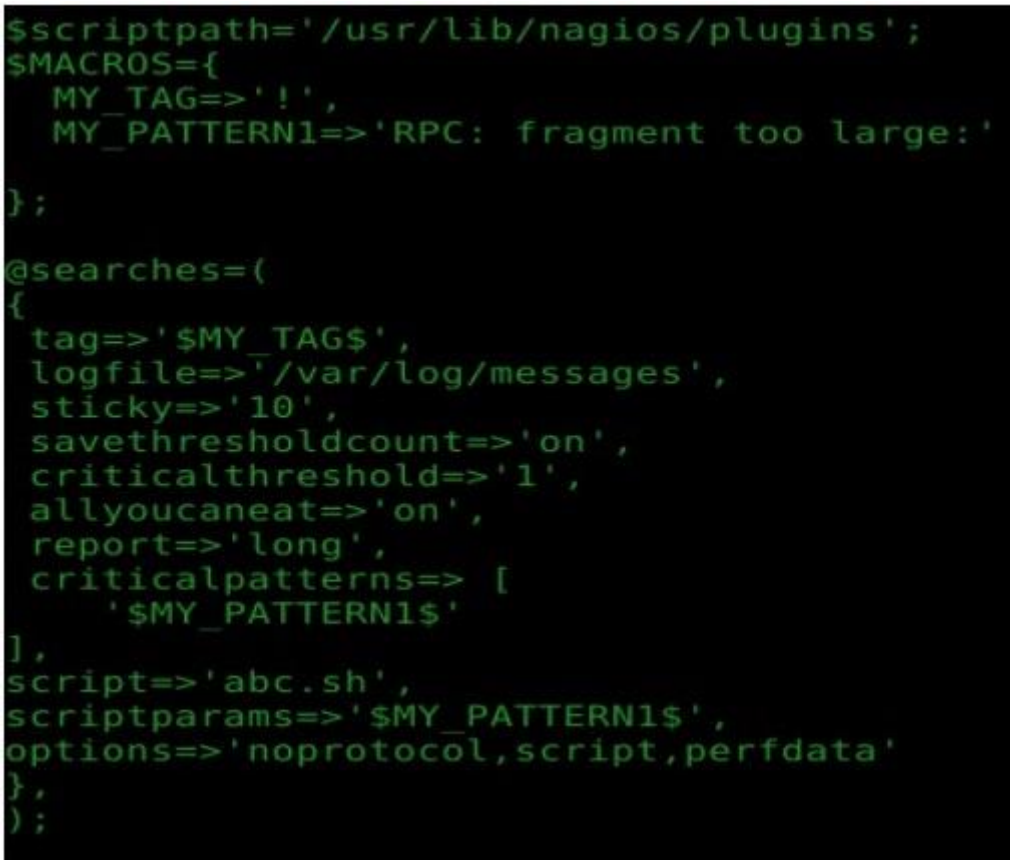
Όσον αφορά τη λειτουργεία του, το Nmap χρησιμοποιεί ακατέργαστα IP πακέτα με νέους τρόπους για να καθορίσει ποιοι κεντρικοί υπολογιστές(hosts) είναι διαθέσιμοι στο δίκτυο, ποιες υπηρεσίες (το όνομα της εφαρμογής και την έκδοση) αυτοί οι hosts προσφέρουν, τι λειτουργικά συστήματα (OS και εκδόσεις) τρέχουν, το είδος του πακέτου και φίλτρων/firewalls που είναι σε χρήση, και δεκάδες άλλα χαρακτηριστικά. Για την επίτευξη του στόχου του, το Nmap στέλνει ειδικά δημιουργημένα πακέτα στον κεντρικό υπολογιστή "στόχο" και στη συνέχεια αναλύει τις απαντήσεις. Σε αντίθεση με πολλά απλά Port scanners, που μόνο στέλνουν πακέτα σε κάποιο προκαθορισμένο σταθερό ρυθμό, το Nmap παρακολουθεί τις συνθήκες του δικτύου (διακυμάνσεις λανθάνουσας κατάστασης, συμφόρηση του δικτύου, η παρέμβαση με στόχο τη σάρωση) κατά τη διεξαγωγή του. [16]

Η έξοδος από το Nmap είναι μια λίστα από σαρωμένους στόχους, με συμπληρωματικές πληροφορίες και κάθε μια εξαρτάται από τις επιλογές που χρησιμοποιήθηκαν. Μεταξύ των βασικών πληροφοριών είναι ο σημαντικός πινάκας των ports (interesting ports table). Αυτός ο πίνακας παραθέτει τον αριθμό θύρας και το πρωτόκολλο, το όνομα υπηρεσίας, και την κατάσταση. Η κατάσταση είναι είτε ανοικτή, φιλτραρισμένη, κλειστή ή αφιλτράριστη. Άνοιγμα, σημαίνει ότι μια εφαρμογή στον υπολογιστή-στόχο εντοπίζει για συνδέσεις/πακέτα της συγκεκριμένης θύρας. Φιλτραρισμένο, σημαίνει ότι ένα τείχος προστασίας, ένα φίλτρο, ή άλλο εμπόδιο του δικτύου μπλοκάρει τη θύρα έτσι ώστε το Nmap δεν μπορεί να πει αν είναι ανοιχτή ή κλειστή. Κλειστό, σημαίνει οι θύρες δεν έχουν εφαρμογή στο να εντοπίζει συνδέσεις και πακέτα, αν και θα μπορούσαν να ανοίξουν σε οποιαδήποτε στιγμή.

Οι θύρες είναι ταξινομημένες ως αφιλτράριστες όταν ανταποκρίνονται στους ανιχνευτές Nmap, αλλά το Nmap δεν μπορεί να προσδιορίσει εάν είναι ανοικτές ή κλειστές. Το Nmap αναφέρει τους συνδυασμούς κατάστασης ανοιχτές/φιλτραρισμένες και κλειστές/φιλτραρισμένες όταν δεν μπορεί να προσδιορίσει ποιο από τα δύο καταστάσεις περιγράφουν μια θύρα. Ο πινάκας της θύρας μπορεί επίσης να περιλαμβάνει λεπτομέρειες έκδοσης του λογισμικού όταν έχει ζητηθεί έκδοση ανίχνευσης. Όταν ένα πρωτόκολλο σάρωσης IP έχει ζητηθεί (-sO), το Nmap παρέχει πληροφορίες σχετικά με τα υποστηριζόμενα πρωτόκολλα IP παρά για ανιχνεύσιμες θύρες.[18]

4.3 Έλεγχος ασφάλειας στο webms

Στη συνέχεια επιχειρούμε να πραγματοποιήσουμε μια επίθεση στον webms host που έχουμε δημιουργήσει, τρέχοντας ένα port scan με τη βοήθεια του nmap το οποίο αναφέραμε πιο πριν, όπου θα πρέπει να μας εμφανίσει το αποτέλεσμα αυτό στο icinga web 2 χρησιμοποιώντας το plugin που φτιαξαμε και χρησιμοποιήσαμε στην εργασία το οποίο φαίνεται παρακάτω στην εικόνα 26 . Αυτό γίνεται στα βήματα που ακολουθούν.



Εικόνα 26. Το plugin check_log_file

Βήμα 8ο: Εκτελώντας την εντολή:

```
tail –f /var/log/messages
```

Εμφανίζεται το αποτέλεσμα που βλέπουμε στην εικόνα 27 όπου μας επιστρέφει τις τελευταίες γραμμές του αρχείου.

```
Feb 2 11:32:10 webms kernel: [8639843.594896] RPC: fragment too large: 218762506
Feb 2 11:32:10 webms kernel: [8639843.595840] RPC: fragment too large: 1195725856
Feb 2 11:32:10 webms kernel: [8639843.597051] RPC: fragment too large: 1330664521
Feb 2 11:32:10 webms kernel: [8639843.597932] RPC: fragment too large: 1330664521
```

Εικόνα 27. Αποτέλεσμα εκτέλεσης του nmap

Το αποτέλεσμα αυτό το γράφουμε στο pattern του plugin Check_logfiles_config που είδαμε πιο πάνω ορίζοντας έτσι το pattern μας ώστε σε περίπτωση που κάποιος άλλος προσπαθήσει να κάνει nmap με το συγκεκριμένο pattern που έχουμε ορίσει να μπορέσουμε να το καταλαβουμε εμφανίζοντας μας το αποτέλεσμα στην οθόνη του icinga web 2.

Βήμα 9ο: Στη συνέχεια, για να κάνουμε το nmap εκτελούμε την εντολή:

```
nmap  -sV –p  1-65535 webms.hua.gr
```

Με την εντολή αυτή, στην ουσία λέμε να τρέξει το nmap -sV, δηλαδή για το Service Version Detection κάνωντας port scanning τις θύρες απο 1-65535, για το μηχάνημα webms.hua.gr και γίνεται καταγραφή στο var_log_messages.! και το αποτέλεσμα εμφανίζεται στην οθόνη του icinga web 2.

Βημα 10ο: Τέλος, για να μπορεί να γίνει η καταγραφή αυτή στο var_log_messages.! πρέπει πρώτα να του έχουμε δώσει δικαιώματα χρήστη nagios απο root, το οποίο γίνεται με την παρακάτω εντολή:

```
Chown nagios:nagios check_logfiles_config._var_log_messages.!
```

Συμπεράσματα

Η ασφάλεια των δικτύων αποτελεί σημαντικό θέμα το οποίο χρήζει ιδιαίτερης σημασίας όσον αφορά τους τρόπους με τους οποίους γίνεται ο έλεγχος ασφάλειας σε αυτά καθώς και την αποτελεσματικότητα και ακεραιότητα τους.

Στην παρούσα διπλωματική εργασία μπορέσαμε με την δημιουργία και παραμετροποίηση ενός plugin του check_log_file να αντιληφθούμε την βασική λειτουργιά του ελέγχου ασφαλείας ενός δικτύου και να καταλάβουμε τον τρόπο λειτουργιάς του λαμβάνοντας ορισμένα αποτελέσματα κατά την εκτέλεση του.

Δοκιμάσαμε σε τρεις διαφορετικούς διακομιστές του κέντρου πληροφορικής και δικτύων να τρέξουμε το configuration αρχείο που αναφέραμε στο 4ο κεφάλαιο και παρατηρήσαμε ότι και στους τρεις το /var/log/messages είχε διαφορετική συμπεριφορά όταν κάναμε ελεγχόμενο port scanning, έτσι δεν μπορέσαμε να εντοπίσουμε κάποιο πρότυπο πέραν αυτού που αναφέραμε ήδη στο 4ο κεφάλαιο για τον διακομιστή webms. Εκτός από το ότι δεν μπορέσαμε να έχουμε ένα ξεκάθαρο πρότυπο αναζήτησης ακόμα και αυτό που δοκιμάσαμε στον προαναφερόμενο διακομιστή ότι δουλεύει δυστυχώς δεν παρουσίαζε αποτελέσματα στο web περιβάλλον του icinga.

Συμπέρασματικα καταλήγουμε ότι χρειάζεται περαιτέρω διερεύνηση των σωστών προτύπων και της λειτουργιάς ενσωμάτωσης του plugin κάτι το οποίο δεν ήταν εφικτό να γίνει στα πλαίσια της παρούσας διπλωματικής εργασίας λόγο των στενών χρονικών ορίων του προγράμματος μεταπτυχιακών σπουδών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. How to enhance your network for the future [Ηλεκτρονικός σύνδεσμος]: http://www.computerweekly.com/feature/Network-monitoring-Essential-Guide
2. Network Monitoring Definition and Solutions - [Ηλεκτρονικός σύνδεσμος]: http://www.cio.com/article/2438133/networking/network-monitoring-definition-and-solutions.html#whatis
3. Basics of Network Monitoring - [Ηλεκτρονικός σύνδεσμος]: http://www.solarwinds.com/basics-of-network-monitoring
4. Network Monitoring Basics - [Ηλεκτρονικός σύνδεσμος]: http://www.slideshare.net/maximillianx/network-monitoring-basics
5. Το πρωτόκολλο SNMP - [Ηλεκτρονικός σύνδεσμος]: http://conta.uom.gr/conta/ekpaideysh/metaptyxiaka/technologies_diktywn/teaching_m/management/snmp.htm
6. The Top 20 Free Network Monitoring and Analysis Tools for Sys Admins - [Ηλεκτρονικός σύνδεσμος]: http://techtalk.gfi.com/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins/comment-page-1/
7. About Icinga 2 - [Ηλεκτρονικός σύνδεσμος]: https://www.icinga.com/
8. Icinga Web 2.0.0 - [Ηλεκτρονικός σύνδεσμος]: https://www.icinga.com/2015/10/02/icinga-web-2-0-0-the-final-version-is-unleashed/
9. Icinga 2 – Architecture - [Ηλεκτρονικός σύνδεσμος]: https://www.icinga.com/products/icinga-2/architecture/
10. About Icinga 2 - [Ηλεκτρονικός σύνδεσμος]: https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/getting-started#installing-database-postgresql-server
11. How to install Icinga 2 on ubuntu - [Ηλεκτρονικός σύνδεσμος]: http://linoxide.com/ubuntu-how-to/install-icinga2-ubuntu-16-04/
12. About check_logfiles - [Ηλεκτρονικός σύνδεσμος]: https://exchange.nagios.org/directory/Plugins/Log-Files/check_logfiles/details
13. About check_logfiles, lab - [Ηλεκτρονικός σύνδεσμος]: https://labs.consol.de/nagios/check_logfiles/

14. What is Port Scanning? - [Ηλεκτρονικός σύνδεσμος]:
https://el.wikipedia.org/wiki/Port_Scanning
15. Nmap & port scanning - [Ηλεκτρονικός σύνδεσμος]:
https://www.itbloom.com/?cat=31
16. What is Nmap - [Ηλεκτρονικός σύνδεσμος]:
https://el.wikipedia.org/wiki/Nmap
17. Το nmap από την αρχή - [Ηλεκτρονικός σύνδεσμος]:
https://deltahacker.gr/to-iperocho-nmap-apo-tin-archi
18. Nmap (1) - Linux man page - [Ηλεκτρονικός σύνδεσμος]:
https://linux.die.net/man/1/nmap