Παρακολούθηση υπηρεσιών Cloud μέσω IDS

Πειραιάς
Ιούνιος 2016

Διπλωματική Εργασία

του Σμπόνια Ορέστη

Επιβλέπων Καθηγητής: Λαμπρινουδάκης Κωνσταντίνος

Τμήμα Ψηφιακών Συστημάτων

Μεταπτυχιακό Ασφάλειας Ψηφιακών Συστημάτων

Πανεπιστήμιο Πειραιά

Περιεχόμενα εικόνων

Περιεχόμενα

Εισαγωγή

Στις μέρες μας η χρήση του διαδικτύου έχει αυξηθεί σε τέτοιο βαθμό που ο απλός χρήστης είναι ανά πάσα στιγμή online μέσω του υπολογιστή, του tablet ή του κινητού του. Αυτό έχει ως αποτέλεσμα οι εταιρίες παροχής υπηρεσιών να προσφέρουν με χαμηλό ή καθόλου κόστος ποικίλες εφαρμογές που έχουν να κάνουν με το cloud. Το πιο συνηθισμένο παράδειγμα αποτελεί ο δωρεάν αποθηκευτικός χώρος μερικών GB στα οποία ο χρήστης μπορεί να ανεβάσει αρχεία, φωτογραφίες και βίντεο.

Οι εταιρίες από την άλλη έχουν στραφεί στις υπηρεσίες cloud κυρίως λόγω του χαμηλότερου κόστους καθώς για παράδειγμα εξαλείφεται το κόστος συντήρησης εξοπλισμού. Επιπλέον, οι υπηρεσίες αυτές προσφέρουν μεγαλύτερη ευκολία και εξοικονόμηση χρόνου σε αυτόν που τις διαχειρίζεται καθώς έχει φερειπείν τη δυνατότητα να πραγματοποιεί μαζικές εγκαταστάσεις servers άμεσα και εύκολα.

Παρόλα αυτά, το cloud κρύβει αρκετούς κινδύνους όσον αφορά την ασφάλεια των πληροφορίων που έχουν να κάνουν με τις βασικές αρχές της ασφάλειας πληροφοριών την Εμπιστευτικότητα, την Διαθεσιμότητα και την Ακεραιότητα. Όλες οι εταιρίες παροχής υπηρεσιών cloud διαβεβαιώνουν τους τελικούς χρήστες ότι τα δεδομένα τους είναι υπέρ το δέον ασφαλή. Η πραγματικότητα όμως είναι διαφορετική καθώς έχουν καταγραφεί διαφόρων ειδών επιθέσεις σε περιβάλλοντα cloud computing που έχουν να κάνουν είτε με διαγραφή δεδομένων, είτε με μη εξουσιοδοτημένη πρόσβαση και τροποποίηση πληροφοριών.

Οι διαχειριστές των δικτυακών συσκευών ή των servers χρησιμοποιούν σε συνδυασμό με τα firewalls, Intrusion Detection Systems (IDS) για την αντιμετώπιση επιθέσεων. Τα IDS αναλύουν την δικτυακή κίνηση ή τις αλλαγές που γίνονται σε κάποιο μηχάνημα και αν εντοπίσουν κακόβουλη κίνηση ειδοποιούν τον διαχειριστή τους με κάποιο alert. Σε περιβάλλοντα cloud δεν έχει επικρατήσει η εγκατάσταση IDS.

Σκοπός της εργασίας αυτής είναι η ανάπτυξη ενός Intrusion Detection System σε περιβάλλον cloud το οποίο θα πραγματοποιεί ελέγχους για μη επιθυμητή κίνηση.

1. Intrusion Detection System

1.1 Ορισμός

Intrusion Detection (Ανίχνευση Εισβολής) είναι η διαδικασία εντοπισμού κακόβουλης κίνησης σε ένα δίκτυο ή σε ένα σύστημα. Intrusion Detection System (Σύστημα Ανίχνευσης Εισβολών) είναι η εφαρμογή ή το φυσικό σύστημα που παρακολουθεί την δικτυακή κίνηση ή το σύστημα ούτως ώστε να εντοπίσει τυχόν δραστηριότητα που παραβιάζει τις πολιτικές ασφαλείας. Η δραστηριότητα αυτή μπορεί να προκληθεί από κάποιο malware, κάποιον επιτιθέμενο που προσπαθεί να αποκτήσει πρόσβαση ή ακόμα και χρήστες των συστημάτων που προσπαθούν να αποκτήσουν περισσότερα δικαιώματα από αυτά που έχουν (π.χ. δικαιώματα administrator). Τα περισσότερα IDS μπορούν να λειτουργήσουν και σαν Intrusion Prevention Systems (Συστήματα Αποτροπής Εισβολών) εμποδίζοντας την ολοκλήρωση των επιθέσεων.

1.2 Είδη IDS

Τα IDS χωρίζονται σε διάφορα είδη ανάλογα με το τι παρακολουθούν:

- Network-Based IDS (NIDS)
- Host-Based IDS (HIDS)

1.2.1 Network-Based IDS

Τα Network-Based IDS αποτελούν το πιο κοινό είδος Intrusion Detection Systems. Η εγκατάστασή τους είναι εύκολη και η θέση τους στο δίκτυο είναι τέτοια ώστε να έχουν τη δυνατότητα να ελέγχουν την δικτυακή κίνηση όλων των εμπλεκόμενων συστημάτων. Ο έλεγχος της κίνησης πραγματοποιείται σε όλα τα επίπεδα του μοντέλου OSI. Στην ίδια κατηγορία υπάγονται και τα Wireless IDS που εξειδικεύονται μόνο στην ασύρματη επικοινωνία και έχουν την δυνατότητα να ελέγχουν αν κάποιος εξωτερικός χρήστης προσπαθεί να αποκτήσει πρόσβαση σε κάποιο access point ή αν υπάρχουν rogue access points.

1.2.2 Host-Based IDS

Τα Host-Based IDS εγκαθίστανται σε συστήματα που χρήζουν παρακολούθησης και κάνουν ελέγχους στις ρυθμίσεις των συστημάτων, στα system calls, στα logs, στα processes που τρέχουν, στην πρόσβαση και στην τροποποίηση των αρχείων και στις αλλαγές σε λειτουργικό ή εφαρμογές. Επιπλέον, ελέγχουν την ακεραιότητα των δεδομένων των συστημάτων, την εφαρμογή των πολιτικών ασφαλείας και κάνουν αναζήτηση για rootkits [1] [2].

1.3 Πλεονεκτήματα κάθε είδους

1.3.1 Πλεονεκτήματα Network-Based IDS

Τα πλεονεκτήματα ενός NIDS είναι τα παρακάτω:

i. **Χαμηλό κόστος συντήρησης:** Το Network-Based IDS μπορεί να υλοποιηθεί σε οποιοδήποτε τμήμα του δικτύου και να παρακολουθεί την κίνηση όλων των συστημάτων του δικτύου χωρίς την εγκατάσταση και συντήρηση κάποιο λογισμικού σ' αυτά.

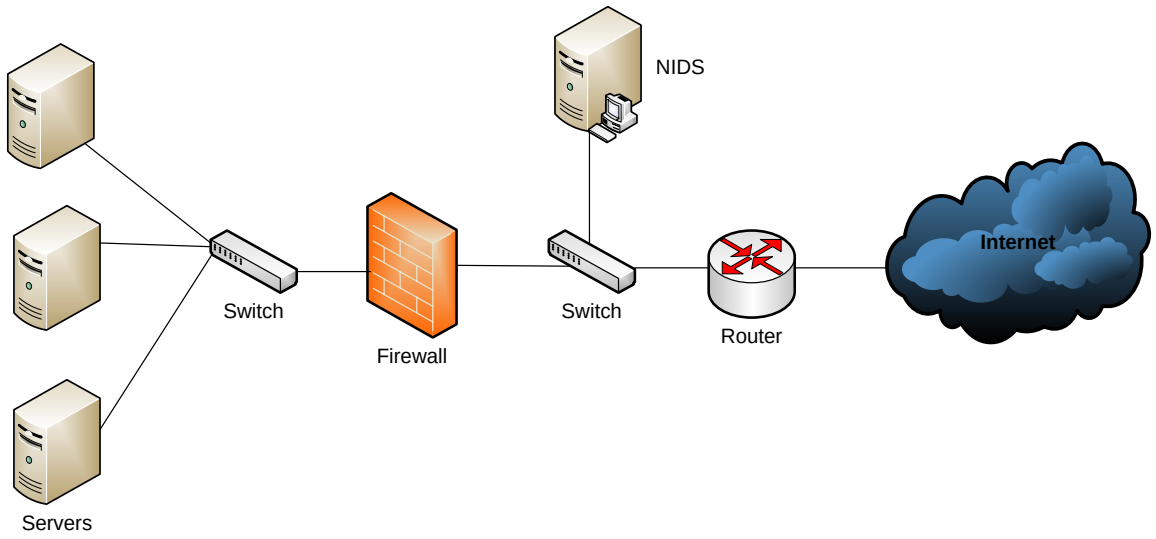
ii. **Απλό στην εφαρμογή:** Το NIDS είναι απλό κατά την εφαρμογή του, καθώς δεν επηρεάζει τα συστήματα ή την υποδομή του δικτύου που ελέγχει. Επίσης, δεν επηρεάζεται από τα διαφορετικά είδη λειτουργικών που έχουν τα συστήματα που παρακολουθεί.

iii. **Ανίχνευση επιθέσεων δικτύου:** Ένα NIDS μπορεί να ελέγξει επιθέσεις μέσω δικτύου σε αντίθεση με τα Host-Based NIDS που δεν έχουν αυτή τη δυνατότητα. Πιο συγκεκριμένα, ελέγχει τα headers όλων των πακέτων για τυχόν κακόβουλη κίνηση όπως για παράδειγμα οι επιθέσεις TCP SYN και IP fragmentation.

iv. **Διατήρηση στοιχείων:** Ένα NIDS χρησιμοποιεί τη live κίνηση του δικτύου για τον εντοπισμό επιθέσεων και έτσι δεν είναι εφικτή η διαγραφή των ενδείξεων της επίθεσης από τον επιτιθέμενο.

v. **Εντοπισμός σε πραγματικό χρόνο και άμεση αντίδραση:** Ένα Network-Based IDS ελέγχει την κίνηση σε πραγματικό χρόνο και έτσι εντοπίζει την κακόβουλη κίνηση τη στιγμή που αυτή πραγματοποιείται. Ανάλογα με τις ρυθμίσεις του NIDS, υπάρχει η δυνατότητα να διακοπεί η επίθεση αυτή πριν καν φτάσει σε κάποιο σύστημα και το θέσει σε κίνδυνο. Αντίθετα, ένα HIDS ελέγχει τις αλλαγές που γίνονται στο σύστημα και μέχρι να αντιδράσει το σύστημα έχει ήδη γίνει τεθεί σε κίνδυνο.

vi. **Εντοπισμός μη επιτυχημένων επιθέσεων:** Ένα NIDS μπορεί να χρησιμεύσει στον εντοπισμό μη επιτυχημένων επιθέσεων αν τοποθετηθεί πριν το firewall που συνδέει το εσωτερικό με το εξωτερικό δίκτυο όπως φαίνεται και παρακάτω. Στην περίπτωση αυτή, οι επιθέσεις που απορρίπτονται από το firewall και δεν φτάνουν το στόχο τους, έχουν εντοπιστεί και καταγραφεί από το NIDS και μπορούν να χρησιμοποιηθούν μελλοντικά για αποτροπή παρόμοιων επιθέσεων.



Εικόνα 1. NIDS πριν το firewall

1.3.2 Πλεονεκτήματα Host-Based IDS

Τα πλεονεκτήματα ενός HIDS είναι τα παρακάτω:

i. **Επιβεβαίωση της επιτυχίας ή αποτυχίας μιας επίθεσης:** Το HIDS χρησιμοποιεί τα logs του συστήματος που είναι εγκατεστημένο, για να καθορίσει αν μια επίθεση ήταν επιτυχημένη ή όχι. Αυτό έχει σαν αποτέλεσμα ο καθορισμός αυτός να γίνεται με μεγαλύτερη ακρίβεια και με λιγότερα false positives σε σχέση με ένα NIDS.
ii. **Παρακολούθηση των δραστηριοτήτων του συστήματος:** Ένα HIDS παρακολουθεί τις προσβάσεις των χρηστών σε αρχεία, τις αλλαγές των δικαιωμάτων των χρηστών στα αρχεία καθώς και τα login, logouts των χρηστών και τις πιθανές εγκαταστάσεις νέου λογισμικού. Οποιαδήποτε αλλαγή συμβαίνει σε επίπεδο συστήματος καταγράφεται μέσω του λειτουργικού σε logs και έτσι το Host-Based IDS μπορεί ενημερωθεί άμεσα για τις μη επιθυμητές αλλαγές σε αντίθεση με το NIDS.
iii. **Εντοπισμός επιθέσεων που ένα NIDS αποτυγχάνει να εντοπίσει:** Ένα HIDS μπορεί να εντοπίσει αλλαγές που έχει κάνει σε αρχεία του συστήματος ένας μη εξουσιοδοτημένος χρήστης. Έτσι τα HIDS είναι ικανά να προστατέψουν τα συστήματα είτε από εσωτερικούς, είτε από εξωτερικούς χρήστες.
iv. **Μη-χρήση hardware:** Το HIDS δεν χρειάζεται επιπλέον hardware για να εγκατασταθεί καθώς εγκαθίσταται στο σύστημα που θα παρακολουθεί.
v. **Χαμηλότερο κόστος εγκατάστασης:** Το Host-Based IDS έχει πολύ χαμηλότερο κόστος κατά την εγκατάστασή του σε σχέση με το NIDS που χρησιμοποιεί δικό του hardware [3].

1.4 Είδη IDS με βάση τους μηχανισμούς ανίχνευσης

Τα IDS χωρίζονται και σε διαφορετικά είδη με βάση τους μηχανισμούς ανίχνευσης εισβολών που διαθέτουν. Αυτά είναι:

- Signature-Based Detection
- Anomaly-Based Detection

1.4.1 Signature-Based Detection

Ένα IDS μπορεί να χρησιμοποιήσει signatures που βασίζονται σε υπάρχουσα κίνηση ούτως ώστε να αναγνωρίσει την μη επιθυμητή κίνηση. Με αυτόν τον τρόπο η αναγνώριση μιας επίθεσης μπορεί να γίνει εύκολα -με ελάχιστη παραμετροποίηση- και πολύ γρήγορα. Ο επιτιθέμενος όμως έχει τη δυνατότητα να τροποποιήσει κάποια πακέτα και να περάσει απαρατήρητος από τα Signature-Based IDS. Παρόλο που υπάρχει αυτή η αδυναμία, το Signature-Based IDS είναι πολύ ακριβές στο είδος των επιθέσεων που αναγνωρίζει.

1.4.2 Anomaly-Based Detection

Ένα Anomaly-Based IDS παρακολουθεί την δικτυακή κίνηση και εντοπίζει τα δεδομένα που περιέχουν λάθη, δεν είναι έγκυρα ή δεν είναι φυσιολογικά. Η μέθοδος αυτή χρησιμεύει στην ανίχνευση κίνησης νέου τύπου κίνησης για την οποία δεν υπάρχει κάποιο signature. Για παράδειγμα, το Anomaly-Based IDS αναγνωρίζει ότι ένα πακέτο του Internet Protocol είναι

τροποποιημένο άρα είναι μη φυσιολογικό αλλά δεν είναι ικανό να καταλάβει με ποιον τρόπο έχει τροποποιηθεί [2].

2. Cloud

2.1 Ορισμός

Το cloud αποτελεί το υπολογιστικό μοντέλο εκείνο που επιτρέπει την πρόσβαση σε κοινόχρηστους υπολογιστικούς πόρους όπως είναι servers, δίκτυα, αποθηκευτικός χώρος, εφαρμογές και υπηρεσίες όποτε ο χρήστης το επιθυμεί. Οι πόροι αυτοί μπορούν να διαχειριστούν εύκολα και γρήγορα προσφέροντας στον χρήστη πολλαπλές δυνατότητες όσον αφορά στην αποθήκευση και στην επεξεργασία δεδομένων [4] [5].

2.2 Μοντέλα Υπηρεσιών Cloud

Τα μοντέλα υπηρεσιών του cloud είναι τα ακόλουθα:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

2.2.1 IaaS

Το μοντέλο Infrastructure as a Service περιλαμβάνει τη βασική δομή του cloud που είναι οι δικτυακές λειτουργίες, οι υπολογιστικοί πόροι και ο αποθηκευτικός χώρος. Η διαχείριση όλων αυτών μπορεί να γίνει εύκολα και γρήγορα. Παραδείγματα IaaS αποτελούν το Amazon EC2, το Google Compute Engine και το Microsoft Azure.

2.2.2 PaaS

Το μοντέλο Platform as a Service περιλαμβάνει ένα περιβάλλον για ανάπτυξη και διαχείριση εφαρμογών, χωρίς να ασχολείται ο χρήστης με τη διαχείριση των πόρων, τη συντήρηση και ενημέρωση του λογισμικού και τη διαχείριση του υλικού. Παραδείγματα PaaS αποτελούν το Microsoft Azure και το Google App Engine.

2.2.3 SaaS

Το μοντέλο Software as a Service περιλαμβάνει ολοκληρωμένα προϊόντα (εφαρμογές) στα οποία την αποκλειστική διαχείριση έχει ο Cloud Service Provider. Οι τελικοί χρήστες δεν ασχολούνται με την διαχείριση ή την συντήρηση της υποδομής του cloud παρά μόνο με τη χρήση της εφαρμογής. Παραδείγματα SaaS αποτελούν τα Google Apps και το Microsoft 365 [4] [5] [6] [7] [8].

2.3 Μοντέλα Ανάπτυξης Cloud

Τα μοντέλα ανάπτυξης του cloud περιλαμβάνουν τα ακόλουθα είδη:

- Private Cloud
- Public Cloud
- Community Cloud

- Hybrid Cloud

2.3.1 Private Cloud

Private cloud ονομάζεται το cloud που η υποδομή του βρίσκεται στις εγκαταστάσεις της εταιρίας που το χρησιμοποιεί, με χρήση εργαλείων virtualization και διαχείρισης πόρων. Σκοπός του μοντέλου αυτού είναι αύξηση της χρήσης των υπαρχόντων υποδομών της εταιρίας καθώς δεν περιλαμβάνει πολλά από τα πλεονεκτήματα του «κλασικού» cloud computing όπως για παράδειγμα η άμεση αυξομείωση των πόρων.

2.3.2 Public Cloud

Public cloud ονομάζεται το cloud που η υποδομή του βρίσκεται στις εγκαταστάσεις του cloud provider. Το μεγάλο πλεονέκτημα του public cloud είναι η σημαντική μείωση του κόστους σε σχέση με το private. Ωστόσο, εγείρονται ζητήματα ασφάλειας πληροφοριών στο public cloud τα οποία δεν υπάρχουν στο private.

2.3.3 Community Cloud

Community cloud ονομάζεται το cloud που η υποδομή του βρίσκεται σε διάφορες εταιρίες που έχουν κοινούς τομείς ενδιαφέροντος. Τα πλεονεκτήματα του community cloud είναι η προσομοίωση ενός public cloud στις υποδομές εταιριών με τις ίδιες πολιτικές ασφάλειας και η κατανομή του κόστους στις εταιρίες αυτές.

2.3.4 Hybrid Cloud

Hybrid cloud ονομάζεται το cloud που αποτελεί σύνθεση 2 ή περισσοτέρων ειδών (private, public ή community) τα οποία αν και παραμένουν ξεχωριστές οντότητες, έχουν συνδεθεί με τέτοιον τρόπο ώστε να επιτρέπουν την μεταφορά δεδομένων και εφαρμογών από το ένα είδος στο άλλο [4] [5] [6] [7] [8].

3. Υλοποίηση

Η υλοποίηση της εργασίας έχει τη μορφή Proof of Concept καθώς δεν επαρκούσαν οι διαθέσιμοι πόροι για την υλοποίηση ενός περιβάλλοντος cloud computing. Για να είναι περισσότερο ρεαλιστικά τα αποτελέσματα έγινε port scanning στο cloud Okeanos της GRNET [9]. Το cloud Okeanos είναι ένα IaaS όπου μπορεί οποιοσδήποτε από το ακαδημαϊκό προσωπικό να εγκαταστήσει Virtual Machines διαφόρων λειτουργικών συστημάτων και να τα διαχειρίζεται μέσω του Web Interface ή μέσω σύνδεσης SSH. Η υλοποίηση της υποδομής έγινε σε ένα laptop όπου εγκαταστάθηκαν VMs διαφόρων λειτουργικών μέσω του VMware Workstation. Πιο συγκεκριμένα, χρησιμοποιήθηκαν Ubuntu 14.04 για την εγκατάσταση του IDS και την εγκατάσταση των servers, Windows 7 64 bit και Kali 1 & 2 για την υλοποίηση των επιθέσεων. Το IDS που χρησιμοποιήθηκε ήταν το Snort στην έκδοση 2.9.8.0 με τα snort rules στην 2.9.7.6. Για ευκολότερη ανάγνωση των alerts του Snort χρησιμοποιήθηκε το γραφικό περιβάλλον του Aanval στην έκδοση 8.

3.1 Snort

Το Snort είναι ένα δωρεάν Network Intrusion Detection System και Network Intrusion Prevention System ανοιχτού κώδικα. Δημιουργήθηκε το 1998 από τον Martin Roesch που ήταν και ο ιδρυτής της Sourcefire στην οποία άνηκε το Snort μέχρι που εξαγοράστηκε το 2013 από την Cisco. Υποστηρίζει διάφορα λειτουργικά συστήματα:

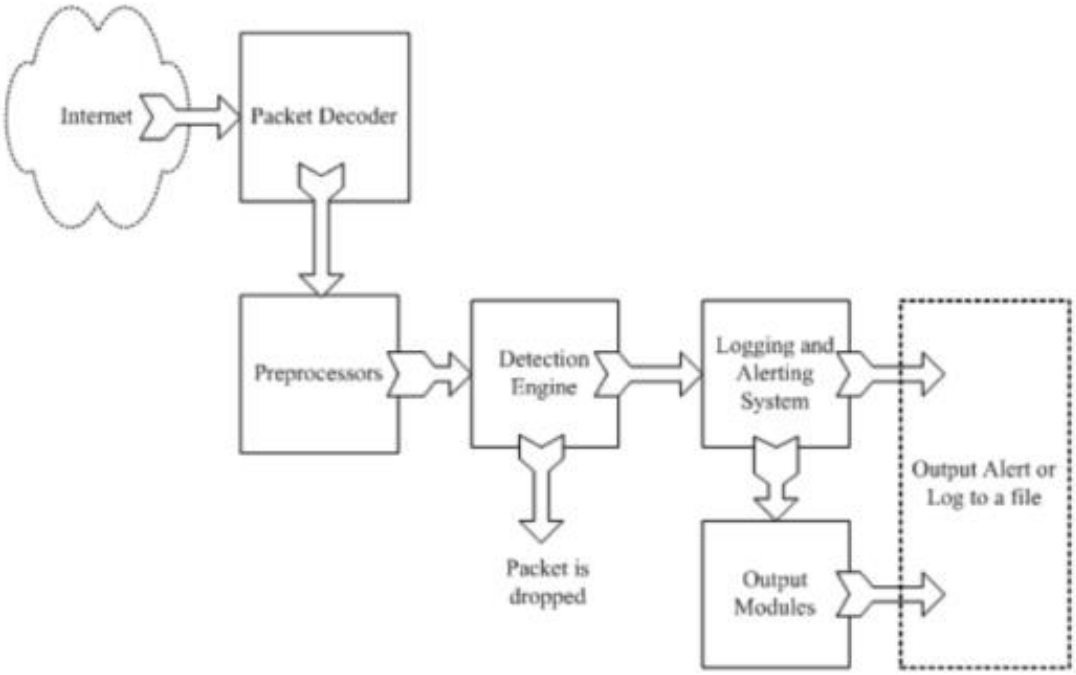
- Linux
- FreeBSD
- OpenBSD
- NetBSD
- Solaris (Sparc και i386)
- HP-UX
- AIX
- IRIX
- MacOS
- Windows

[10] [11] [12]

3.2 Αρχιτεκτονική Snort

Η αρχιτεκτονική του Snort αποτελείται από τα εξής βασικά στοιχεία:

- Packet decoder
- Preprocessors
- Detection engine
- Logging and alerting system
- Output modules



Εικόνα 2. Αρχιτεκτονική Snort

3.2.1 Packet Decoder

Ο packet decoder μαζεύει τα πακέτα από τα δικτυακά interfaces του μηχανήματος που είναι εγκατεστημένο το Snort μέσω της βιβλιοθήκης libpcap και τα προετοιμάζει για να περάσουν από τους preprocessors. Τα υποστηριζόμενα interfaces μπορούν να είναι Ethernet, SLIP, PPP και άλλα.

3.2.2 Preprocessors

Οι preprocessors είναι τα στοιχεία του snort που ταξινομούν τα πακέτα πριν κριθούν ως κακόβουλα από την detection engine. Ορισμένοι preprocessors παράγουν alerts όταν εντοπίζουν κακόβουλα πακέτα από τις τροποποιήσεις που έχουν γίνει στα headers τους. Επιπροσθέτως, οι preprocessors χρησιμοποιούνται για defragmentation των πακέτων ούτως ώστε να είναι εφικτό να διαβαστεί ολόκληρο το πακέτο για να κριθεί αν είναι κακόβουλο ή όχι.

3.2.3 Detection Engine

Η detection engine αποτελεί το σημαντικότερο στοιχείο του Snort καθώς κρίνει εάν υπάρχει ύποπτη δραστηριότητα σε ένα πακέτο. Χρησιμοποιεί τα rules του Snort, για να ελέγξει αν ταιριάζουν με κάποιο πακέτο. Σε περίπτωση που ταιριάζει κάποιο rule με ένα πακέτο τότε είτε εμφανίζεται alert, είτε το πακέτο αποθηκεύεται στα logs. Αν δεν ταιριάζει με κάποιο rule τότε το πακέτο γίνεται drop. Λόγω της ύπαρξης μεγάλου όγκου πακέτων είναι σημαντικό να υπάρχουν αρκετοί διαθέσιμοι υπολογιστικοί πόροι έτσι ώστε να είναι μπορεί η detection engine να

επεξεργάζεται το σύνολο του όγκου των πακέτων. Οι παράγοντες που επηρεάζουν τον φόρτο της είναι οι ακόλουθοι:

- Ο αριθμός των rules
- Η ισχύς της CPU του μηχανήματος που είναι εγκατεστημένο το Snort
- Η ταχύτητα του εσωτερικού διαύλου bus που χρησιμοποιείται από το Snort
- Ο φόρτος του δικτύου

Επιπλέον, η detection engine μπορεί να διαχωρίσει ένα πακέτο σε μέρη για να μπορέσει να το επεξεργαστεί. Τα μέρη αυτά είναι:

- Το IP header του πακέτου
- Το transport layer header όπως είναι τα TCP, UDP κλπ.
- Το application layer header όπως είναι τα DNS, FTP, SNMP.
- Το payload του πακέτου δηλαδή κάποιο αλφαριθμητικό που βρίσκεται μέσα στα δεδομένα του πακέτου.

3.2.4 Logging and Alerting System

Τα logs και τα alerts του Snort συγκεντρώνονται στο directory /var/log/snort σε όποια μορφή έχει επιλέξει ο χρήστης του snort. Οι διαθέσιμες μορφές είναι: syslog, pcap, unified2.

3.2.5 Output Modules

Τα output modules ελέγχουν τι output βγαίνει από το logging and alerting system και έτσι ανάλογα με την παραμετροποίηση που έχει γίνει στο Snort μπορούν να έχουν τα επόμενα αποτελέσματα:

- Απλή καταγραφή των logs είτε στο default directory, είτε σε οποιοδήποτε άλλο.
- Αποστολή SNMP traps.
- Αποστολή των logs σε syslog server.
- Αποστολή των logs σε βάση δεδομένων MySQL ή Oracle.
- Δημιουργία XML αρχείου.
- Αλλαγή παραμετροποίησης σε routers και firewalls.
- Αποστολή Server Message Block (SMB) μηνυμάτων σε μηχανήματα Microsoft Windows. [10]

4. Επιθέσεις

Το αρχικό στάδιο μιας επίθεσης αποτελείται από την αναγνώριση του δικτύου (reconnaissance). Στο cloud Okeanos χρησιμοποιήθηκε το εργαλείο nmap (περιγράφεται παρακάτω) για να ανακαλυφθούν τα υποψήφια θύματα. Επισυνάπτεται το σχετικό αρχείο με τα αποτελέσματα.

Okeanos_nmap_results

Όπως φαίνεται υπάρχουν διάφορα Virtual Machines με ανοιχτές ports, όπως 21, 22, 25, 80, 443, 8080 καθώς και πολλές άλλες. Για να προσομοιωθούν οι διαφορετικού τύπου επιθέσεις δημιουργήθηκαν locally Virtual Machines με διάφορες ports ανοιχτές.

4.1 Επιθέσεις Αναγνώρισης

4.1.1 Εργαλείο nmap

Το nmap είναι ένα εργαλείο που χρησιμοποιείται για network scanning και δίνει τη δυνατότητα στον χρήστη, να σαρώσει μεγάλα δίκτυα σε λίγο χρόνο. Τα αποτελέσματα που δίνει περιέχουν πληροφορίες σχετικά με το λειτουργικό ενός network element ή server, την εφαρμογή που τρέχει και τις port που είναι ανοιχτές [13].

Στο configuration file του snort έγιναν uncomment οι γραμμές:

preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { low }

include $PREPROC_RULE_PATH/preprocessor.rules

για να αναγνωριστούν τα portscans.

Έτσι εκτελώντας την εντολή:

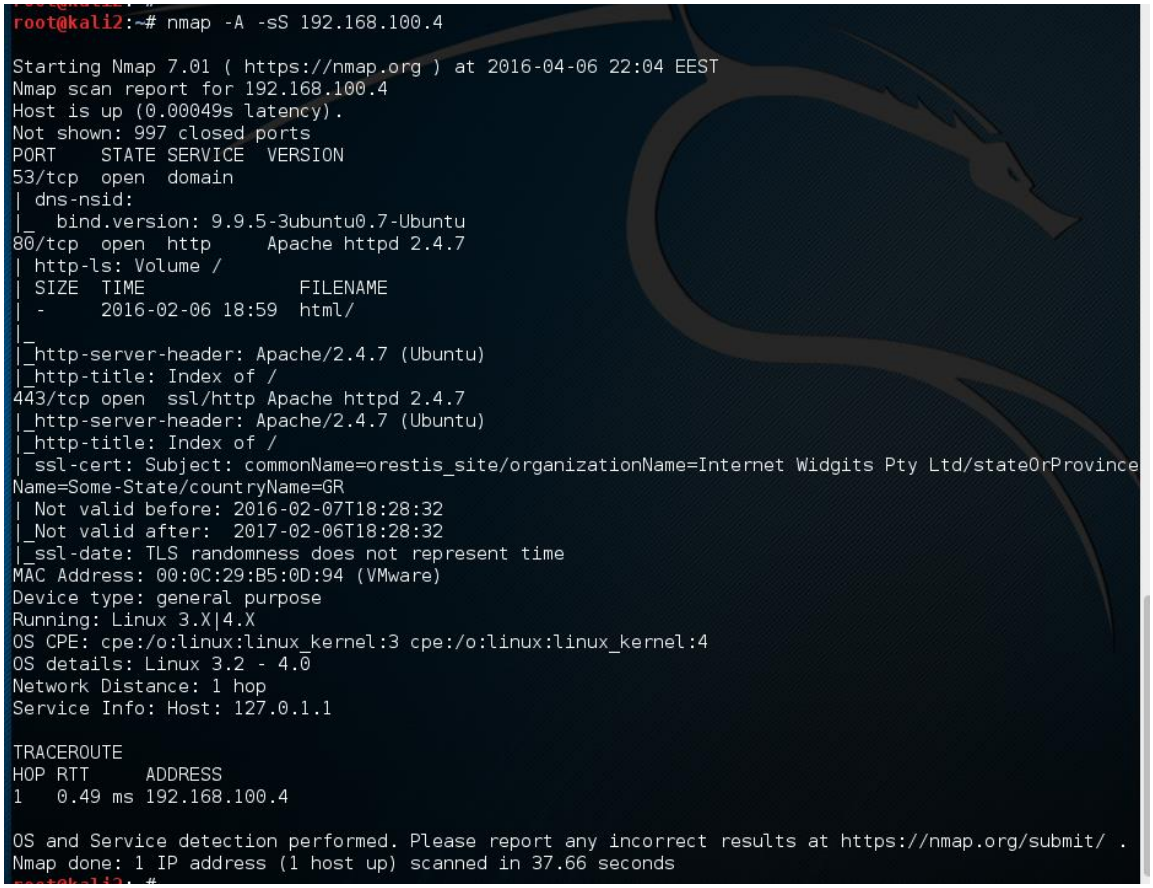
nmap -A -sS 192.168.100.4

όπου:

-A: αναγνωρίζει το λειτουργικό και την έκδοσή του

-sS: κάνει TCP SYN scan για μεγαλύτερη ταχύτητα στο port scanning

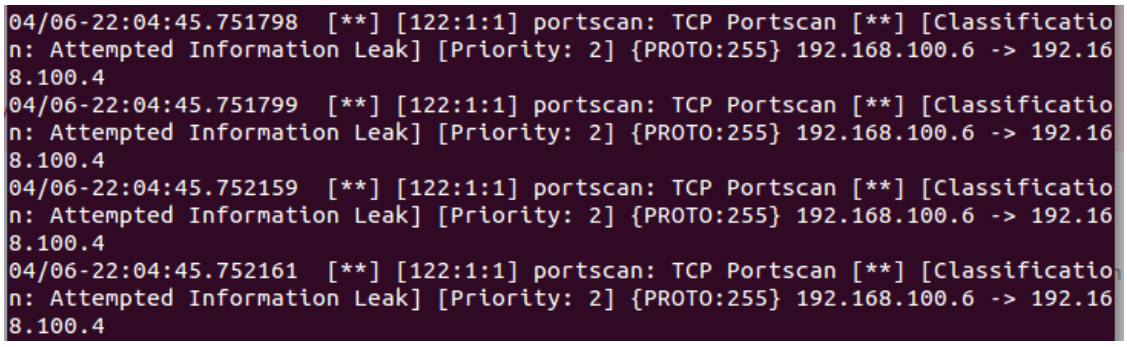
προκύπτουν τα παρακάτω αποτελέσματα στο Kali.



Εικόνα 3. Αποτελέσματα nmap -A -sS

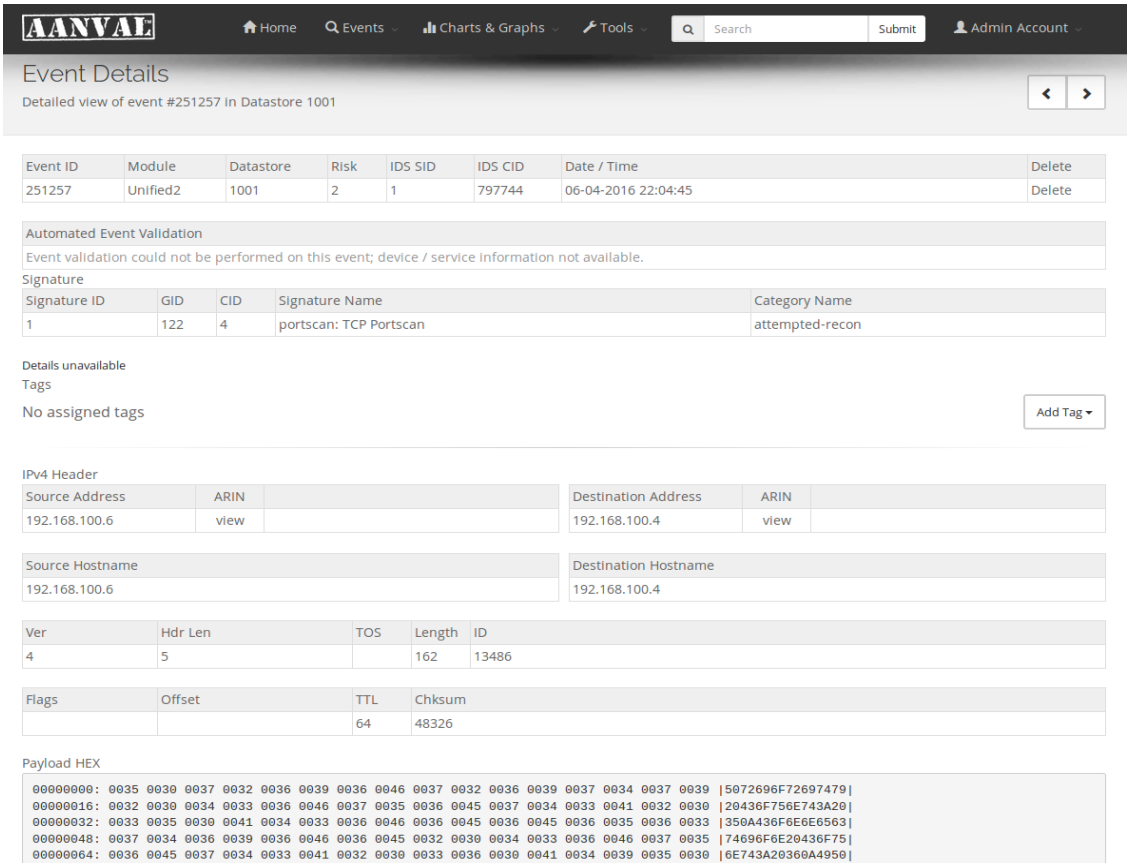
Οι ports που είναι ανοιχτές είναι οι 53, 80 & 443, που σημαίνει ότι πρόκειται για DNS server που είναι παράλληλα και Web Server για http και https. Φανερές είναι και οι εκδόσεις των εφαρμογών Bind (DNS) και Apache, οι ημερομηνίες του πιστοποιητικού του Web server, η MAC address καθώς και το λειτουργικό και η έκδοσή του (Ubuntu με Linux kernel 3.2 - 4.0).

Το snort εμφάνισε τα παρακάτω alerts.



Εικόνα 4. Snort alerts για το nmap -A -sS

Στο aanval εμφανίζεται το alert *TCP Portscan* με περισσότερες πληροφορίες.



Εικόνα 5. Aanval alert για το nmap -A -sS

Εκτελώντας την εντολή:

nmap -sA 192.168.100.4

όπου:

-sA: TCP ACK scan

δεν εμφανίζονται αποτελέσματα στο Kali καθώς το συγκεκριμένο scan εμφανίζει μόνο ότι οι ports είναι unfilterted χωρίς να ξεχωρίζει αν είναι ανοιχτές ή κλειστές. Επίσης, το snort δεν εμφανίζει κάποιο alert.

Παρόμοια αποτελέσματα έχουμε με τα -sY, -sW & -sM.

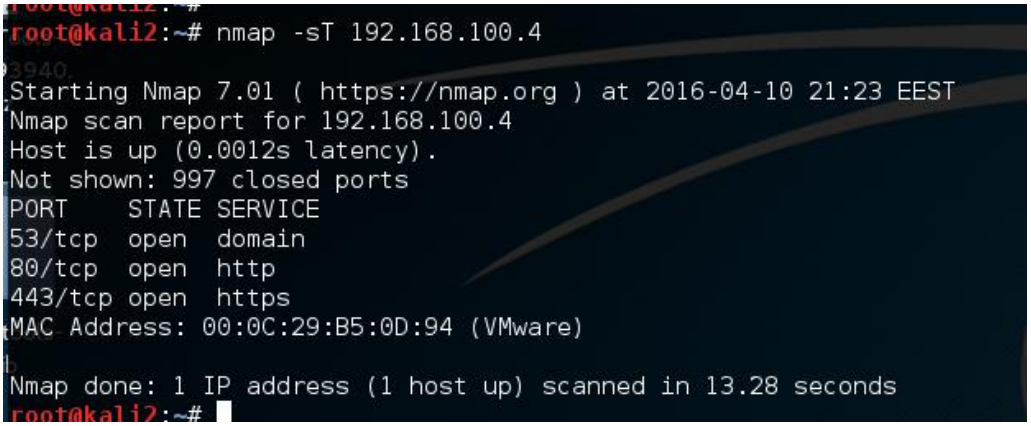
Εκτελώντας την εντολή:

nmap -sT 192.168.100.4

όπου:

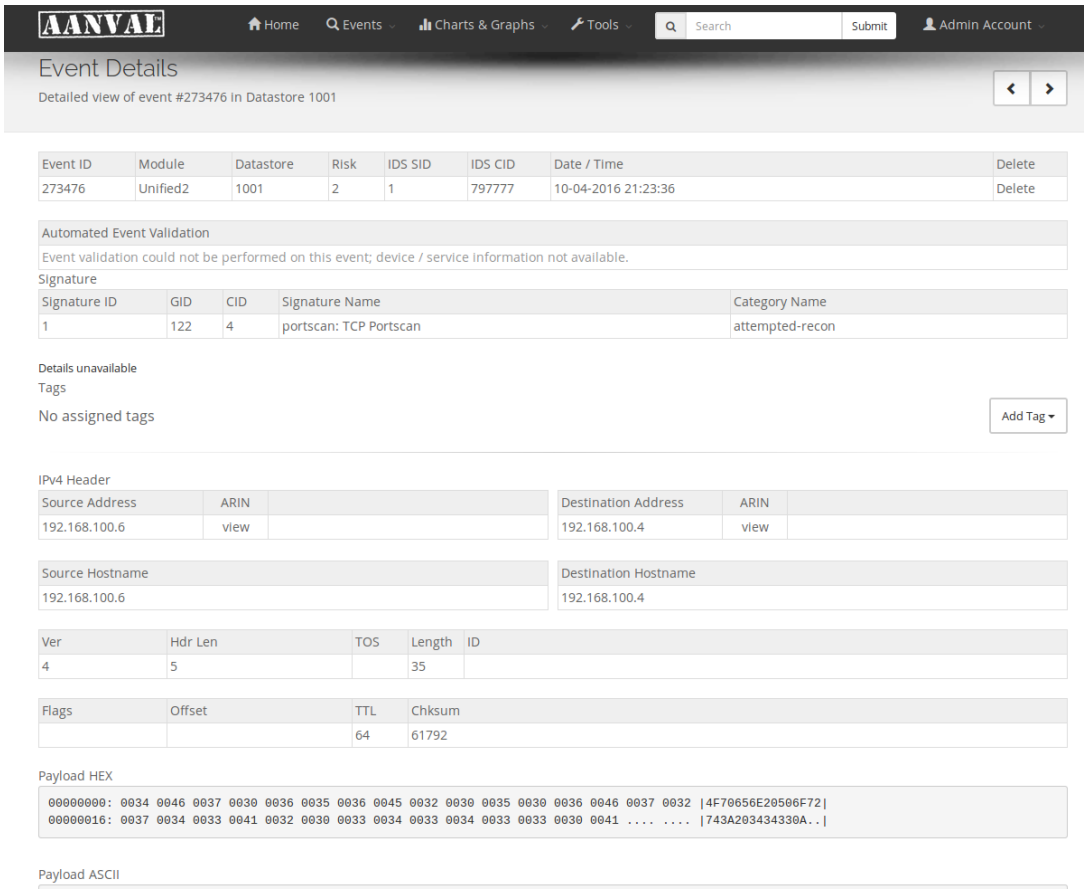
-sT: TCP Connect scan

εμφανίζονται οι ανοιχτές ports στο Kali.



Εικόνα 6. Αποτελέσματα nmap -sT

Και το alert *TCP Portscan* στο web interface του snort το Aanval.



Εικόνα 7. Aanval alert για το nmap -sT

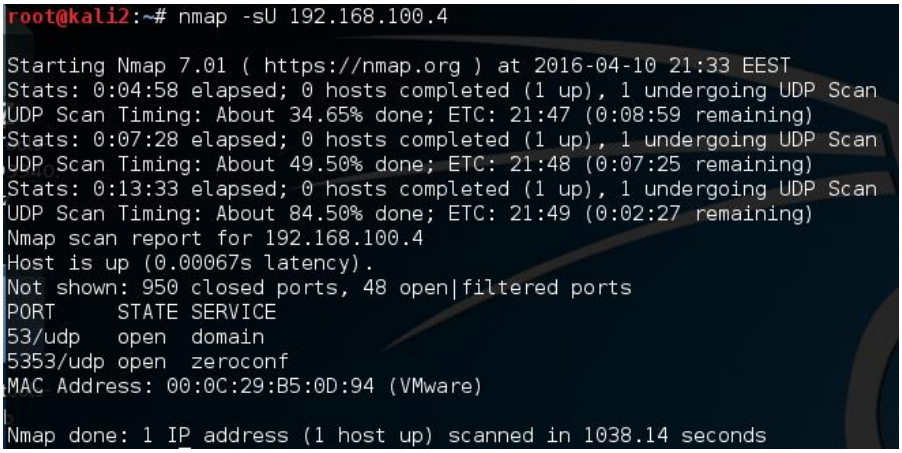
Εκτελώντας την εντολή:

nmap -sU 192.168.100.4

όπου:

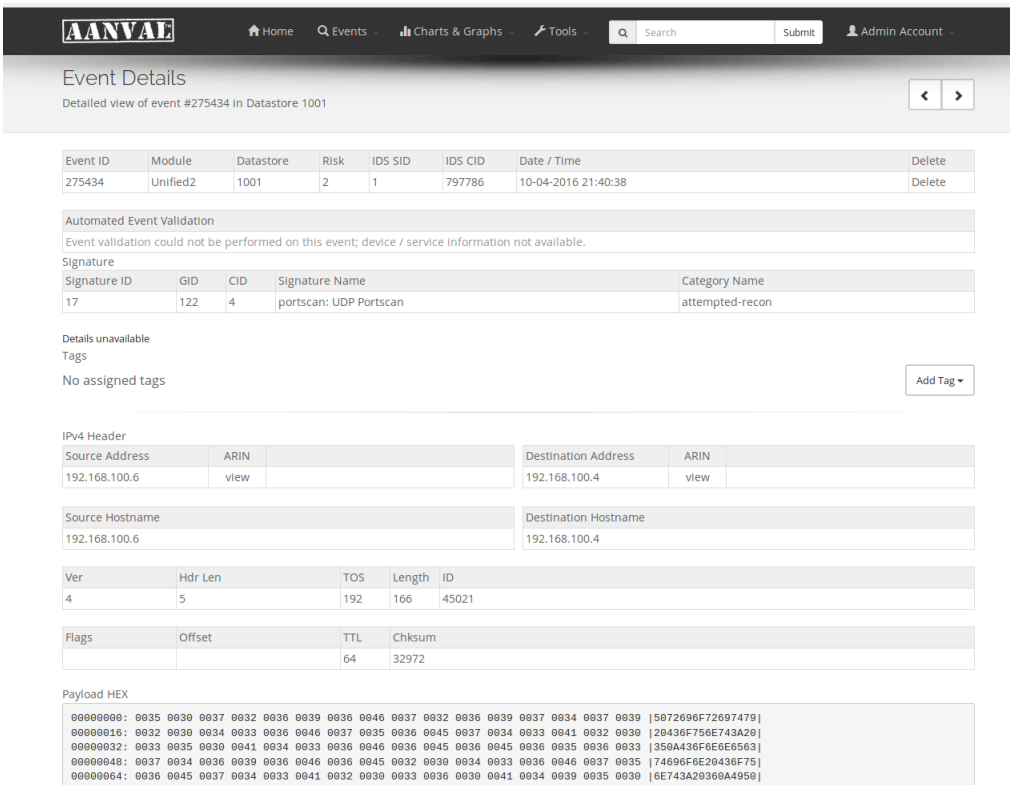
-sT: UDP scan

εμφανίζονται οι ανοιχτές ports στο Kali, αλλά το συγκεκριμένο scan απαιτεί παραπάνω χρόνο.



Εικόνα 8. Αποτελέσματα nmap -sU

Και το alert *UDP Portscan* στο Aanval.



Εικόνα 9. Aanval alert για το nmap -sU

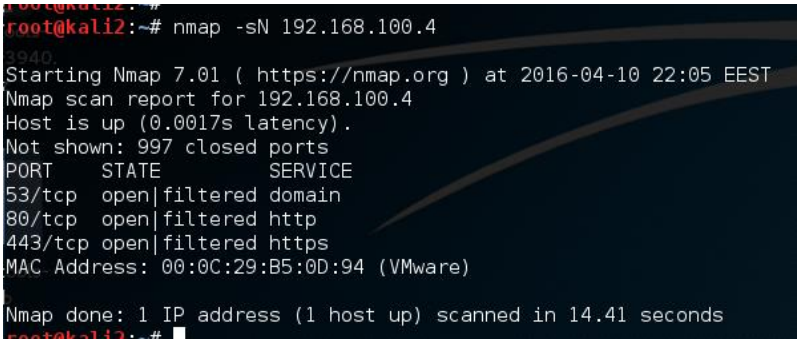
Εκτελώντας την εντολή:

nmap -sN 192.168.100.4

όπου:

-sN: Null Scan (To TCP flag header είναι 0)

εμφανίζονται οι ανοιχτές ή φιλτραρισμένες ports στο Kali, ενώ το snort δεν αναγνώρισε το scan.



Εικόνα 10. Αποτελέσματα nmap -sN

Τα ίδια αποτελέσματα εμφανίστηκαν και με τα -sF (TCP FIN bit), -sX (Xmas scan: FIN, PSH & URG flags).

Εκτελώντας την εντολή:

nmap -sZ 192.168.100.4

όπου:

-sZ: SCTP Cookie ECHO scan

εμφανίζονται φιλτραρισμένες άλλες ports στο Kali, ενώ το snort δεν αναγνώρισε το scan.

Εικόνα 11. Αποτελέσματα nmap -sZ

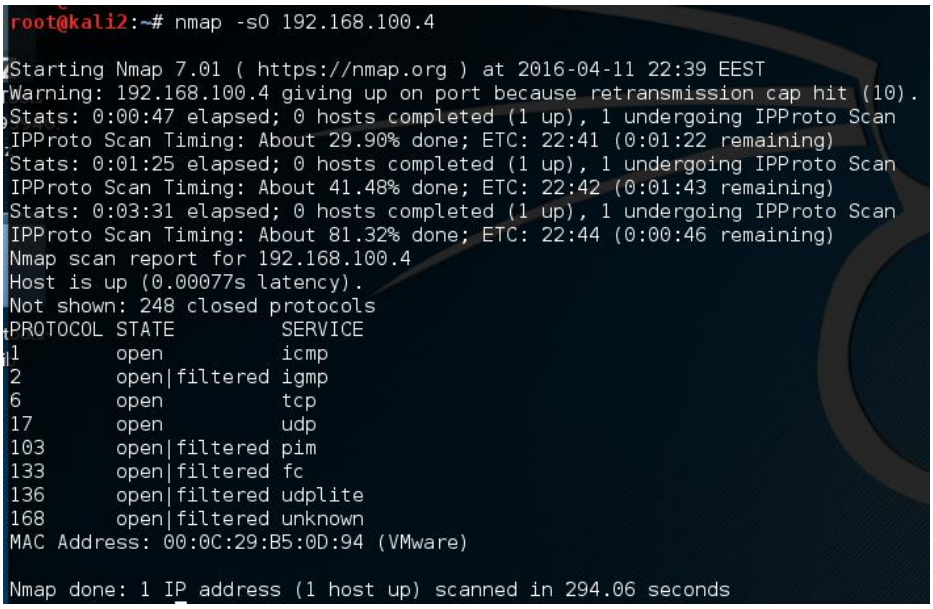
Εκτελώντας την εντολή:

nmap -sO 192.168.100.4

όπου:

-sO: IP protocol scan

εμφανίζονται ανοιχτές και φιλτραρισμένες ports που δεν χρησιμοποιούνται από κάποιο πρωτόκολλο. Το Aanval επίσης δεν εμφάνισε τίποτα.



Εικόνα 12. Αποτελέσματα nmap -sO

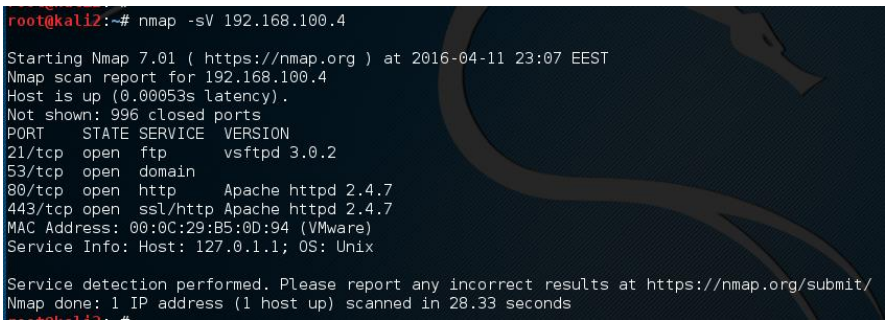
Εκτελώντας την εντολή:

nmap -sV 192.168.100.4

όπου:

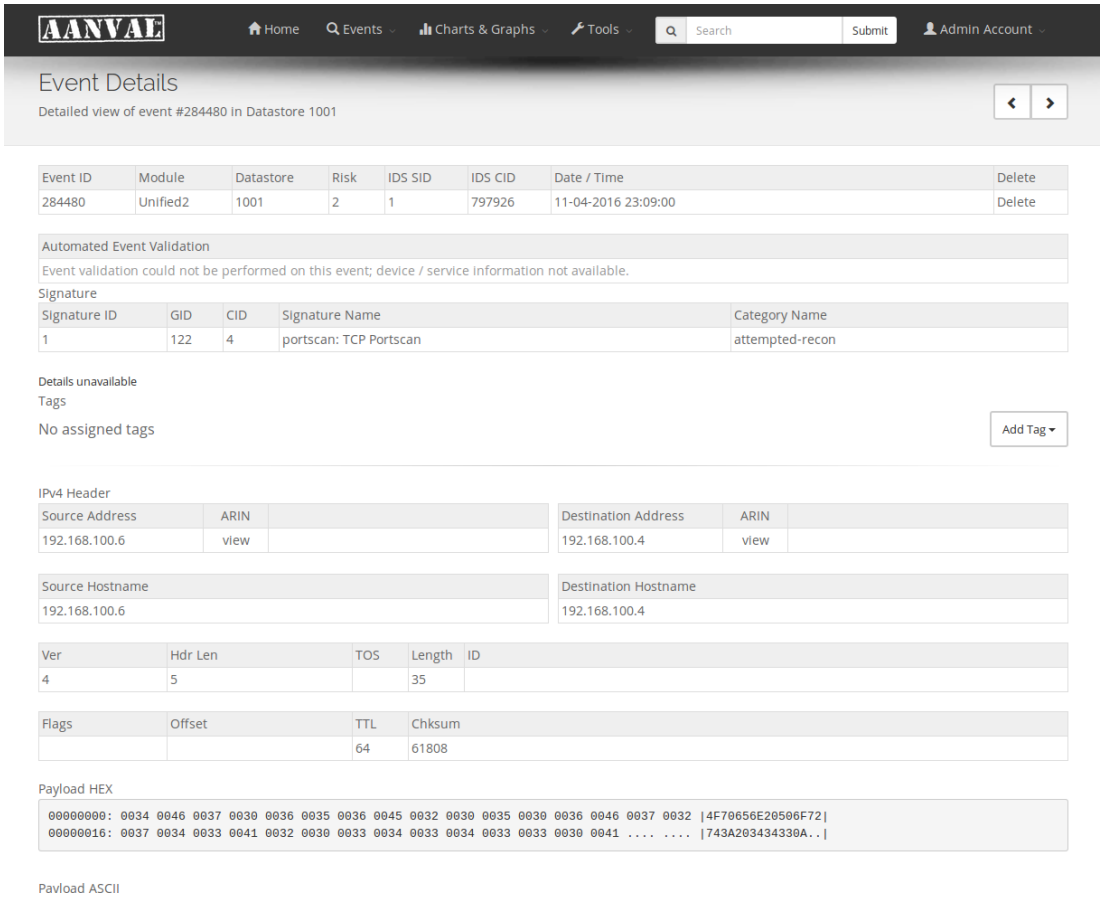
-sV: Version detection

εμφανίζονται οι ανοιχτές ports μαζί με τις εκδόσεις των εφαρμογών που τρέχουν σ' αυτές τις ports.



Εικόνα 13. Αποτελέσματα nmap -sV

Το Aanval εμφάνισε το TCP Portscan.

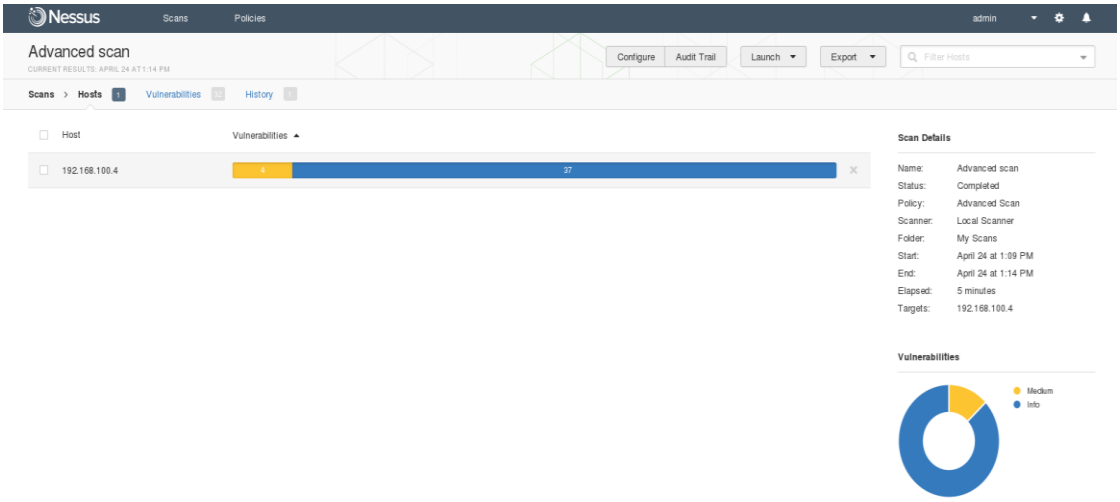


Εικόνα 14. Aanval alert για το nmap –sV

4.1.2 Nessus

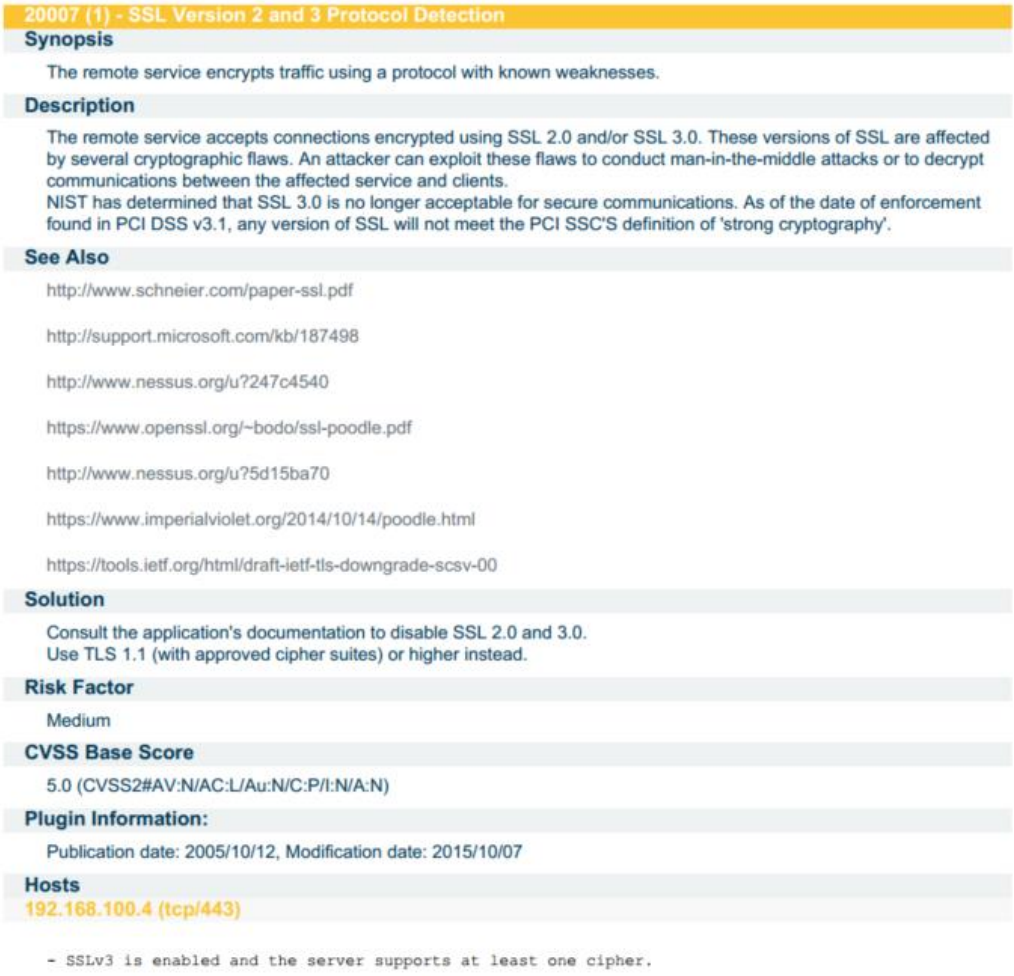
Το Nessus αποτελεί ένα εργαλείο για vulnerability scanning. Δημιουργήθηκε το 1998 από τον Renaud Deraison ως δωρεάν security scanner και από το 2005 η εταιρία Tenable Network Security το παρέχει μέσω license. Παρέχεται είτε μέσω συνδρομής, είτε δωρεάν για περιορισμένο αριθμό IP διευθύνσεων και χρησιμοποιεί έναν μεγάλο αριθμό από plugins για την αναζήτηση των ευπαθειών [14].

Τρέχοντας το *Advanced Scan* και το *Web Application Scan* το Nessus εμφάνισε ένα αριθμό από vulnerabilities που ήταν κυρίως medium risk και informational. Τα medium αφορούσαν τις εκδόσεις του SSL και τα directories του Web Server. Το Aanval εμφάνισε και αυτό κάποια alerts, κυρίως για port scanning. Παρακάτω φαίνονται τα αποτελέσματα του *Advanced Scan*.



Εικόνα 15. Nessus Advanced Scan

Εικόνα 16. Αποτελέσματα του Advanced Scan



20007 (1) - SSL Version 2 and 3 Protocol Detection

Synopsis

The remote service encrypts traffic using a protocol with known weaknesses.

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws. An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.
NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC'S definition of 'strong cryptography'.

See Also

http://www.schneier.com/paper-ssl.pdf

http://support.microsoft.com/kb/187498

http://www.nessus.org/u?247c4540

https://www.openssl.org/~bodo/ssl-poodle.pdf

http://www.nessus.org/u?5d15ba70

https://www.imperialviolet.org/2014/10/14/poodle.html

https://tools.ietf.org/html/draft-ietf-tls-downgrade-scsv-00

Solution

Consult the application's documentation to disable SSL 2.0 and 3.0.
Use TLS 1.1 (with approved cipher suites) or higher instead.

Risk Factor

Medium

CVSS Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

Plugin Information:

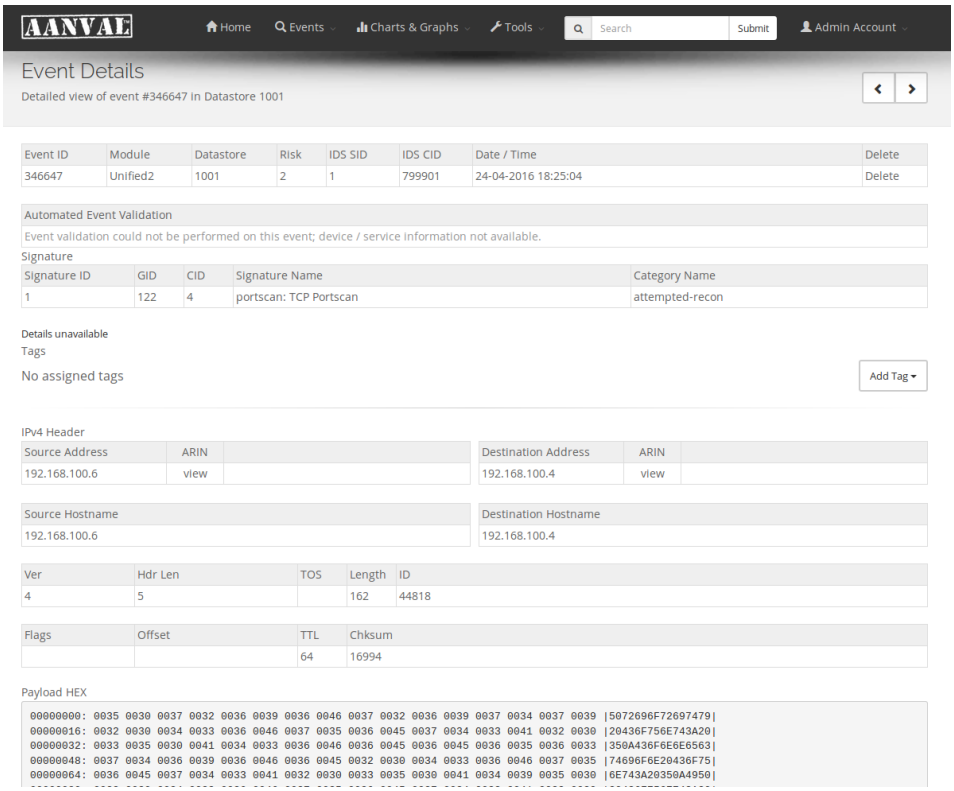
Publication date: 2005/10/12, Modification date: 2015/10/07

Hosts

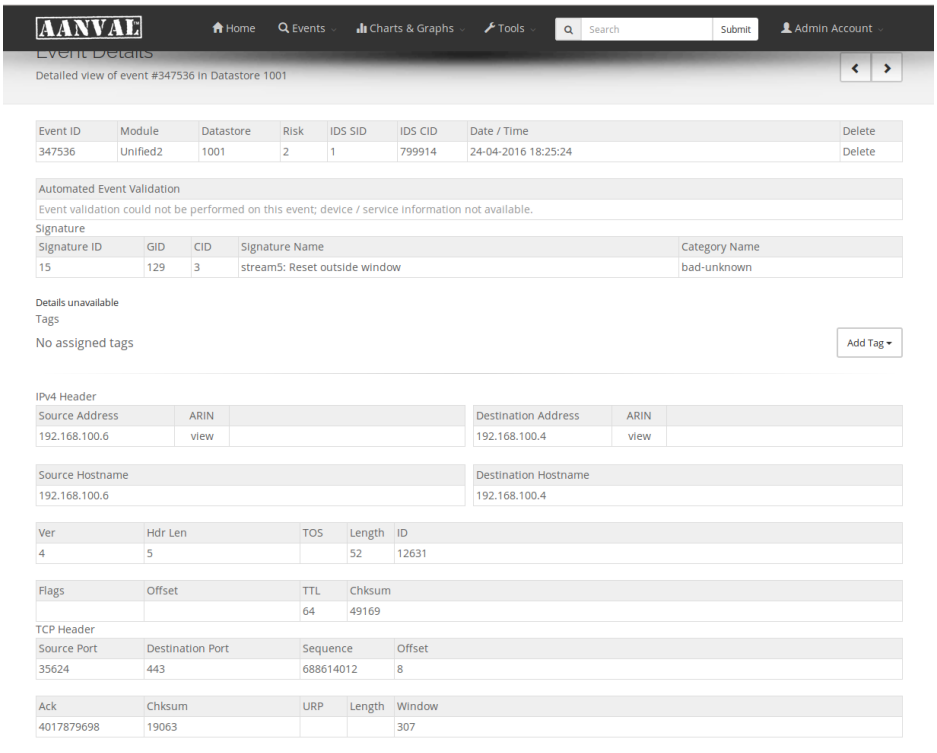
192.168.100.4 (tcp/443)

```
- SSLv3 is enabled and the server supports at least one cipher.
```

Εικόνα 17. SSL version detection από το Advanced Scan του Nessus



Εικόνα 18. Aanval alert για το Advanced Scan του Nessus



Εικόνα 19. Aanval alert για το Advanced Scan του Nessus

Περισσότερα αποτελέσματα από τα scan του Nessus υπάρχουν στα επισυναπτόμενα αρχεία.

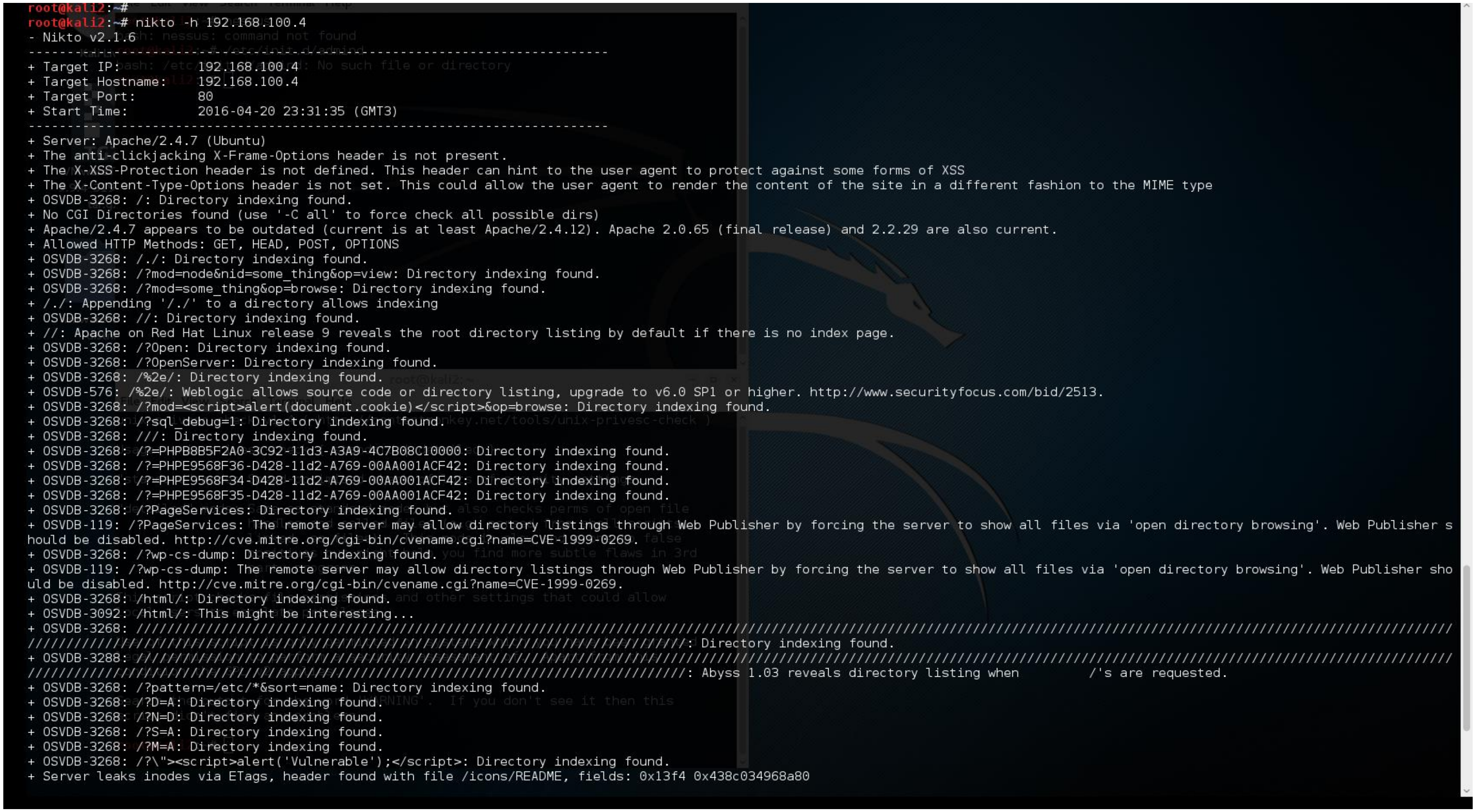

Advanced_scan.pdf


Web_app_scan.pdf

4.1.3 Nikto

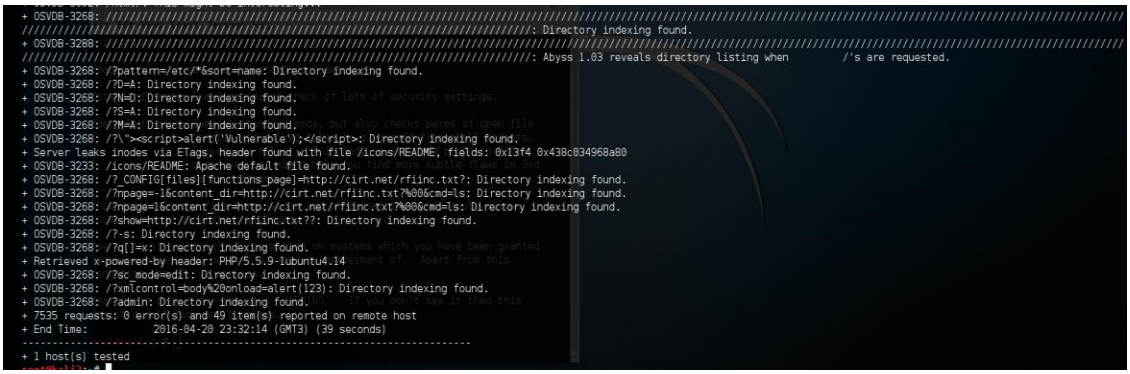
Το Nikto είναι ένας web server scanner που περιλαμβάνει διαφόρων ειδών tests και κάνει ελέγχους για outdated server versions σε πάνω από 1250 servers. Επιπλέον, έχει τη δυνατότητα να πραγματοποιήσει ελέγχους σε configuration files όπως index files, HTTP server options κλπ.

Το Nikto εμφάνισε διάφορα αποτελέσματα όπως φαίνονται και παρακάτω.



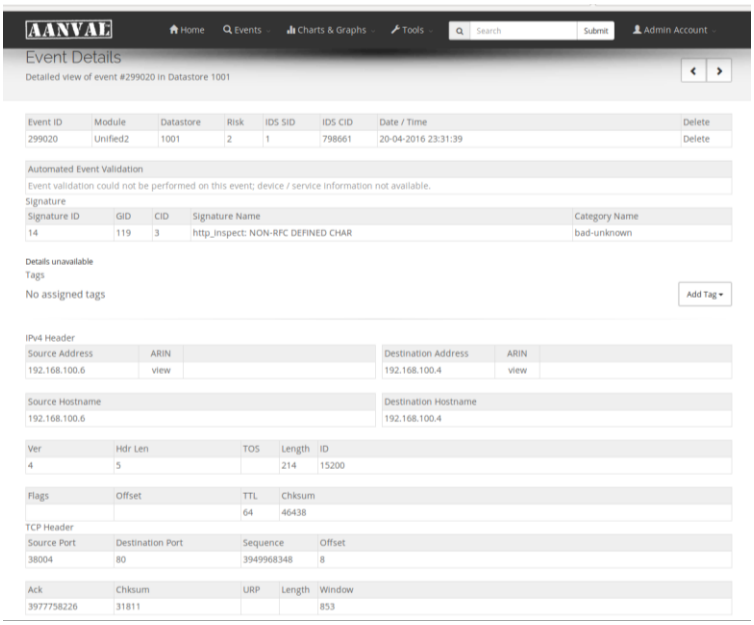
```
root@kali2:~#
root@kali2:~# nikto -h 192.168.100.4
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.100.4
+ Target Hostname:    192.168.100.4
+ Target Port:        80
+ Start Time:         2016-04-20 23:31:35 (GMT3)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ 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
+ OSVDB-3268: /: Directory indexing found.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /./: Directory indexing found.
+ OSVDB-3268: /?mod=node&nid=some_thing&op=view: Directory indexing found.
+ OSVDB-3268: /?mod=some_thing&op=browse: Directory indexing found.
+ /./: Appending '/./' to a directory allows indexing
+ OSVDB-3268: //: Directory indexing found.
+ //: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ OSVDB-3268: /?Open: Directory indexing found.
+ OSVDB-3268: /?OpenServer: Directory indexing found.
+ OSVDB-3268: /%2e/: Directory indexing found.
+ OSVDB-576: /%2e/: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. http://www.securityfocus.com/bid/2513.
+ OSVDB-3268: /?mod=<script>alert(document.cookie)</script>&op=browse: Directory indexing found.
+ OSVDB-3268: /?sql_debug=1: Directory indexing found.
+ OSVDB-3268: ///: Directory indexing found.
+ OSVDB-3268: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: Directory indexing found.
+ OSVDB-3268: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: Directory indexing found.
+ OSVDB-3268: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: Directory indexing found.
+ OSVDB-3268: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: Directory indexing found.
+ OSVDB-3268: /?PageServices: Directory indexing found.
+ OSVDB-119: /?PageServices: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher s
hould be disabled. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269.
+ OSVDB-3268: /?wp-cs-dump: Directory indexing found.
+ OSVDB-119: /?wp-cs-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher sho
uld be disabled. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269.
+ OSVDB-3268: /html/: Directory indexing found.
+ OSVDB-3092: /html/: This might be interesting...
+ OSVDB-3268: ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////: Directory indexing found.
+ OSVDB-3288: ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////: Abyss 1.03 reveals directory listing when         /'s are requested.
+ OSVDB-3268: /?pattern=/etc/*&sort=name: Directory indexing found.
+ OSVDB-3268: /?D=A: Directory indexing found.
+ OSVDB-3268: /?N=D: Directory indexing found.
+ OSVDB-3268: /?S=A: Directory indexing found.
+ OSVDB-3268: /?M=A: Directory indexing found.
+ OSVDB-3268: /?\"><script>alert('Vulnerable');</script>: Directory indexing found.
+ Server leaks inodes via ETags, header found with file /icons/README, fields: 0x13f4 0x438c034968a80
```

Εικόνα 20. Αποτελέσματα Nikto

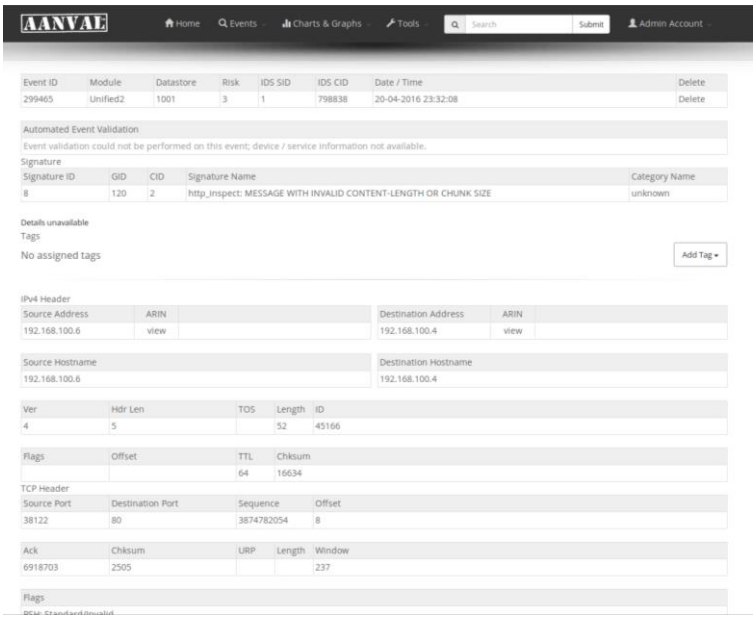


Εικόνα 21. Αποτελέσματα Nikto

Το Aanval εμφάνισε μόνο κάποια alerts HTTP inspect.



Εικόνα 22. Aanval alert από το Nikto



Εικόνα 23. Aanval alert από το Nikto

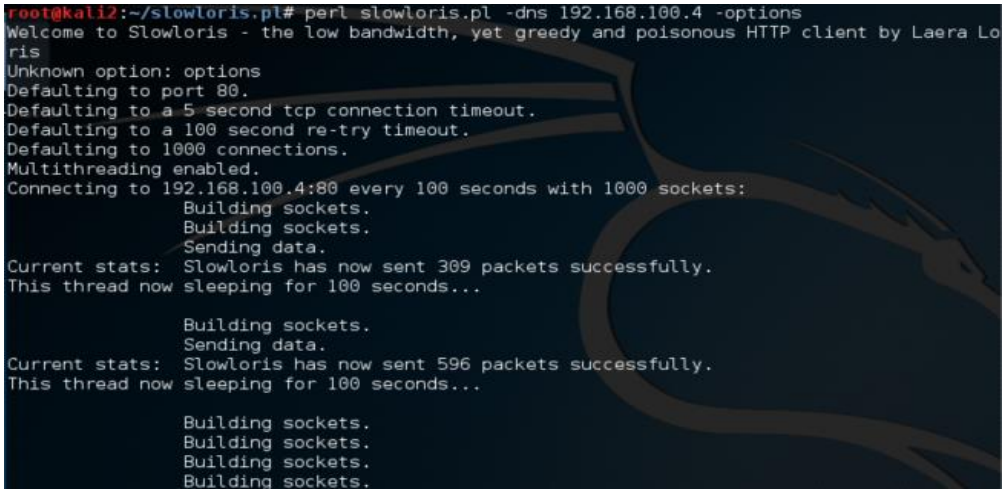
4.2 Επιθέσεις DoS

4.2.1 Επίθεση Slowloris

Η επίθεση slowloris αποτελεί επίθεση τύπου Denial of Service (DoS). Ο επιτιθέμενος έχει τη δυνατότητα μέσω ενός μόνο υπολογιστή και πολύ χαμηλού bandwidth να σταματήσει τη λειτουργία ενός web server. Η επίθεση επιτυγχάνεται μέσω της δημιουργίας πολλαπλών συνδέσεων προς τον server-θύμα οι οποίες παραμένουν ανοιχτές. Αυτό έχει ως αποτέλεσμα να μην μπορεί να εξυπηρετήσει τις νέες συνδέσεις από άλλους clients.

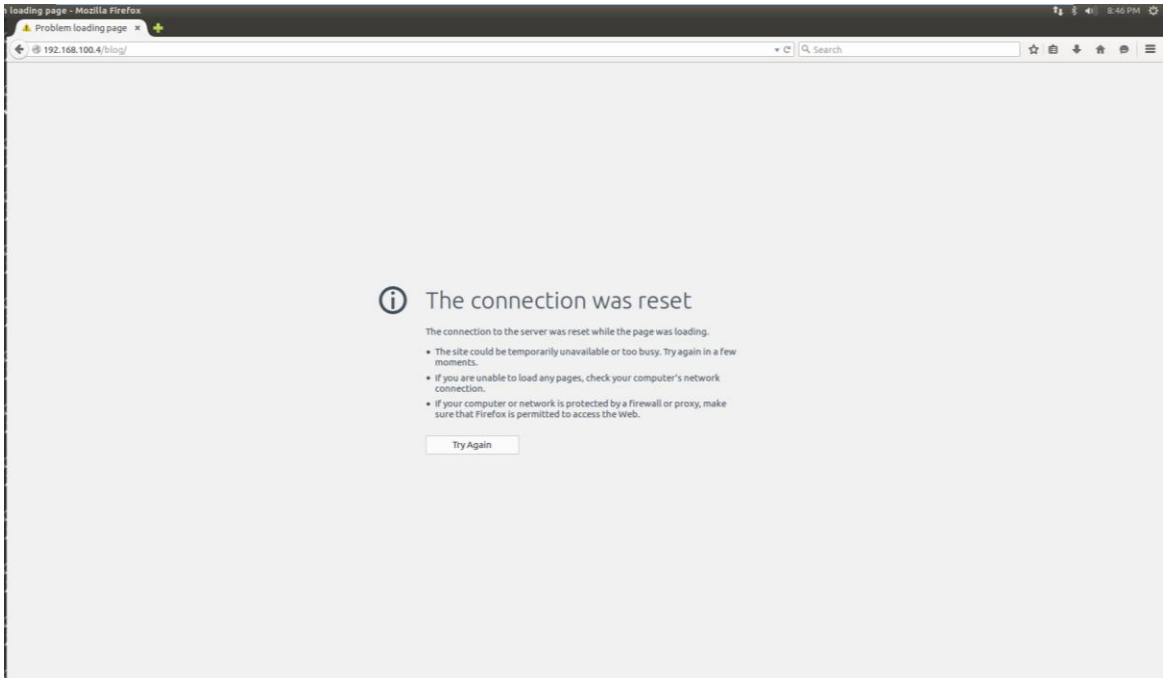
Η επίθεση υλοποιήθηκε με την εκτέλεση του perl script:

perl slowloris.pl -dns 192.168.100.4 -options



Εικόνα 24. Εκτέλεση επίθεσης slowloris

Η οποία και ανάγκασε τον Apache server να μην μπορεί να εξυπηρετήσει τους clients.



Εικόνα 25. Αδυναμία εξυπηρέτησης του client από τον Apache server.

Στην προκειμένη περίπτωση το Snort δεν αναγνώρισε την επίθεση, λόγω της έλλειψης αντίστοιχου rule. Επίσης, ούτε κάποιο άλλο DoS rule αναγνώρισε την συγκεκριμένη επίθεση.

4.2.2 Επίθεση Slowhttptest

Η επίθεση slowhttptest είναι επίθεση DoS που γίνεται στο Application Layer και πιο συγκεκριμένα στο πρωτόκολλο HTTP.

Εκτελέστηκε με την ακόλουθη εντολή:

slowhttptest -c 1000 -B -g -o my_body_stats -i 110 -r 200 -s 8192 -t FAKEVERB -u https://192.168.100.4 -x 10 -p 3

όπου:

-c: αριθμός των connections

-B: slow down in body section (όχι header section)

-g: τα στατιστικά βγαίνουν σε CSV & HTML αρχεία

-o: output file

-i: μεσοδιάστημα σε δευτερόλεπτα μεταξύ των δεδομένων που στέλνονται ανά connection

-r: connections ανά δευτερόλεπτο

-s: τιμή σε byte του Content-Length header

-t: HTTP verb (method, e.g. GET,PORT etc.)

-u: url

-x: max length των follow up data

-p: probe connection timeout, μετά το οποίο ο server είναι inaccessible

Εικόνα 26. Εκτέλεση επίθεσης slowhttptest

Σύμφωνα με το output file ο Apache server σταματάει να ανταποκρίνεται μετά από 6 δευτερόλεπτα αλλά στην πραγματικότητα ο client δεν μπορεί να φορτώσει τη σελίδα από την πρώτη στιγμή. Το snort για ακόμη μία φορά δεν μπορεί να αντιληφθεί την επίθεση.

4.2.3 Εργαλείο hping3

Το hping3 είναι ένα εργαλείο δικτύου που έχει τη δυνατότητα να αποστείλει πακέτα TCP/IP παρόμοια με αυτά που στέλνει το ping. Με το hping3 μπορεί να γίνει port scanning, να δοκιμαστούν firewall rules και να ελεγχθεί η απόδοση του δικτύου για διάφορα πρωτοκόλλα και για διαφορετικού μεγέθους πακέτα.

Δοκιμάστηκε η επόμενη εντολή:

hping3 -c 10000 -d 120 -S -w 64 -p 21 --flood --rand-source 192.168.100.4

όπου:

-c 10000: αριθμός των πακέτων που στέλνονται

-d 120: μέγεθος πακέτου σε byte

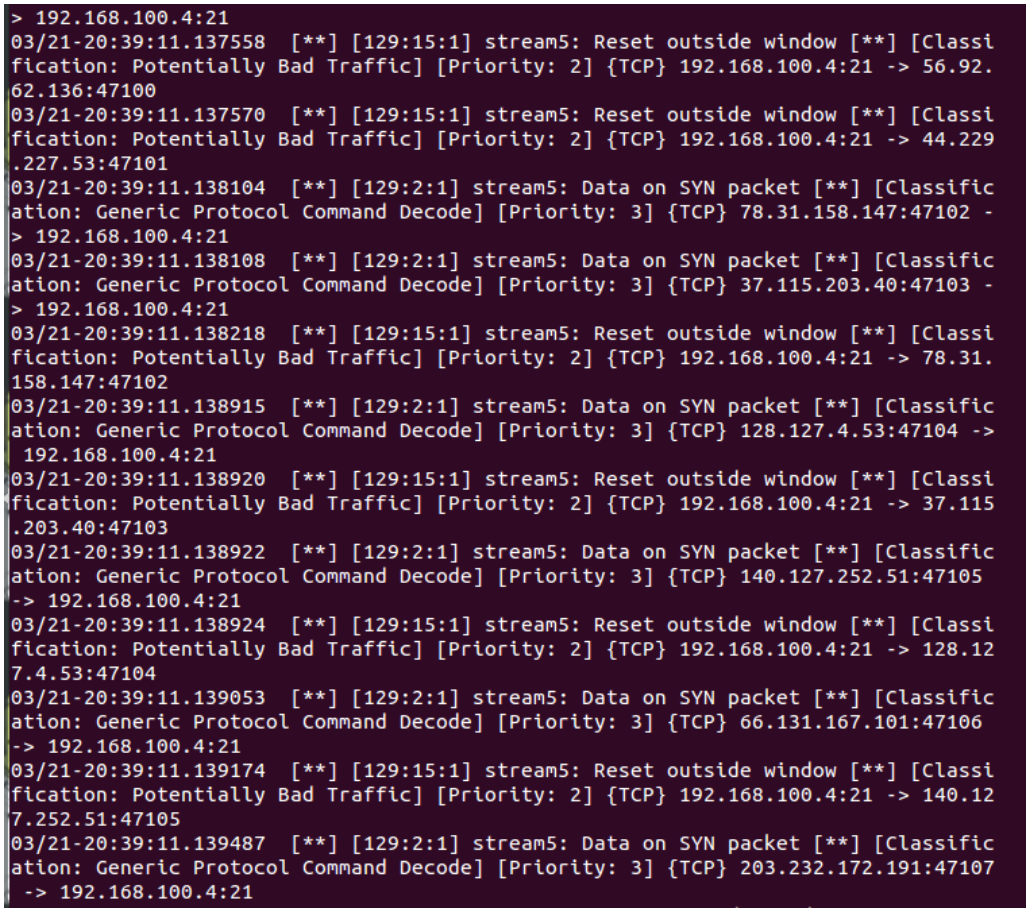
-S: αποστολή μόνο SYN πακέτων

-w 64: μέγεθος TCP window

-p 21: port του υπολογιστή στόχου

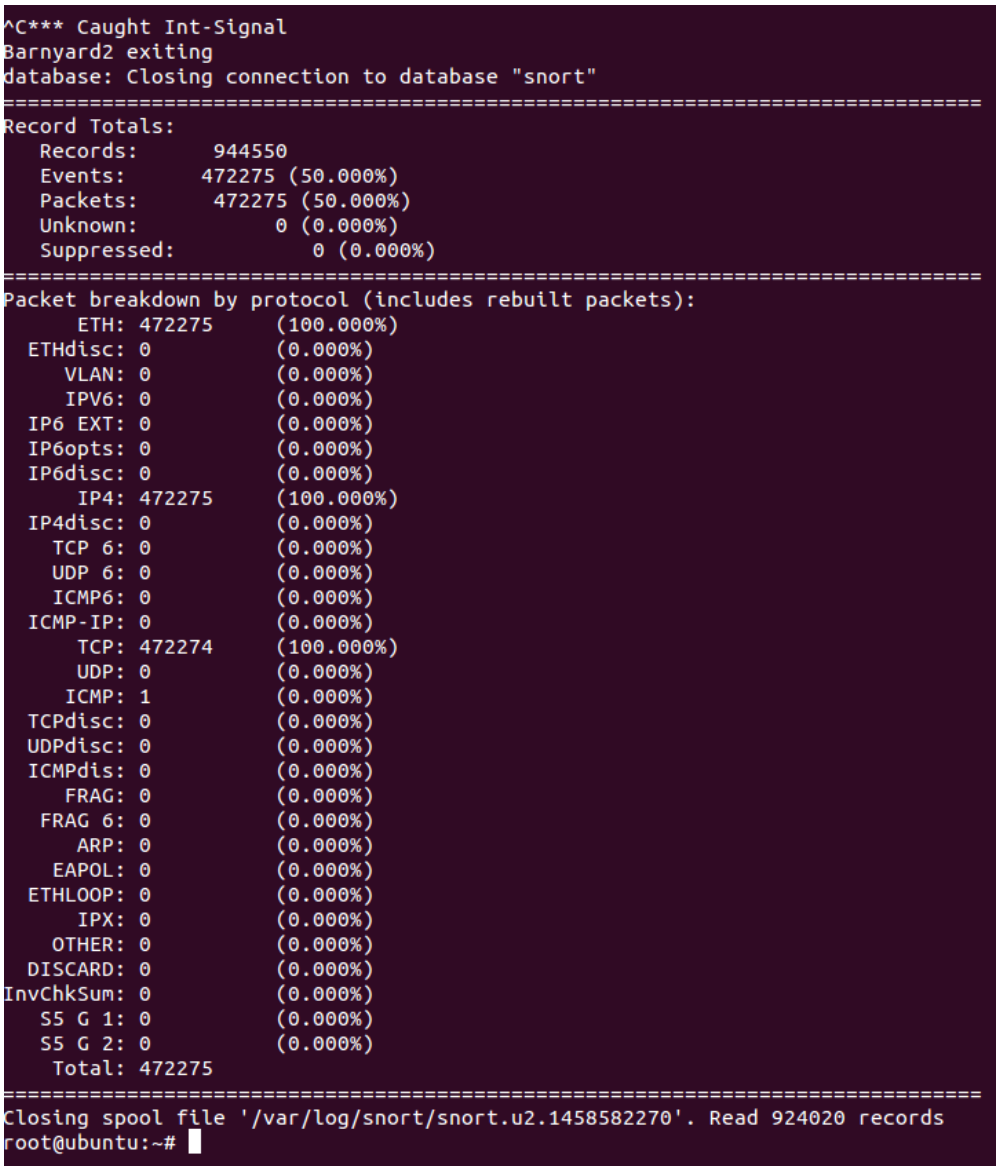
--flood: αποστολή πακέτων όσο το δυνατόν γρηγορότερα, χωρίς να φαίνονται οι απαντήσεις στην οθόνη

--rand source: χρήση τυχαίας διεύθυνσης IP για τον επιτιθέμενο



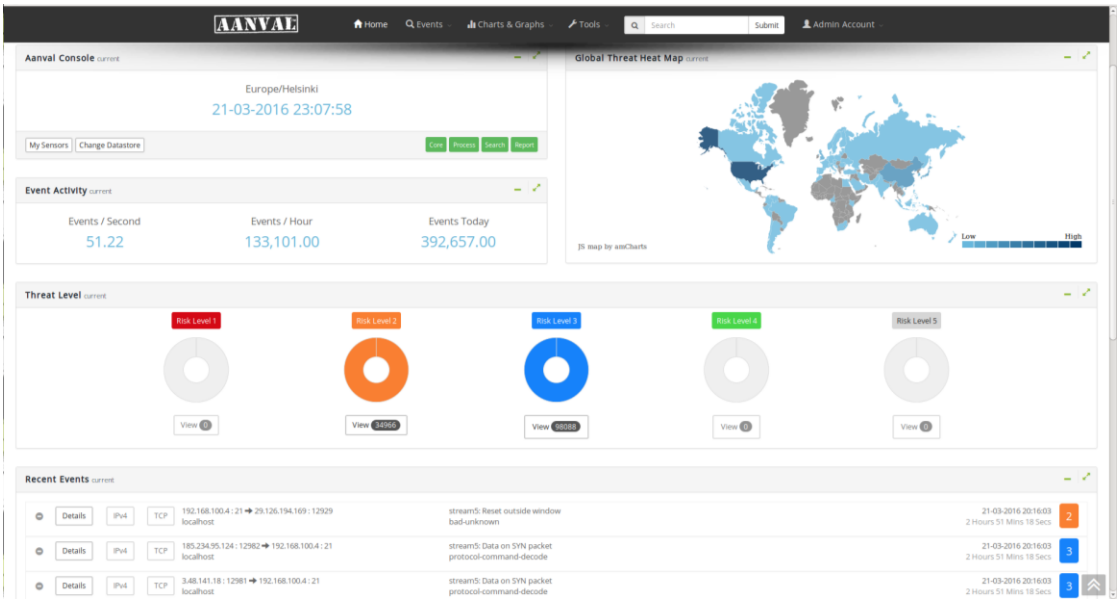
Εικόνα 27. Snort alerts για το hping3

Η επίθεση διήρκεσε 24 λεπτά, όπου εστάλησαν 14620519 πακέτα. Το snort χρειάστηκε περίπου 3 ώρες για να ολοκληρώσει την επεξεργασία της επίθεσης.

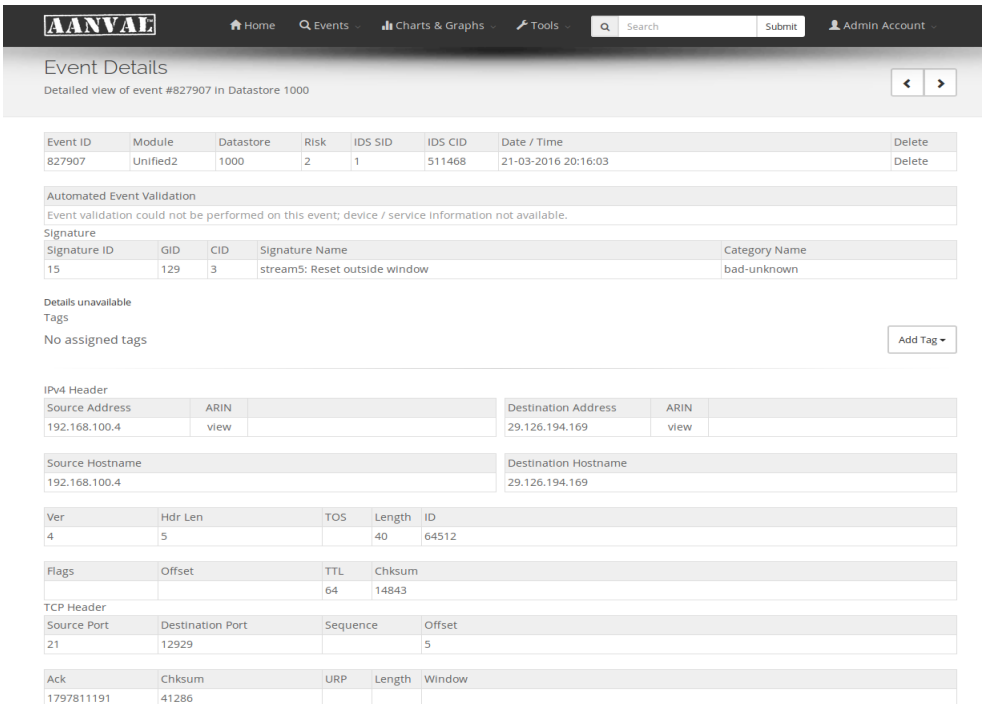


Εικόνα 28. Αποτελέσματα Snort

Στο aanval εμφανίστηκαν τα ακόλουθα αποτελέσματα. Στον χάρτη φαίνεται ότι οι επιθέσεις προήλθαν από διαφορετικές τοποθεσίες ανά τον κόσμο λόγω της χρήσης τυχαίας IP διεύθυνσης.



Εικόνα 29. Γενικό μενού του Aanval μετά την επίθεση hping3



Εικόνα 30. Snort alert από την επίθεση hping3 όπως φαίνεται στο Aanval

Στο VM-θύμα παρατηρήθηκε καθυστέρηση στο ping (109ms average) σε σχέση με το συνηθισμένο που είναι περίπου 81 ms καθώς και packet loss.

```
orestis@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=84.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=93.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=100 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=101 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=103 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=100 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=128 time=142 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=128 time=96.5 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=128 time=175 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=128 time=93.3 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=128 time=142 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=128 time=83.4 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=128 time=122 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=128 time=110 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=128 time=103 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=128 time=102 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=128 time=85.1 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=128 time=146 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=128 time=87.5 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=128 time=121 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=128 time=96.7 ms
^C
--- 8.8.8.8 ping statistics ---
23 packets transmitted, 21 received, 8% packet loss, time 22294ms
rtt min/avg/max/mdev = 83.467/109.288/175.955/23.702 ms
```

Εικόνα 31. Καθυστέρηση δικτύου και packet loss στο VM-θύμα κατά την επίθεση

```
orestis@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=82.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=81.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=81.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=81.2 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=81.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=81.4 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=128 time=81.4 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=128 time=81.0 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=128 time=81.2 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=128 time=81.1 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=128 time=81.6 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=128 time=83.2 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=128 time=80.8 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=128 time=81.4 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=128 time=81.0 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=128 time=81.7 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=128 time=83.2 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=128 time=83.3 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=128 time=81.3 ms
^C
--- 8.8.8.8 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18034ms
rtt min/avg/max/mdev = 80.859/81.685/83.319/0.849 ms
```

Εικόνα 32. Ping στο VM-θύμα κάτω από φυσιολογικές συνθήκες

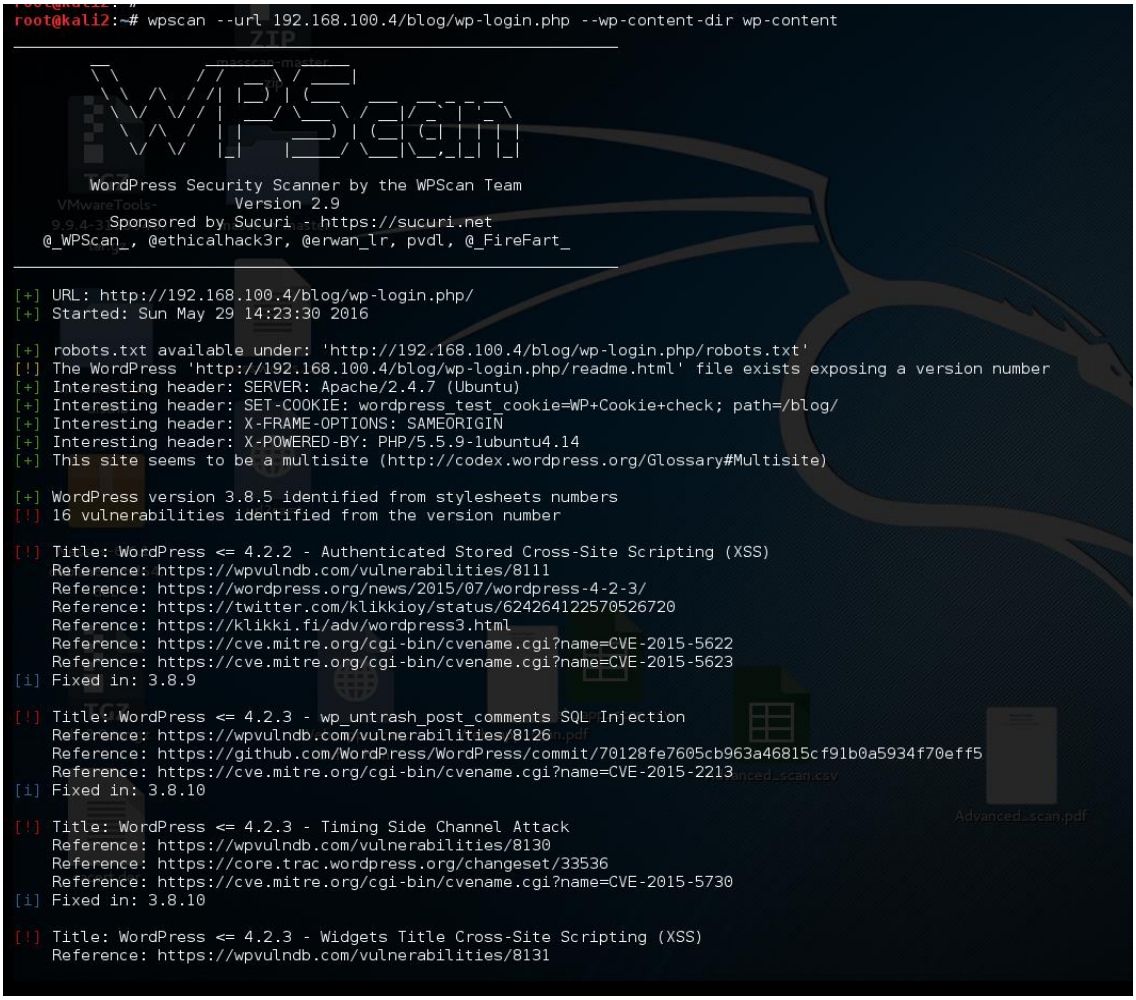
4.3 Επιθέσεις στο Wordpress

4.3.1 WPScan

Το WPScan είναι ένα vulnerability scanner για Wordpress sites [15]. Εκτελέστηκε με την παρακάτω εντολή:

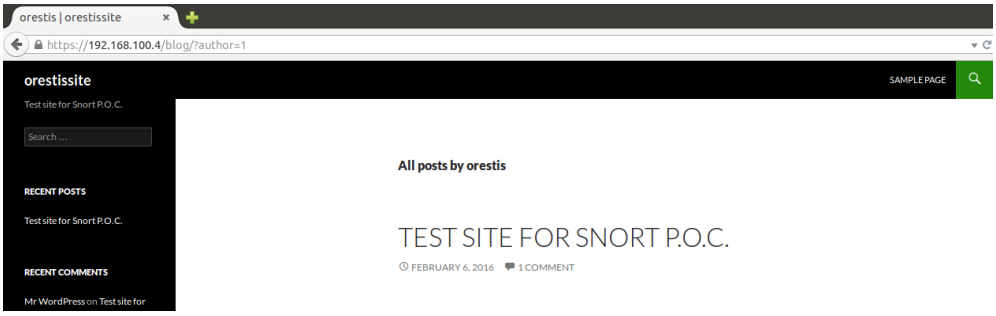
wpscan --url 192.168.100.4/blog/wp-login.php --wp-content-dir wp-content

και εμφανίστηκαν διάφορες αδυναμίες της συγκεκριμένης έκδοσης (3.8.5) όπως XSS, SQL injection, Time side channel attack κλπ.



Εικόνα 33. Εκτέλεση WPScan

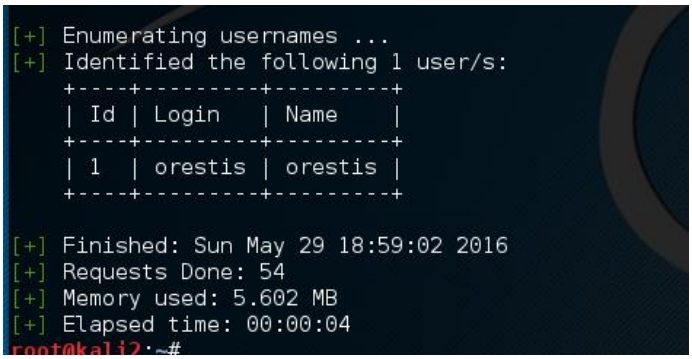
Αν κάποιος ανοίξει το https://192.168.100.4/blog/?author=1 βλέπει ότι ο user και λογικά ο admin του site έχει το username orestis.



Εικόνα 34. Admin του wordpress site

Το οποίο μπορεί να βρεθεί και με το wpscan με την εντολή:

wpscan --url 192.168.100.4/blog --enumerate u

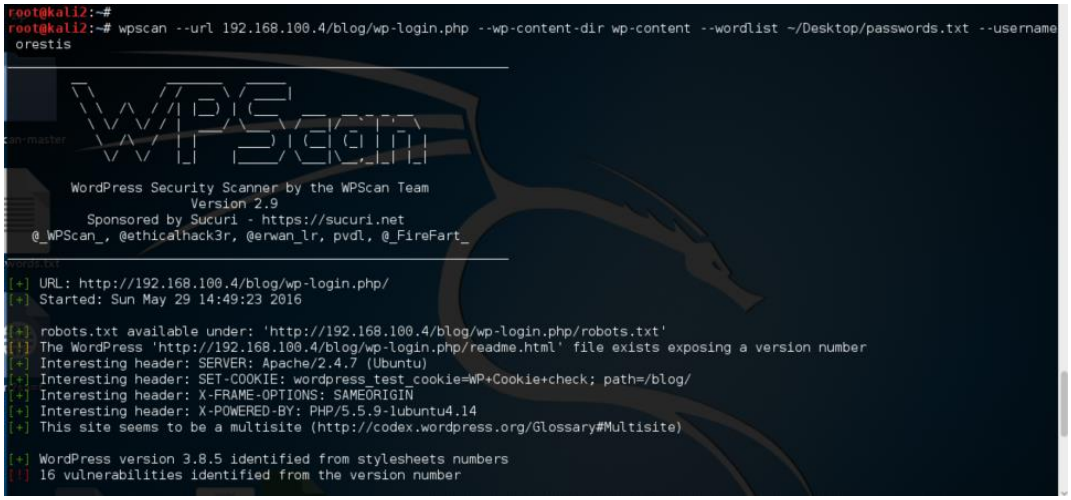


Εικόνα 35. Όλοι οι users του wordpress site

Επομένως, αυτό το username θα χρησιμοποιηθεί για να γίνει brute force για το password του admin. Αυτό επιτυγχάνεται με την ακόλουθη εντολή:

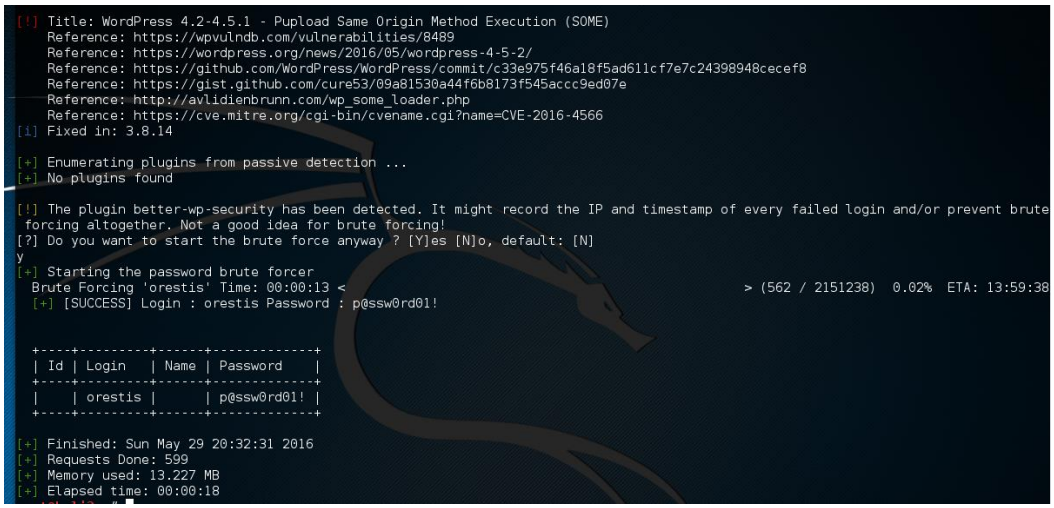
wpscan --url 192.168.100.4/blog/wp-login.php --wp-content-dir wp-content --wordlist ~/Desktop/passwords.txt --username orestis

όπου χρησιμοποιείται το αρχείο passwords.txt που περιλαμβάνει περίπου 2.000.000 παραδείγματα passwords.



Εικόνα 36. Brute force με το wpscan

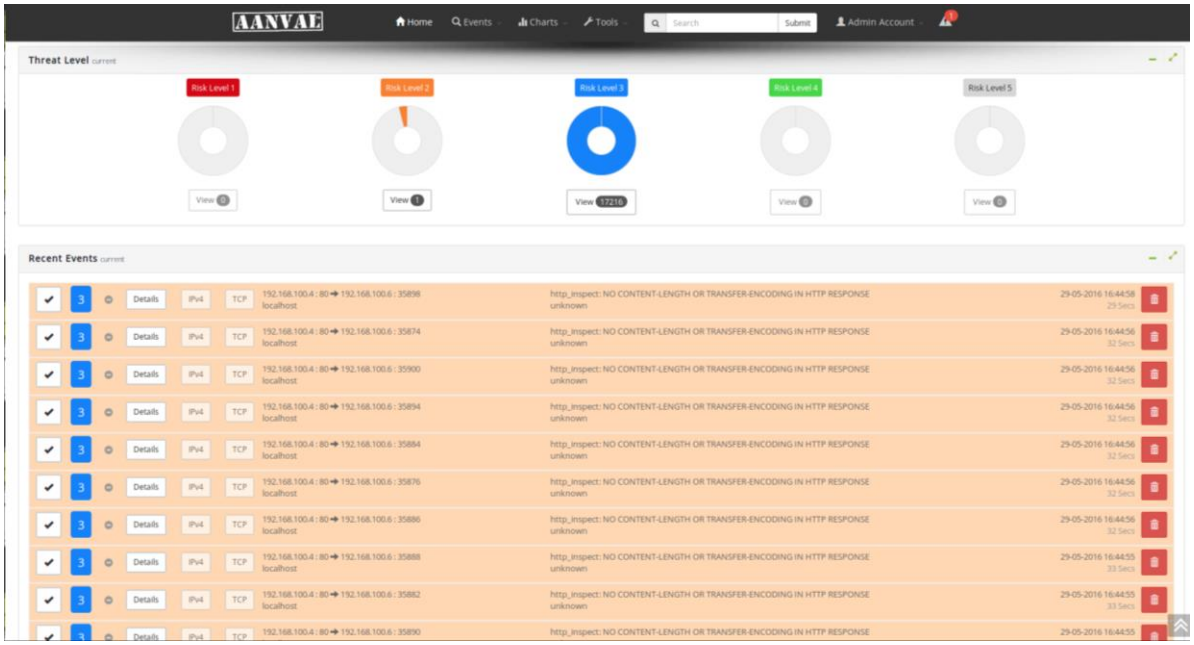
Η διαδικασία ολοκληρώνεται μετά από κάποια δευτερόλεπτα και είναι επιτυχής.



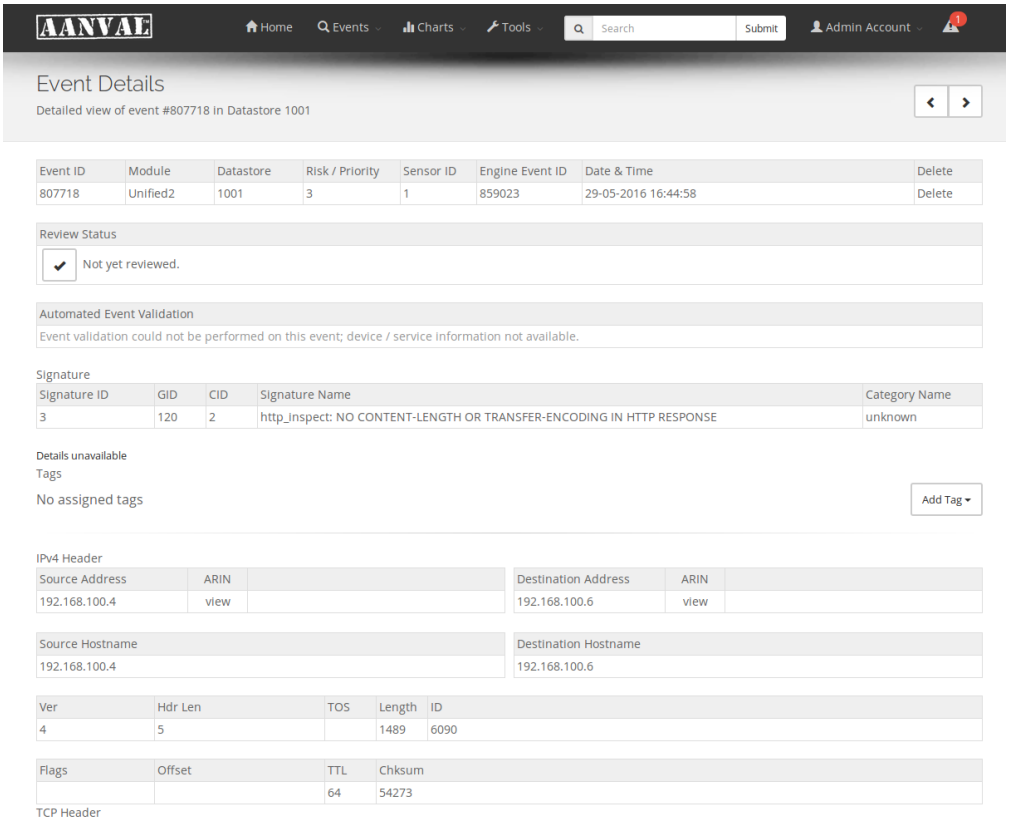
Εικόνα 37. Αποτέλεσματα password brute force με το wpscan

Στην συγκεκριμένη περίπτωση το snort έβγαλε μόνο το alert:

http_inspect: NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE



Εικόνα 38. Brute force alerts



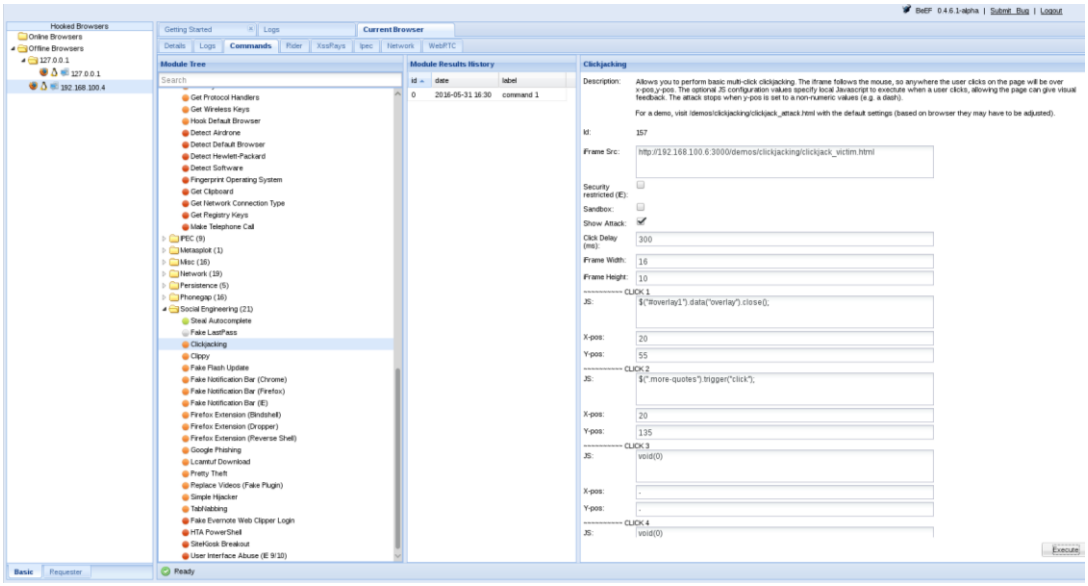
Εικόνα 39. Brute force alert http_inspect

Το οποίο και είναι false positive alert (βλ. Κεφάλαιο False Positives).

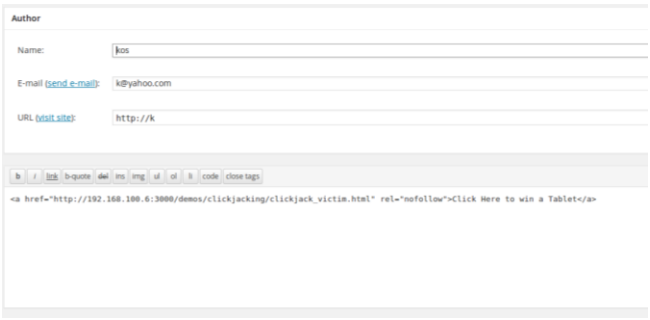
4.3.2 Επίθεση Clickjacking

Η επίθεση clickjacking έχει ως στόχο να εξαπατήσει τον χρήστη ούτως ώστε να κάνει click σε ένα link και να οδηγηθεί εκεί που θέλει ο κακόβουλος. Για την επίθεση αυτή χρησιμοποιήθηκε το εργαλείο BeEF (Browser Exploitation Framework) το οποίο έχει ως στόχο τον web browser [16].

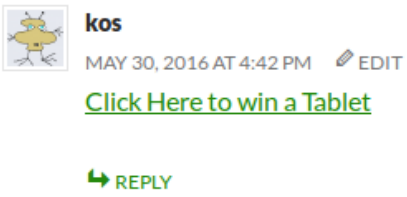
Έγινε το κατάλληλο configuration στο BeEF και φτιάχτηκε και ένα σχόλιο στο wordpress το οποίο έχει τη μορφή: “Click here to win a tablet”.



Εικόνα 40. Δημιουργία επίθεσης clickjacking μέσω BeEF

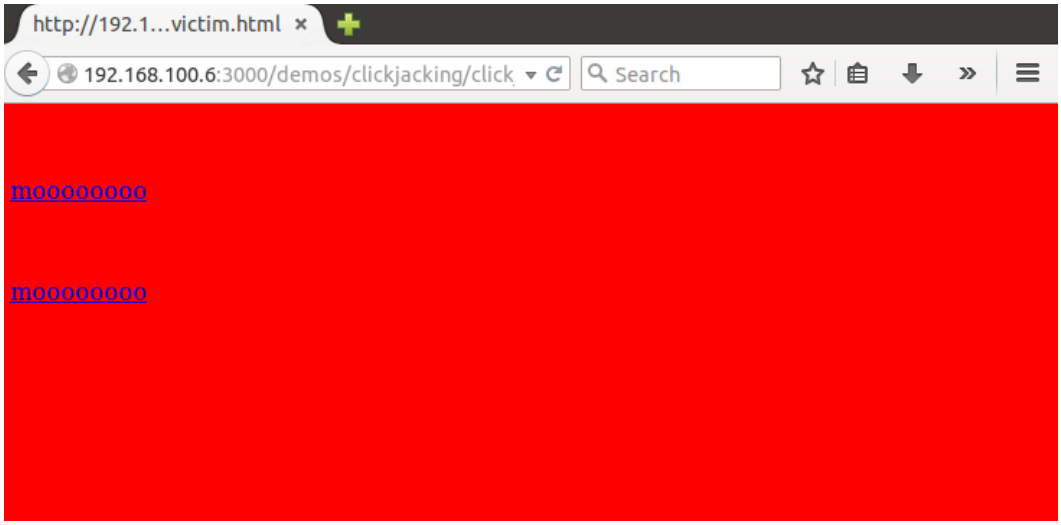


Εικόνα 41. Δημιουργία σχολίου για το clickjacking



Εικόνα 42. Εμφάνιση σχολίου

Στο οποίο αν κάνει click κάποιος χρήστης καταλήγει στο παρακάτω αποτέλεσμα.



Εικόνα 43. Αποτέλεσμα του clickjacking

Με αυτόν τον τρόπο μπορεί ο επιτιθέμενος να οδηγήσει τον χρήστη εκεί που θέλει π.χ. facebook login page, email login page και να πάρει τα credentials του.

Το snort δεν ήταν ικανό να αναγνωρίσει τη συγκεκριμένη επίθεση όπως φάνηκε στο Aanval.

4.3.3 XSS

Οι επιθέσεις τύπου XSS εισάγουν κακόβουλο κώδικα (script) σε ιστοσελίδες ξεγελώντας κατά αυτόν τον τρόπο τον browser του χρήστη, ο οποίος και εκτελεί τον κώδικα αυτό. Αυτό έχει ως αποτέλεσμα ο κακόβουλος να αποκτάει πρόσβαση σε cookies, session tokens και ευαίσθητες πληροφορίες που ανταλλάσσει ο browser με την συγκεκριμένη ιστοσελίδα.

Στη συγκεκριμένη περίπτωση ανακαλύφθηκε αδυναμία για XSS μέσω των σχολίων στο site όπου και μπήκε το κακόβουλο script. Το αποτέλεσμα ήταν ένα αναδυόμενο παράθυρο όταν ο χρήστης φόρτωνε τη συγκεκριμένη σελίδα που ήταν το σχόλιο.

Εικόνα 44. Δημιουργία σχολίου για το XSS

Εικόνα 45. Αναδυόμενο παράθυρο λόγω του XSS

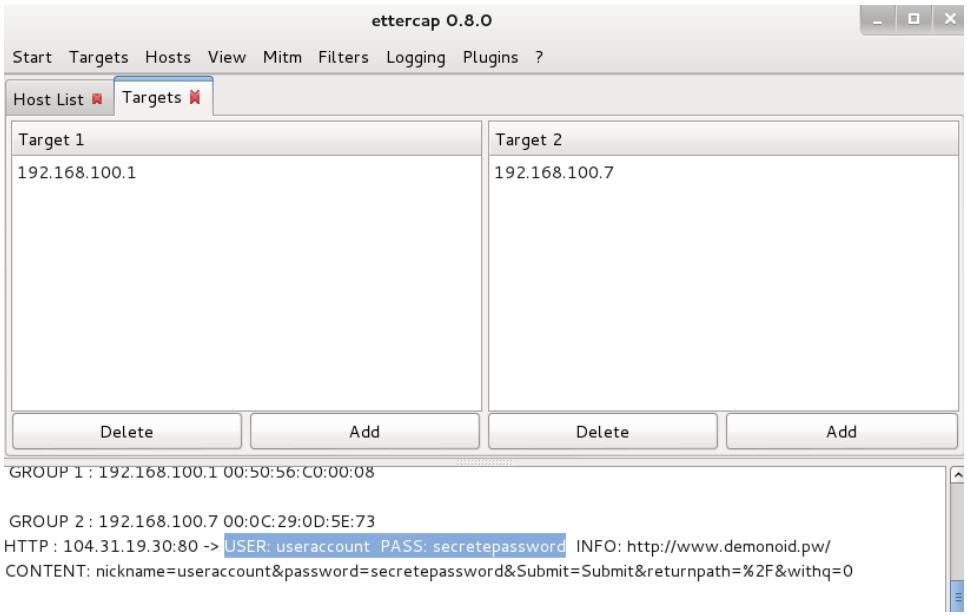
Με αυτόν τον τρόπο αποδεικνύεται ότι ο κακόβουλος μπορεί να εισάγει το κατάλληλο script ούτως ώστε να ξεγελάσει τον browser του τελικού χρήστη. Το Snort δεν έβγαλε κάποιο alert.

4.4 Επίθεση Man in the middle

Ettercap

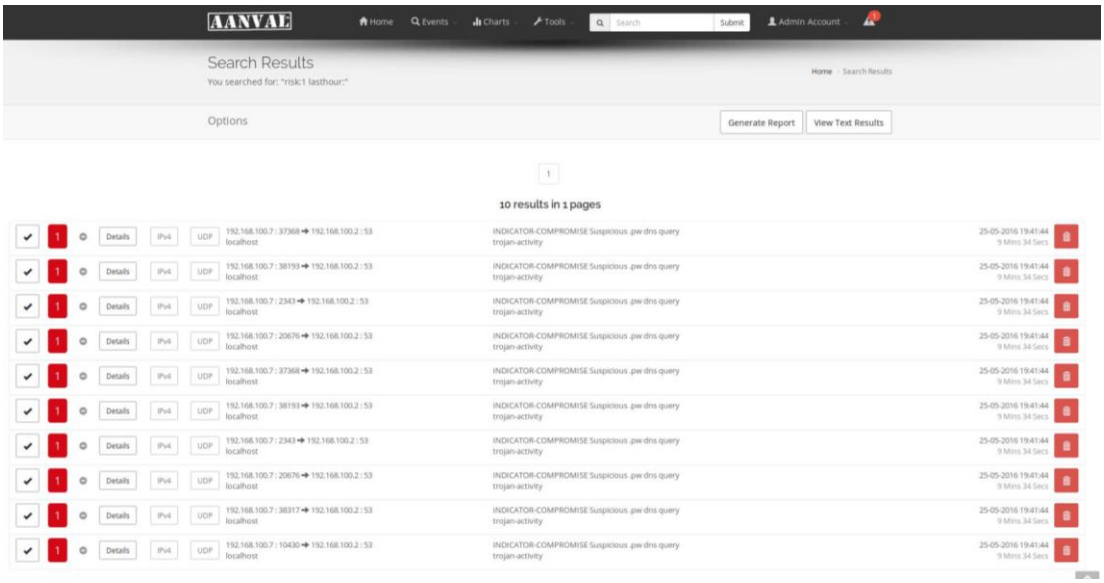
Το ettercap αποτελεί μια ολοκληρωμένη σουίτα για επιθέσεις τύπου man in the middle [17]. Μπορεί να κάνει sniffing σε live κίνηση καθώς και φιλτράρισμα περιεχομένου.

Το ettercap ρυθμίστηκε για να «ακούει» την κίνηση από την IP 192.168.100.7 μέσω arpspoofing. Όταν ο χρήστης για παράδειγμα επιχειρήσει να κάνει login σε μια σελίδα που δεν είναι https, τα credentials εμφανίζονται στο ettercap, όπως φαίνεται και στην εικόνα παρακάτω.

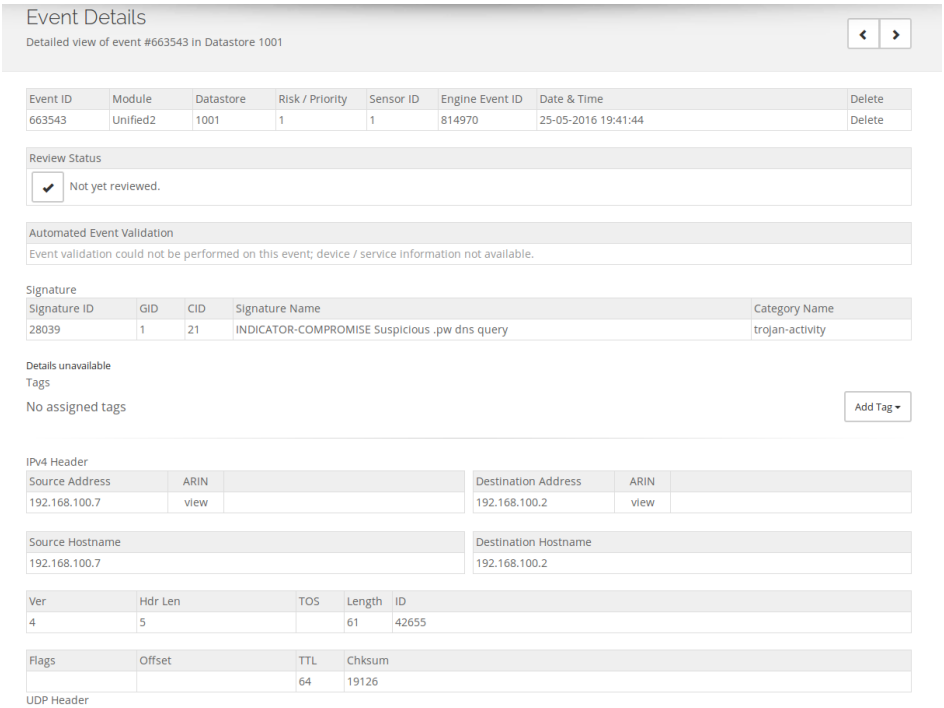


Εικόνα 46. Εμφάνιση credentials στο ettercap

Στην συγκεκριμένη περίπτωση επειδή τα domain που τελειώνουν σε .pw είναι χαρακτηρισμένα ως blacklist λόγω trojan, εμφανίζονται τα επόμενα alerts στο aanval.

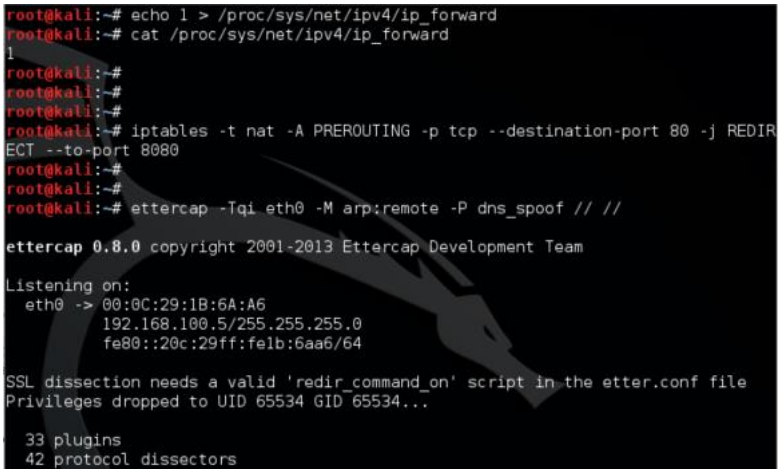


Εικόνα 47. Trojan alerts στο aanval



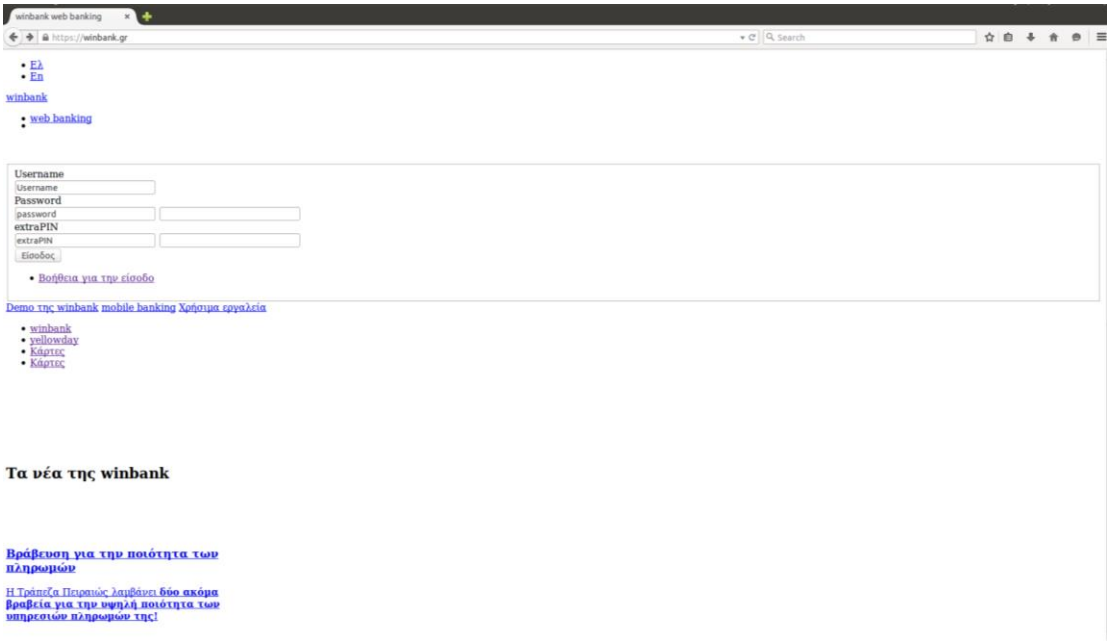
Εικόνα 48. Λεπτομέρειες του trojan alert

Δοκιμάστηκε επίθεση μέσω πρωτοκόλλου DNS στο ettercap, όπου δηλώθηκε το domain τράπεζας να αντιστοιχεί σε private IP διεύθυνση. Το αποτέλεσμα ήταν ότι ο χρήστης φόρτωσε την ψεύτικη σελίδα της τράπεζας που είχε υλοποιηθεί στο web server του επιτιθέμενου, αντί να αποκτήσει πρόσβαση στο πραγματικό site της. Έτσι ο κακόβουλος είναι ικανός να αποσπάσει πρόσβαση στο e-banking του θύματος.



Εικόνα 49. Υλοποίηση επίθεσης στο πρωτόκολλο DNS μέσω ettercap

Εικόνα 50. Ενεργοποίηση DNS spoof plugin



Εικόνα 51. Ψεύτικη ιστοσελίδα της τράπεζας

Το aanval δεν εμφάνισε κάτι σε αυτήν την επίθεση.

4.5 Επίθεση σε Windows

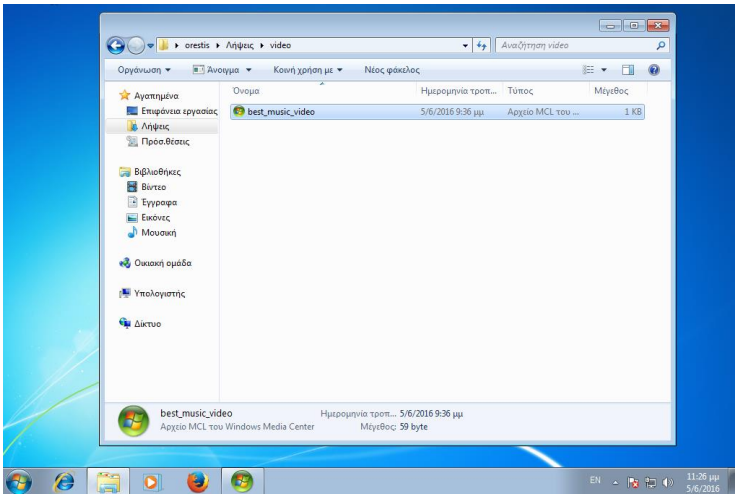
Windows 7 backdoor

Σύμφωνα με το CVE-2015-2509 [18] τα Windows Vista SP2, Windows 7 SP1, Windows 8 και Windows 8.1 επιτρέπουν σε επιτιθέμενους να αποκτήσουν απομακρυσμένη πρόσβαση σε αυτά μέσω ενός Media Center link file (mcl). Για την επίθεση χρησιμοποιήθηκε το Metasploit [19] όπου μέσω του module ms15_100_mcl φτιάχνεται το κακόβουλο αρχείο, το οποίο και αποστέλλεται στο θύμα.

```
msf exploit(ms15_100_mcl_exe) > exploit

[-] Exploit failed: The following options failed to validate: LHOST.
msf exploit(ms15_100_mcl_exe) > set LHOST 192.168.100.6
LHOST => 192.168.100.6
msf exploit(ms15_100_mcl_exe) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.100.6:4444
[*] Server started.
msf exploit(ms15_100_mcl_exe) > [*] Malicious executable at \\192.168.100.6\xAIPJT\best_video.exe...
[*] Creating 'best_music_video.mcl' file ...
[+] best_music_video.mcl stored at /root/.msf5/local/best_music_video.mcl
```

Εικόνα 52. Δημιουργία αρχείου μέσω του module ms15_100_mcl

Μόλις το θύμα τρέξει το αρχείο, ανοίγει το reverse session.

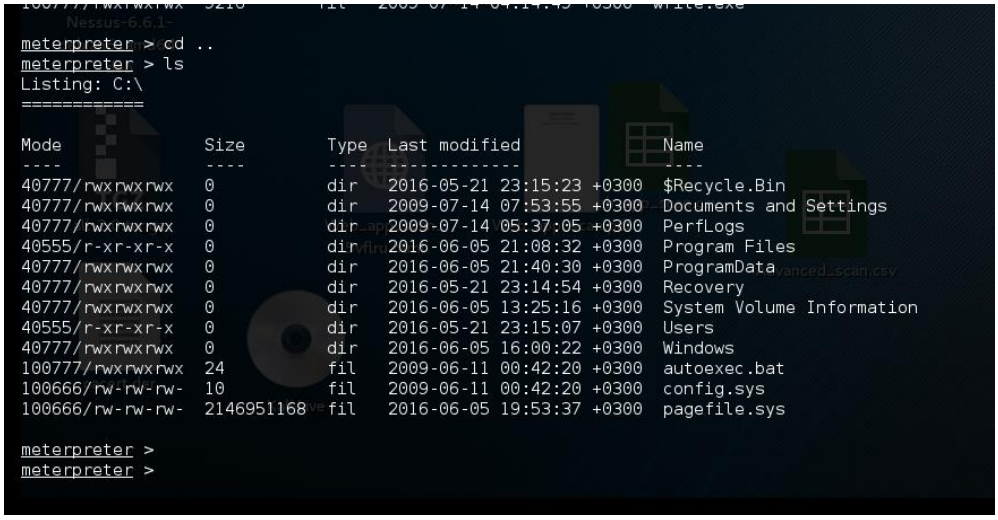


Εικόνα 53. Εκτέλεση αρχείου

```
msf exploit(ms15_100_mcl_exe) > sessions -l

Active sessions
===============

No active sessions.

msf exploit(ms15_100_mcl_exe) >
[*] Sending stage (957487 bytes) to 192.168.100.18
[*] Meterpreter session 1 opened (192.168.100.6:4444 -> 192.168.100.18:49821) at 2016-06-05 21:38:16 +0300
sessions -l

Active sessions
===============

  Id  Type                   Information                  Connection
  --  ----                   -----------                  ----------
  1   meterpreter x86/win32  orestis-PC\orestis @ ORESTIS-PC  192.168.100.6:4444 -> 192.168.100.18:49821 (192.168.100.18)
```

Εικόνα 54. Άνοιγμα reverse session

Με το άνοιγμα του reverse session ο κακόβουλος έχει πλήρη πρόσβαση στον υπολογιστή του θύματος και έχει δικαιώματα ίδια με του χρήστη που έτρεξε το αρχείο.



Εικόνα 55. Περιεχόμενα του C directory των Windows

Το Snort δεν αναγνώρισε την επίθεση, ακόμα και όταν έγιναν uncomment *indicator-shellcode.rules,* τα σχετικά με reverse tcp rules και μπήκαν και κάποια επιπλέον. Τα rules που έγιναν uncommented:

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"INDICATOR-SHELLCODE Metasploit payload windows_meterpreter_reverse_ord_tcp"; content:"|FC 31 DB 64 8B 43 30 8B 40 0C 8B 50 1C 8B 12 8B 72 20 AD AD 4E 03 06 3D 32 33 5F 32 75 EF 8B 6A 08 8B 45 3C 8B 4C 05 78 8B 4C 0D 1C 01 E9 8B 41|"; fast_pattern:only; classtype:shellcode-detect; sid:30476; rev:1;)

alert tcp any any -> any any (msg:"INDICATOR-SHELLCODE Metasploit windows/reverse_tcp stager transfer attempt"; flow:established; content:"|FC E8 86 00 00 00 60 89 E5 31 D2 64 8B 52 30 8B 52 0C 8B 52 14 8B 72 28 0F B7 4A 26 31 FF 31 C0 AC 3C 61 7C 02 2C 20 C1 CF 0D 01 C7 E2 F0 52 57|"; fast_pattern:only; reference:url,www.metasploit.com/learn-more/how-do-i-use-it/documentation.jsp; classtype:shellcode-detect; sid:30227; rev:1;)

Και τα rules που μπήκαν:

alert tcp any any -> any any (msg:"Metasploit User Agent String"; flow:to_server,established; *content:"User-Agent|3a| Mozilla/4.0 (compatible\; MSIE 6.1\; Windows NT)|0d 0a|"; http_header; classtype:trojan-activity; reference:url,blog.didierstevens.com/2015/03/16/quickpost-metasploit-user-agent-strings/; sid:1618001; rev:1;)*

alert tcp any any -> any any (msg:"Metasploit User Agent String"; flow:to_server,established; content:"User-Agent|3a| Mozilla/4.0 (compatible\; MSIE 7.0\; Windows NT 6.0\; Trident/4.0\; SIMBAR={7DB0F6DE-8DE7-4841-9084-28FA914B0F2E}\; SLCC1\; .N|0d 0a|"; http_header; classtype:trojan-activity; reference:url,blog.didierstevens.com/2015/03/16/quickpost-metasploit-user-agent-strings/; sid:1618003; rev:1;)

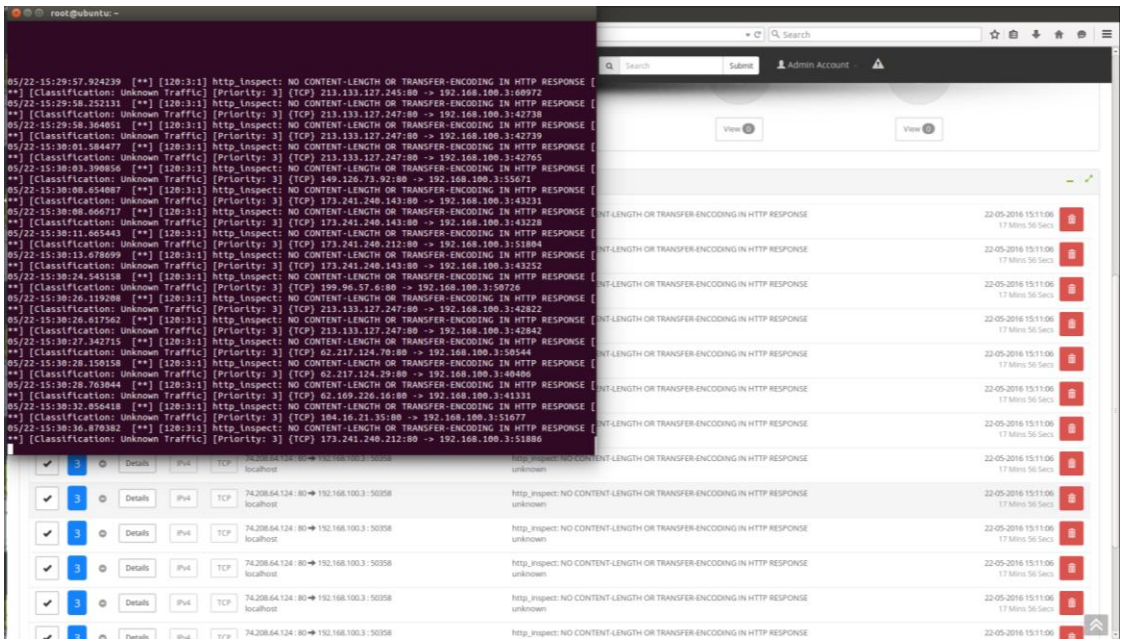
alert tcp any any -> any any (msg:"Metasploit User Agent String"; flow:to_server,established; content:"User-Agent|3a| Mozilla/4.0 (compatible\; Metasploit RSPEC)|0d 0a|"; http_header; classtype:trojan-activity; reference:url,blog.didierstevens.com/2015/03/16/quickpost-metasploit-user-agent-strings/; sid:1618004; rev:1;)

alert tcp any any -> any any (msg:"Metasploit User Agent String"; flow:to_server,established; content:"User-Agent|3a| Mozilla/5.0 (compatible\; Googlebot/2.1\; +http://www.google.com/bot.html)|0d 0a|"; http_header; classtype:trojan-activity; reference:url,blog.didierstevens.com/2015/03/16/quickpost-metasploit-user-agent-strings/; sid:1618006; rev:1;)

4.6 False positives

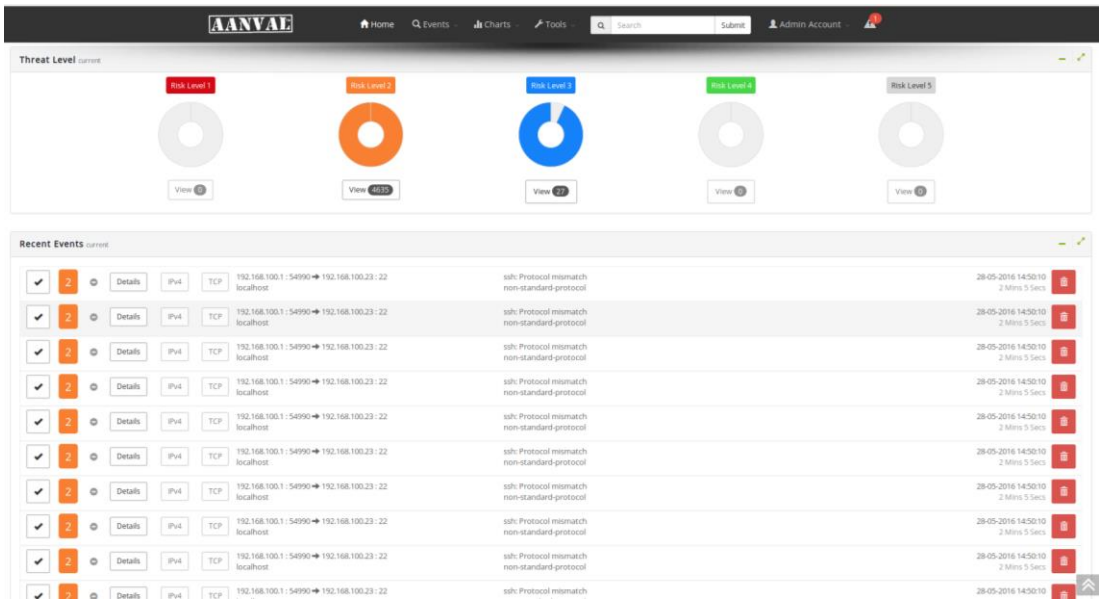
Κατά την χρήση του snort υπήρχαν false positives, δηλαδή alerts τα οποία έκαναν την εμφάνισή τους χωρίς να υπάρχει κάποια επίθεση. Το πιο συνηθισμένο alert που εμφανιζόταν κατά το web browsing από κάποιο VM ήταν το:

http_inspect: NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE

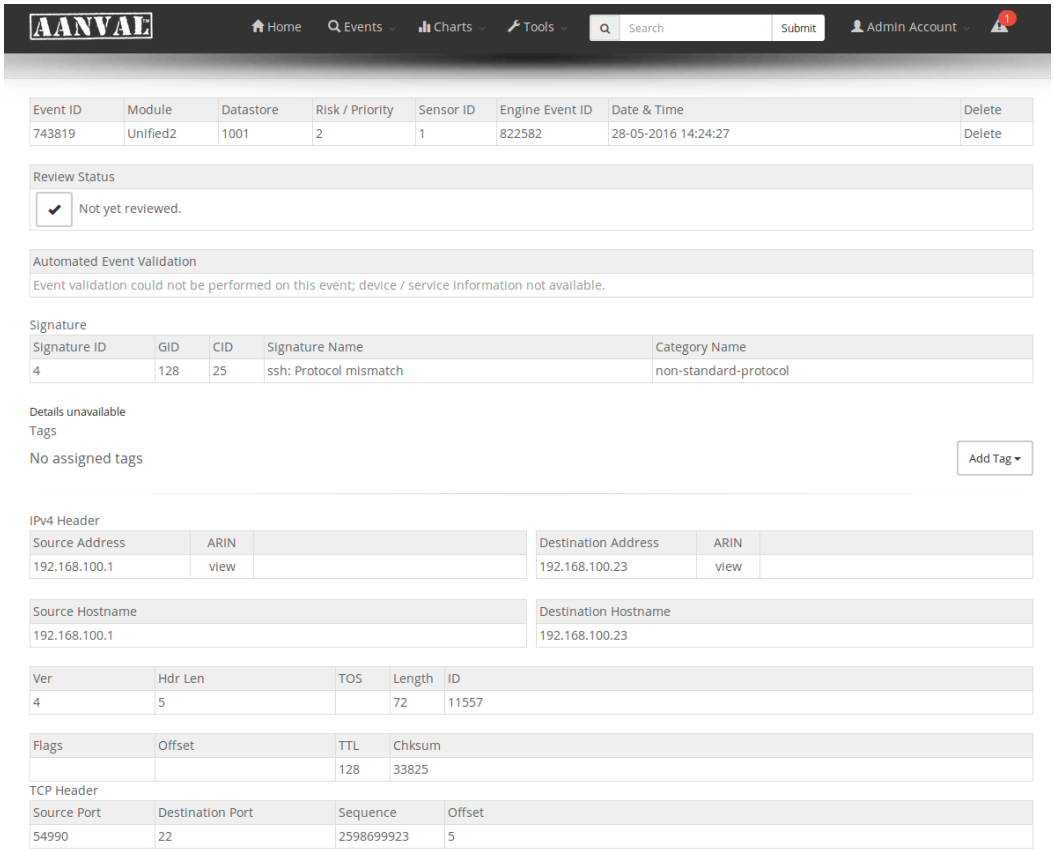


Εικόνα 56. False positive http_inspect

False positives υπήρχαν και κατά τη σύνδεση μέσω SSH πρωτοκόλλου σε κάποιο VM. Πιο συγκεκριμένα, εμφανίζονται πολλαπλά alerts κατά όταν υπήρχε σύνδεση SSH λόγω ενός bug παλαιότερης έκδοσης του Snort, το οποίο όπως φαίνεται υπάρχει ακόμα.



Εικόνα 57. SSH false positives alerts στο Aanval



Εικόνα 58. SSH false positive alert

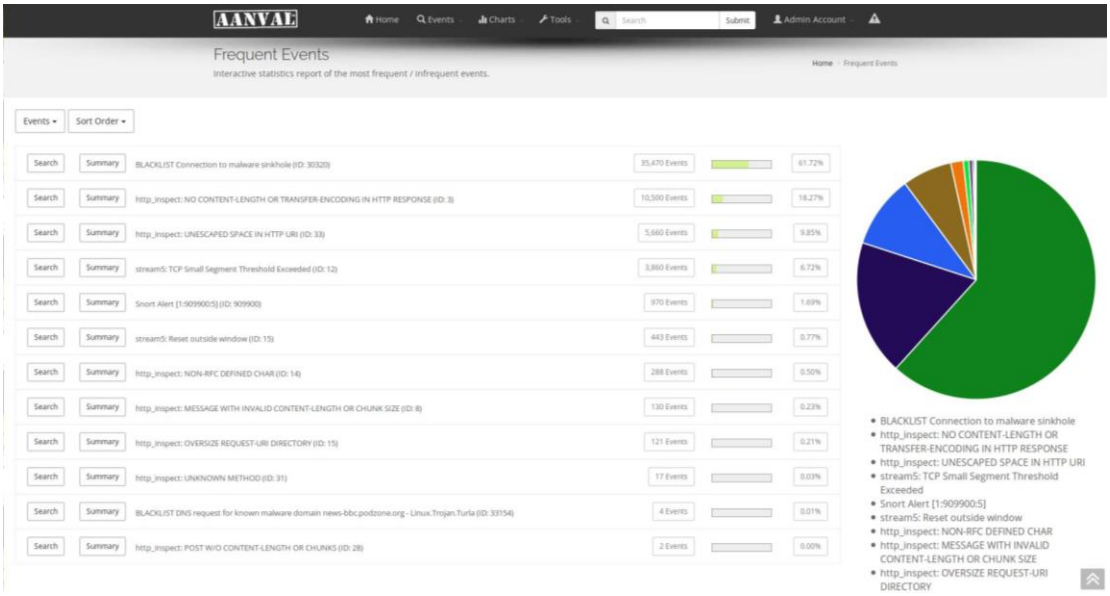
Για την επίλυση του συγκεκριμένου false positive, έγινε commented στο preprocessor.rules το συγκεκριμένο signature που προκαλούσε το alert.

Εικόνα 59. SSH rule που έγινε commented

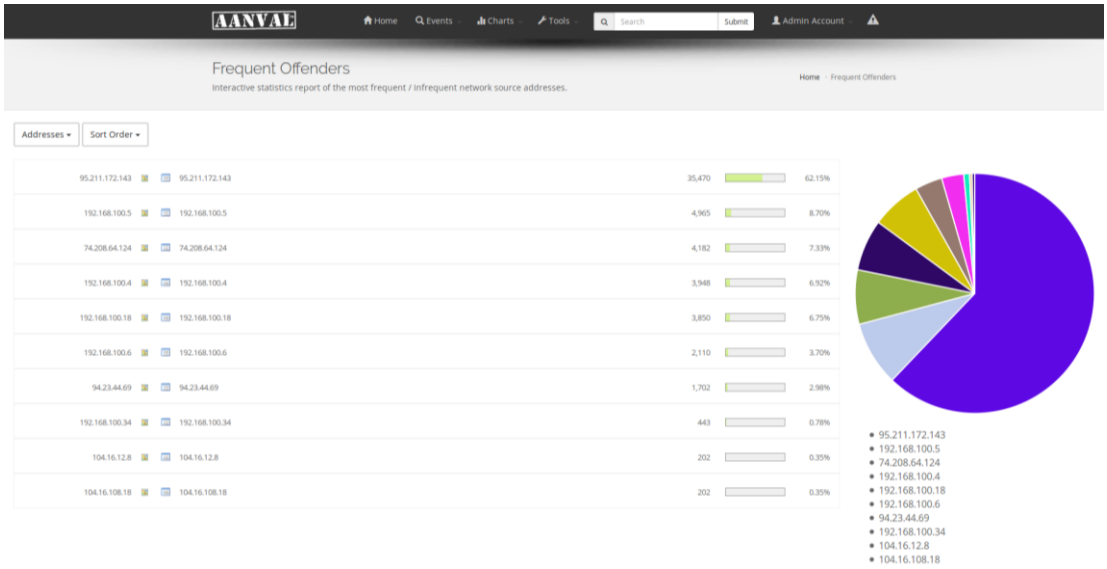
Αποτελέσματα και Συμπεράσματα

Το γραφικό περιβάλλον του Aanval διατήρησε κάποια συνολικά αποτελέσματα που φαίνονται παρακάτω. Πιο συγκεκριμένα, υπήρχαν αποτελέσματα σχετικά με τη συχνότητα εμφάνισης των rules, των IP διευθύνσεων που εξαπέλυαν τις επιθέσεις και των διευθύνσεων που ήταν οι στόχοι, των ports που εμφανιζόταν η περισσότερη κίνηση και των επιπέδων κρισιμότητας των rules.

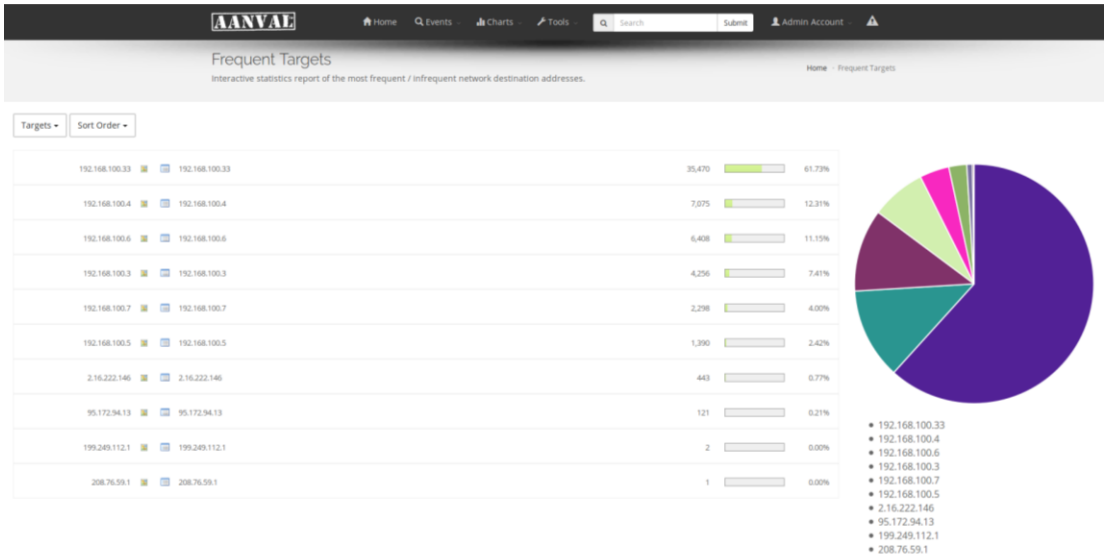
Στα συγκεκριμένα αποτελέσματα αναπόφευκτα υπάρχουν πάρα πολλά false positives. Παρόλα αυτά περιλαμβάνονται οι διευθύνσεις IP των VM που έλαβαν μέρος στην υλοποίηση που είναι της μορφής 192.168.100.xxx.



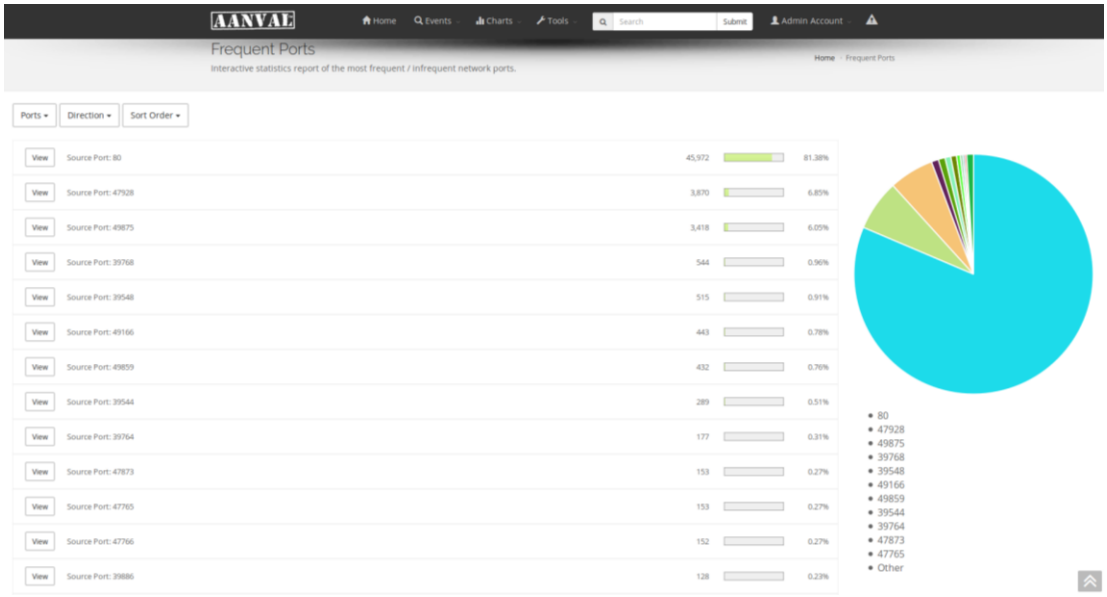
Εικόνα 60. Συχνότητα εμφάνισης των rules



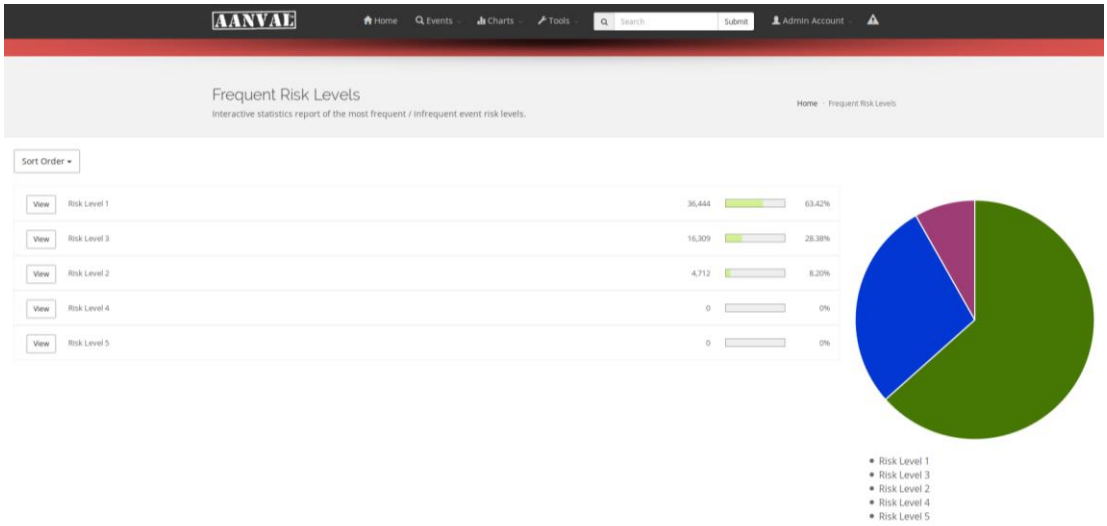
Εικόνα 61. Συχνότητα εμφάνισης IP διευθύνσεων των επιτιθέμενων



Εικόνα 62. Συχνότητα εμφάνισης IP διευθύνσεων των στόχων



Εικόνα 63. Συχνότητα εμφάνισης των ports με την περισσότερη κίνηση



Εικόνα 64. Συχνότητα εμφάνισης των επιπέδων κρισιμότητας των rules

Το Snort όπως φάνηκε μπορεί να αποτελέσει ένα πολύ ισχυρό εργαλείο στα χέρια ενός έμπειρου Network Security administrator. Οι δυνατότητες που έχει είναι πάρα πολλές και ο χρήστης του μπορεί να το παραμετροποιήσει όπως αυτός θέλει και να εκμηδενίσει τα false positives. Είναι εφικτό να εγκατασταθεί ακόμα και σε εταιρικό περιβάλλον όπου οι απαιτήσεις είναι αυξημένες και να προστατεύει το δίκτυο χωρίς προβλήματα.

Το Snort θα μπορούσε να εγκατασταθεί σε ένα περιβάλλον cloud και να προστατεύει την υποδομή του χωρίς σημαντικό κόστος. Κατά τις επιθέσεις που δοκιμάστηκαν το Snort, ως ένα Network-Based Intrusion Detection System, κατάφερε να ανακαλύψει τις επιθέσεις που αφορούσαν το δικτυακό κομμάτι και κυρίως τις επιθέσεις αναγνώρισης δικτύου και τις επιθέσεις

Denial of Services (DoS). Η αποτροπή των επιθέσεων αναγνώρισης εμποδίζει τον κακόβουλο να συνεχίσει στην κύρια επίθεση. Επίσης, η αποτροπή των επιθέσεων DoS εκμηδενίζει το downtime ενός server ή μιας εφαρμογής το οποίο μεταφράζεται σε κόστος για τον ιδιοκτήτη.

Η καλύτερη λύση σε ένα cloud περιβάλλον είναι ο συνδυασμός NIDS και HIDS τα οποία θα είναι κατανεμημένα σε διάφορα σημεία του δικτύου (NIDS), και σε όλους τους hosts (HIDS) και θα αποστέλλουν τα logs ή τα alerts σε κάποιο κεντρικό σημείο για καλύτερη διαχείριση.

Βιβλιογραφία

[1] P. M. Karern Scarfone, Guide to Intrusion Detection and Prevention Systems (IPDS), NIST, 2007.

[2] Intrusion Detection Systems, (IATAC), Information Assurance Technology Analysis Center, 2009.

[3] Intrusion Detection Systems: Definition, Need and Challenges, SANS Institute, 2001.

[4] T. G. Peter Mell, The NIST Definition of Cloud Computing, 2009.

[5] "Cloud computing," [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing. [Accessed 12 6 2016].

[6] Introduction to Cloud Computing, Dialogic Corporation, 2010.

[7] "Types of cloud computing," Amazon, [Online]. Available: https://aws.amazon.com/types-of-cloud-computing/. [Accessed 12 6 2016].

[8] Q. F. Hassan, Demystifying Cloud Computing, 2011.

[9] "Okeanos," [Online]. Available: https://okeanos.grnet.gr/.

[10] R. U. Rehman, Intrusion Detection Systems with Snort, New Jersey: Pearson Education, 2003.

[11] "Snort," [Online]. Available: https://snort.org/.

[12] "Snort (software)," [Online]. Available: https://en.wikipedia.org/wiki/Snort_(software). [Accessed 15 6 2016].

[13] "Nmap," [Online]. Available: https://nmap.org/.

[14] "Nessus," [Online]. Available: https://www.tenable.com/products/nessus-vulnerability-scanner.

[15] "WPScan," [Online]. Available: http://wpscan.org/.

[16] "BeEF," [Online]. Available: http://beefproject.com/.

[17] "Ettercap," [Online]. Available: https://ettercap.github.io/ettercap/.

[18] "CVE-2015-2509," [Online]. Available: https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-2509.

[19] "Metasploit," [Online]. Available: https://www.metasploit.com/.