



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος **Ασφάλεια Δικτύων**

Άσκηση	Απαλλακτική Εργασία
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Κουσουννής Κωνσταντίνος ρ14086
Ημερομηνία παράδοσης	24-06-2019



Εκφώνηση Άσκησης

2. Σχεδιασμός Πολιτικής Ανίχνευσης Εισβολών Δικτύου και υλοποίηση της πολιτικής με τη διαμόρφωση του network-based IDS συστήματος bro.

Να αναλύσετε, να σχεδιάσετε και να υλοποιήσετε μέσω της κατάλληλης διαμόρφωσης του Network-based IDS συστήματος snort μία Πολιτική Ανίχνευσης Δικτυακών Εισβολών (Intrusion Detection Policy), με βάση τα παρακάτω βήματα. Η εργασία θα συνοδεύεται από κατάλληλη τεκμηρίωση η οποία θα περιλαμβάνει αναλυτική περιγραφή με τα παρακάτω στοιχεία.

Σημεία υλοποίησης στην υποδομή: Η διαμόρφωση του snort μπορεί να γίνει σε εικονικό μηχάνημα το οποίο θα έχει το ρόλο του DMZ (VM-2).

Α) Περιγραφή συστήματος και Καθορισμός Πολιτικής Ανίχνευσης Εισβολών (5 μονάδες)

- Περιγραφή του συστήματος snort. Περιγραφή συστημάτων εισβολών δικτύου (Network IDS), βασικές λειτουργίες, περιγραφή του συστήματος snort (βασικές λειτουργίες, οδηγίες εγκατάστασης, βασικές οδηγίες διαμόρφωσης και χρήσης (περίπου 2000 λέξεις, 2 μονάδες)).
- Καθορισμός της Πολιτικής Ανίχνευσης Εισβολών. Σε αυτό το βήμα θα περιγράψετε την Πολιτική Ανίχνευσης Εισβολών Δικτύου που θα πρέπει να ικανοποιείται για το υπό έλεγχο δίκτυο. Θα πρέπει να περιλαμβάνει κατ' ελάχιστο την περιγραφή των υπηρεσιών που πρέπει να ελέγχονται, το επίπεδο ελέγχου, τους ελάχιστους ελέγχους εισβολών ανά προστατευόμενη υπηρεσία ή εσωτερικό δίκτυο, το επίπεδο καταγραφής, τη διαδικασία ελέγχου των αρχείων καταγραφής, τη διαδικασία ανανέωσης των κανόνων και της πολιτικής κτλ. (Υπόδειξη: Έμφαση θα πρέπει να δοθεί στον καθορισμό της πολιτικής ανίχνευσης εισβολών ώστε να είναι σαφής και καλά ορισμένη η πολιτική πρόσβασης για κάθε υπηρεσία ή εσωτερική περιοχή δικτύου που πρέπει να προστατευθεί. Η υλοποίηση που θα πραγματοποιηθεί στο Β' μέρος της εργασίας θα πρέπει να ακολουθεί την πολιτική αυτή) (3 μονάδες).

Β) Υλοποίηση και έλεγχος εφαρμογής της πολιτικής ανίχνευσης εισβολών (5 μονάδες)

- Διαμόρφωση συστήματος ανίχνευσης εισβολών (IDS). Να διαμορφώσετε κατάλληλα το σύστημα snort ώστε να υλοποιεί την Πολιτική Ανίχνευσης Δικτυακών Εισβολών, όπως ορίστηκε στο Α μέρος. Να δημιουργηθεί και να παραδοθεί αναλυτικό εγχειρίδιο εγκατάστασης και παραμετροποίησης του λογισμικού. (3 μονάδες)
- Έλεγχος καταγραφής επιθέσεων. Από έναν άλλο κόμβο που θα έχει το ρόλο του επιτιθέμενου να δοκιμάσετε να προσομοιώσετε κάποιες επιθέσεις χρησιμοποιώντας κατάλληλα εργαλεία. Παράδειγμα επιθέσεις μπορεί να περιλαμβάνουν χαρτογράφηση του δικτύου, επιθέσεις άρνησης υπηρεσίας (Denial of Service attacks) κτλ. Να αναλύσετε τα αρχεία καταγραφής του εργαλείου (2 μονάδες)



Α) Περιγραφή συστήματος και Καθορισμός Πολιτικής Ανίχνευσης Εισβολών (5 μονάδες)

1. Περιγραφή του συστήματος snort. Περιγραφή συστημάτων εισβολών δικτύου (Network IDS), βασικές λειτουργίες, περιγραφή του συστήματος snort (βασικές λειτουργίες, οδηγίες εγκατάστασης, βασικές οδηγίες διαμόρφωσης και χρήσης (περίπου 2000 λέξεις, 2 μονάδες).

Περιγραφή συστημάτων εισβολών δικτύου (Network IDS)

Το σύστημα ανίχνευσης εισβολών δικτύου (Network IDS) αποκτά το όνομά του από το γεγονός ότι παρακολουθεί ολόκληρο το δίκτυο. Πιο συγκεκριμένα, παρακολουθεί ένα ολόκληρο τμήμα δικτύου. Κανονικά, μια κάρτα διασύνδεσης δικτύου υπολογιστή (NIC) η οποία λειτουργεί με μια μη αδιάκριτη λειτουργία (δηλαδή να μην κρυφακούει στο δίκτυο). Με αυτόν τον τρόπο λειτουργίας, μόνο τα πακέτα που προορίζονται για τη διεύθυνση ελέγχου πρόσβασης συγκεκριμένων μέσων (MAC) του Ελεγκτής διεπαφής δικτύου (NIC) προωθούνται στη στοίβα για ανάλυση. Άρα το σύστημα ανίχνευσης εισβολών δικτύου (Network IDS) παρακολουθεί την κίνηση δικτύου που δεν προορίζεται για τη δική του διεύθυνση MAC. Σε αυτή την αδιάκριτη λειτουργία (promiscuous mode), το σύστημα ανίχνευσης εισβολών δικτύου (Network IDS) μπορεί να παρακολουθήσει (κρυφακούσει) όλες τις επικοινωνίες σε ένα κομμάτι του δικτύου. Η λειτουργία σε αδιάκριτη λειτουργία (promiscuous mode) είναι απαραίτητη για την προστασία του δικτύου.

Περιγραφή του συστήματος snort

Το snort είναι ένα σύστημα ανίχνευσης εισβολών δικτύου (Network IDS) και ένα εργαλείο ανίχνευσης και καταγραφής πακέτων. Δημιουργός του είναι ο Martin Roesch ο οποίος ξεκίνησε την ανάπτυξη του κώδικα του σε γλώσσα προγραμματισμού C, ενώ σήμερα έχουν εμπλακεί πολλοί ερευνητές με στόχο την προσθήκη νέων λειτουργιών και την βελτίωση των δυνατοτήτων του.

Το όνομα snort προέρχεται από το γεγονός ότι η εφαρμογή είναι ένας ανιχνευτής. Διατίθεται από την Gnu General Public Licence (GPL) η οποία καθιστά ελεύθερη την χρήση και ανάπτυξη του κώδικα από τον καθένα και μπορεί κάποιος να κατεβάσει την εφαρμογή δωρεάν μέσω της ιστοσελίδας www.snort.org. Είναι μια εφαρμογή η οποία έχει αναπτυχθεί με τέτοιο τρόπο ώστε να λειτουργήσει με ένα ευρύ φάσμα λειτουργικών συστημάτων.

Η εφαρμογή έχει μεγάλη πιθανότητα ανίχνευσης μιας επίθεσης η οποία μπορεί να συμβαίνει σε ένα δίκτυο και ο χρήστης μπορεί να ρυθμίσει με μεγάλη ευελιξία τις επιθέσεις που θα ανιχνεύσει καθώς και τον τρόπο που θα παρουσιάσει τα αποτελέσματα του.

Για την λειτουργία της εφαρμογής σε λειτουργικό σύστημα Linux είναι αναγκαία η ύπαρξη από τις βιβλιοθήκες libpcap, pcre, libnet, Barnyard ενώ σε περιβάλλον Windows είναι αναγκαίες οι βιβλιοθήκες WinPcap και το Barnyard.

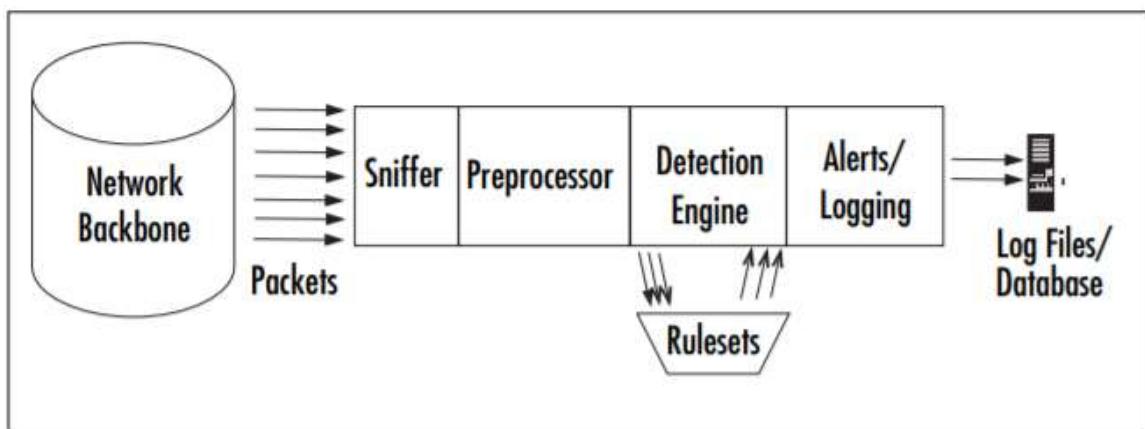
Η τοποθέτηση του snort ανάλογα με την τοπολογία του δικτύου μπορεί να γίνει σε πολλά σημεία. Τα σημαντικότερα σημεία όμως είναι αυτά είναι τα σημεία όπου μπαίνει και βγαίνει η κίνηση.



Αρχιτεκτονική του snort

Η αρχιτεκτονική του snort αποτελείται από τέσσερα βασικά στοιχεία.

1. Το συλλέκτη πακέτων(sniffer), ο οποίος συλλέγει όλα τα διακινούμενα πακέτα.
2. Τους προ επεξεργαστές (preprocessors), που κανονικοποιούν την κίνηση και ανιχνεύουν ύποπτες δραστηριότητες οι οποίες δεν μπορούν να ανιχνευθούν μέσω υπογραφών.
3. Τη μηχανή ανίχνευσης (detection engine), που ανιχνεύει ύποπτες δραστηριότητες βάση υπογραφών
4. Τη έξοδο (output) , η οποία ανάλογα με τις ρυθμίσεις του συστήματος μπορεί να είναι η αποστολή ειδοποίησεων για τις ύποπτες δραστηριότητες ή και η καταγραφή της παρατηρούμενης δικτυακής κίνησης



Αρχιτεκτονική του snort [1]

Οι προ επεξεργαστές, η μηχανή ανίχνευσης και οι έξοδοι του Snort είναι πρόσθετα στοιχεία (plugins). Σε προγενέστερες εκδόσεις του Snort ήταν ενσωματωμένα στον πυρήνα του συστήματος, αλλά στην πορεία διαχωρίστηκαν προκειμένου να είναι ευκολότερη η τροποποίηση τόσο του πυρήνα του συστήματος όσο και του κάθε πρόσθετου στοιχείου ξεχωριστά.

Ανιχνευτής(Sniffer): Αποτελεί το πρώτο υποσύστημα που συμμετέχει στην επεξεργασία των πακέτων. Ξεκινάει με την ανίχνευση των πακέτων τα οποία πρέπει να τα αποκωδικοποίησει. Αποτελείται από μια σειρά από αποκωδικοποιητές ο καθένας από τους οποίους αποκωδικοποιεί συγκεκριμένα στοιχεία από κάθε πρωτόκολλο. Επίσης προσθέτει τις επικεφαλίδες για κάθε επίπεδο, ethernet header, IP header, TCP Header, payload.

Προ επεξεργαστές(Preprocessor): Οι προ επεξεργαστές υποστηρίζουν ανίχνευση για δραστηριότητες οι οποίες δεν μπορούν να υλοποιηθούν από τους πρότυπους κανόνες του Snort. Ο χρήστης μπορεί να υλοποιήσει τους δικούς του και να τους συμπεριλάβει εύκολα στο Snort. Σχετίζονται είτε με την εξαγωγή στατιστικών που αφορούν πακέτα που επεξεργάζεται το Snort, είτε με την προετοιμασία τους πριν αυτά καταλήξουν στο επόμενο υποσύστημα, το Decode engine.



Σύστημα Ανίχνευσης (Detection Engine): Η μηχανή ανίχνευσης εξετάζει τα πακέτα και τα ελέγχει σχετικά με τους κανόνες του snort για να φανεί αν συμβαίνει αν συμβαίνει επίθεση. Αποτελείται από δύο φάσης. Στην πρώτη φάση, η οποία εκτελείται στην εκκίνηση της εφαρμογής, διαβάζονται οι κανόνες και οργανώνονται για περαιτέρω επεξεργασία. Η δεύτερη φάση εκτελείται όταν το snort εντοπίσει ένα πακέτο να κινείται στο δίκτυο και αφού περάσει το στάδιο Decode Engine. Σε αυτό το σημείο ελέγχεται το πακέτο σε σχέση με τους καταχωρημένους κανόνες και εντοπίζεται αν εκτελείται κάποια επίθεση.

Output Engine: Αποθηκεύει το πακέτο με διάφορους τρόπους και επισημαίνει τις ειδοποιήσεις που προκύπτουν από το Snort.

Οδηγίες Εγκατάστασης

Εγκαταστήστε το λογισμικό ανίχνευσης εισβολών snort. Στη συνέχεια εκτελέστε τις ακόλουθες λειτουργίες:

```
File Machine View Input Devices Help
root@kwstas-VirtualBox: ~
root@kwstas-VirtualBox:~# apt-get install snort
```

Θα κάνουμε εγκατάσταση του snort σε ένα μηχάνημα kali Linux
Με την εντολή **apt-get install snort**

```
File Machine View Input Devices Help
root@kwstas-VirtualBox: ~
root@kwstas-VirtualBox:~# snort -V
[*]-> Snort! <*-  
Version 2.9.7.0 GRE (Build 149)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.7.4  
Using PCRE version: 8.38 2015-11-23  
Using ZLIB version: 1.2.8
root@kwstas-VirtualBox:~#
```

Βλέπουμε την έκδοση του snort που κάναμε εγκατάσταση με την εντολή **snort -V**.



Όπως βλέπουμε έχουμε την έκδοση snort 2.9.7.0.5
Εκτέλεση το snort σαν network Ids

Για να εκτελεστεί το Snort σε network Ids θα πρέπει να δοθεί η παράμετρος -c, η τιμή της οποίας θα είναι το configuration αρχείο του Snort (συνήθως το snort.conf), το οποίο περιέχει τις ρυθμίσεις που θα καθορίσουν τον τρόπο λειτουργίας του Snort.

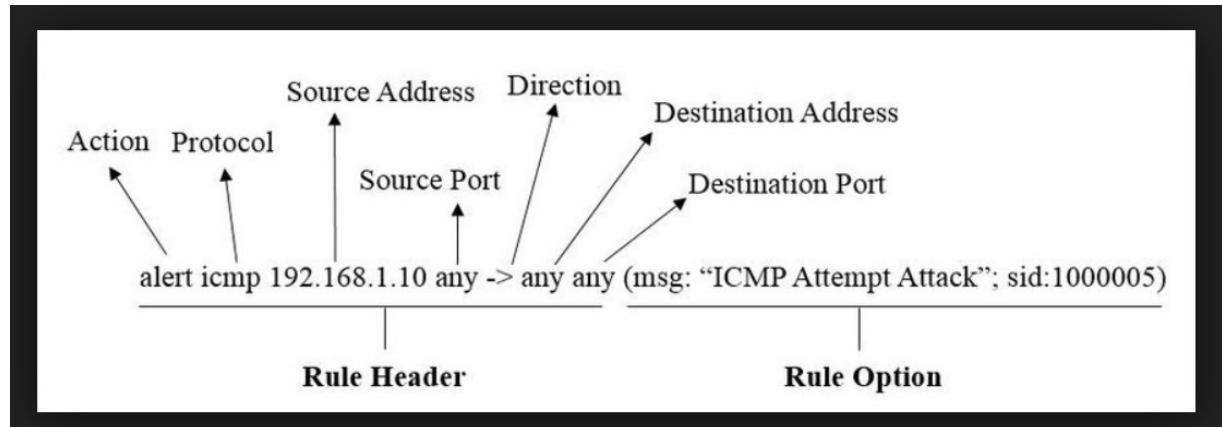
```
root@kwstas-VirtualBox:/etc/snort/rules
root@kwstas-VirtualBox:/etc/snort/rules# snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s8
```

snort -c /etc/snort/snort.conf

Κανόνες (rules) του Snort

Όπως αναφέρθηκε και παραπάνω οι κανόνες του Snort περιγράφουν τα χαρακτηριστικά ενός πακέτου καθώς και την ενέργεια που θα εκτελεστεί σε περίπτωση που θα βρεθεί πακέτο που να ικανοποιεί όλες τις συνθήκες του κανόνα. Μπορούν να γράφουν σε ASCII μορφή και χωρίζονται σε δύο λογικά μέρη το Rule Header και τα Rule Options.

Rule Header: Περιλαμβάνει το πρωτόκολλο, την IP διεύθυνση πηγής και προορισμού, πληροφορίες για την θύρα πηγής και προορισμού καθώς και την ενέργεια του κάθε κανόνα.



Rule Header: Περιλαμβάνει το πρωτόκολλο, την IP διεύθυνση πηγής και προορισμού, πληροφορίες, για την θύρα πηγής και προορισμού καθώς και την ενέργεια του κάθε κανόνα (action).

Η ενέργεια του κάθε κανόνα (**action**) είναι η ενέργεια που θα εκτελεστεί όταν κάποιο πακέτο ταιριάζει με αυτό που περιγράφεται στον κανόνα και μπορεί να είναι μια από τις πέντε που ακολουθούν, ενώ επίσης μπορεί να ορίσει και τους δικούς του τύπους ο κάθε χρήστης.

- Alert, δημιουργεί μια ειδοποίηση και στη συνέχεια καταγράφει το πακέτο. Αποτελεί τον τρόπο με τον οποίο επισημαίνει την ανίχνευση της επίθεσης.
- Log, καταγράφει το πακέτο στον δίσκο.
- Pass, επιτρέπεται να περάσει το πακέτο



- Activate, δημιουργεί ένα alert και ενεργοποιεί ένα Dynamic Rule
- Dynamic, ενεργοποιείται από ένα activate Rule και ενεργεί σαν log Rule.

Πρωτόκολλο (Protocol): Το είδος του πρωτοκόλλου στο οποίο ανήκει κάθε πακέτο. Μπορεί να είναι tcp, udp, icmp, ip.

IP πηγής (Source IP): Η IP διεύθυνση του αποστολέα που βρίσκεται στο header του πακέτου.

Θύρα Πηγής (Source Port): Η πόρτα προορισμού του πακέτου.

Rule Options (Ρυθμίσεις των κανόνων): Περιλαμβάνουν πληροφορία για τα χαρακτηριστικά στα οποία θα ελέγχει το πακέτο.

- Option Keyword: Το λεκτικό που υποδηλώνει το όνομα-είδος τις επιλογής.
- Option Argument: οι παράμετροι που δέχεται το option σε σχέση με τις οποίες θα ελεγχθεί το πακέτο.

Υπάρχουν τέσσερις κατηγορίες στις οποίες χωρίζονται οι ρυθμίσεις των κανόνων.

Βοηθητικές: Παρέχουν επιπλέον πληροφορίες για τον κανόνα χωρίς να εμπλέκονται στην αναζήτηση. Λεκτικά με την εντολή msg το οποίο καθορίζει το μήνυμα που θα εμφανιστεί μαζί με το περιεχόμενο του πακέτου, το reference που επιτρέπει στον κανόνα να αναφερθεί σε ένα εξωτερικό σύστημα αναγνώρισης επιθέσεων. Ως reference μπορεί να απεικονιστεί και ένα μοναδικό url. Επίσης, ένα άλλο λεκτικό είναι το sid το οποίο χρησιμοποιείται ως αναγνωριστικό κλειδί ανάμεσα στους κανόνες σε συνδυασμό με το rev, το οποίο αντιστοιχεί στον αριθμό αναθεώρησης (revision number) και επιτρέπει την ανανέωση των κανόνων με νέες πληροφορίες. Άκομα ένα λεκτικό είναι το classtype το οποίο χρησιμοποιείται στην ταξινόμηση των κανόνων σε διαφορετικές τάξεις, όπου κάθε τάξη αντιστοιχεί σε ένα είδος επίθεσης στο σύστημα με συγκεκριμένη προτεραιότητα και τέλος είναι το priority το οποίο αναθέτει μια προτεραιότητα στον κάθε κανόνα. Όσο πιο μικρή η τιμή του πεδίου αυτού τόσο πιο σημαντικός ο κανόνας.

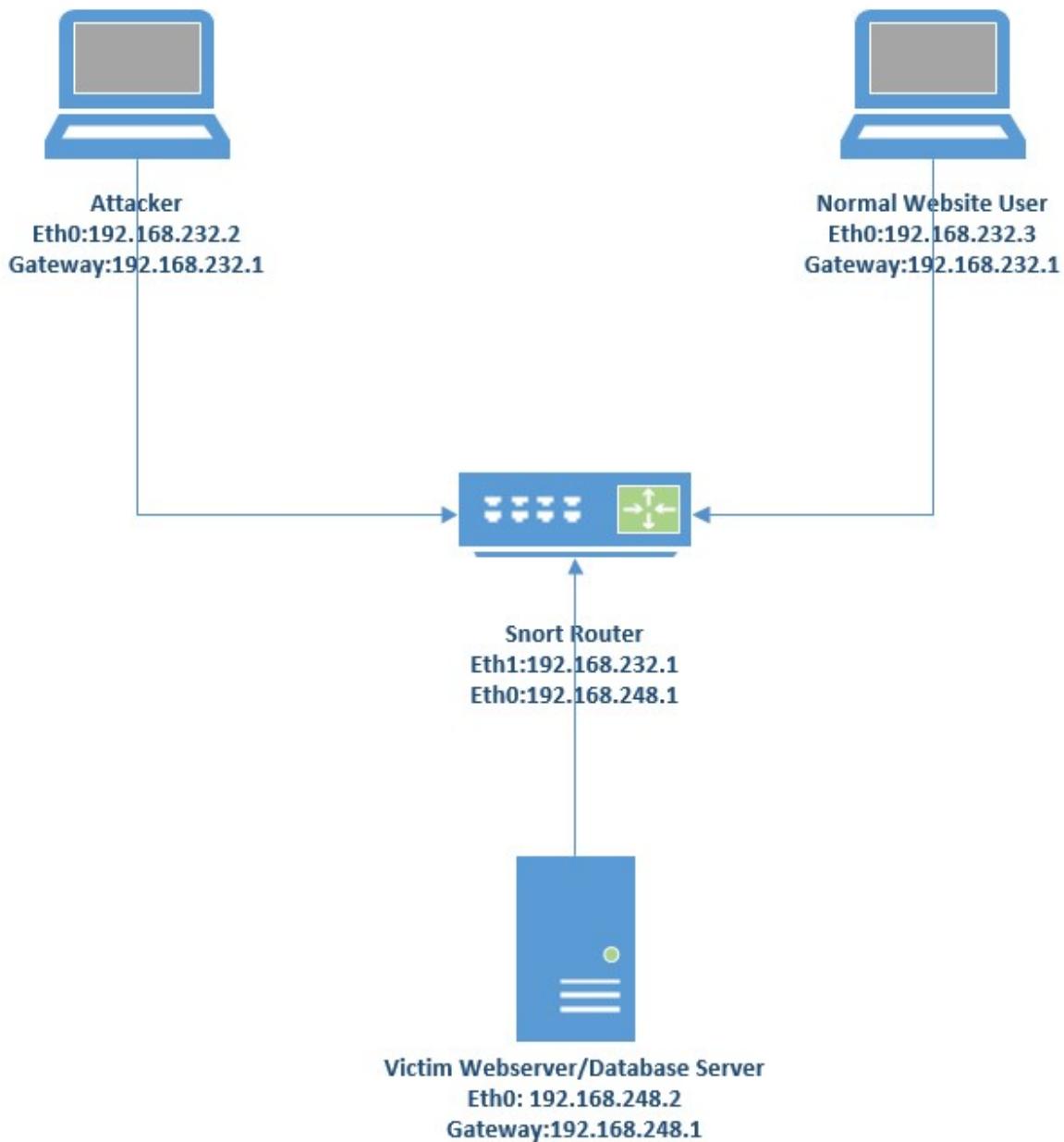
Περιεχομένου: Απεικονίζουν δεδομένα που θα αναζητηθούν στο περιεχόμενο του πακέτου. Το πιο σημαντικό αυτής της ρύθμισης είναι το λεκτικό content το οποίο είναι από τα πιο σημαντικό χαρακτηριστικά του Snort αφού επιτρέπει τον ορισμό κανόνων από τον χρήστη. Στους κανόνες ο χρήστης αναζητά συγκεκριμένες συμβολοσειρές στο περιεχόμενο (payload) του πακέτου. Όταν υπάρχει η επιλογή content εκτελείται πρώτα ένας έλεγχος της συμβολοσειράς του πακέτου με τον αλγόριθμο αναζήτησης Boyer-Moore και στην συνέχεια εξετάζονται και οι υπόλοιπες παράμετροι. Το όρισμα μπορεί να δοθεί ως απλή συμβολοσειρά ή σε δεκαεξαδική μορφή με το διαχωριστικό " | ".

Μη-Περιεχομένου: Παρέχουν πληροφορίες εκτός του περιεχομένου (payload) του πακέτου. Λεκτικά που χρησιμοποιούνται είναι το fragoffset, το tll, το tos, το id, το ipopts, το Fragbits, το dsizes, το flags, το flow, το seq, το ack, το window, το itype, το icode, το icmp_id, το rpc, το ip_proto, το same_ip.s

Ολοκληρωμένης αναζήτησης: Παρέχουν την εφαρμογή συγκεκριμένων γεγονότων όταν επαληθευτεί ο κανόνας. Εδώ χρησιμοποιούνται τα λεκτικά logto, session, react και tag.



2)Καθορισμός της Πολιτικής Ανίχνευσης Εισβολών. Σε αυτό το βήμα θα περιγράψετε την Πολιτική Ανίχνευσης Εισβολών Δικτύου που θα πρέπει να ικανοποιείται για το υπό έλεγχο δίκτυο. Θα πρέπει να περιλαμβάνει κατ' ελάχιστο την περιγραφή των υπηρεσιών που πρέπει να ελέγχονται, το επίπεδο ελέγχου, τους ελάχιστους ελέγχους εισβολών ανά προστατευόμενη υπηρεσία ή εσωτερικό δίκτυο, το επίπεδο καταγραφής, τη διαδικασία ελέγχου των αρχείων καταγραφής, τη διαδικασία ανανέωσης των κανόνων και της πολιτικής κτλ. (Υπόδειξη: Έμφαση θα πρέπει να δοθεί στον καθορισμό της πολιτικής ανίχνευσης εισβολών ώστε να είναι σαφής και καλά ορισμένη η πολιτική πρόσβασης για κάθε υπηρεσία ή εσωτερική περιοχή δικτύου που πρέπει να προστατευθεί. Η υλοποίηση που θα πραγματοποιηθεί στο Β' μέρος της εργασίας θα πρέπει να ακολουθεί την πολιτική αυτή) (3 μονάδες).



Για την εργασία θα δημιουργήσουμε ένα μικρό δίκτυο στο οποίο θα γίνει η εγκατάσταση του ανιχνευτή Snort.

NIDS (NETWORK INTRUSION DETECTION SYSTEM)

Περιβάλλον Δοκιμής

Όλη η προσομοίωση θα πραγματοποιηθεί σε ένα εικονικό περιβάλλον, το εικονικό περιβάλλον που θα χρησιμοποιηθεί είναι το **Vmware Workstation pro 2014**



Το περιβάλλον που θα χρησιμοποιηθεί για το εσωτερικό και εξωτερικό δίκτυο το οποίο θα αποτελείτε από τους απλούς χρήστες καθώς και το router από το οποίο θα περνάει η κίνηση θα χρησιμοποιήσουμε για όλα το λειτουργικό σύστημα τα **ubuntu-16.04.1-desktop-amd64**.

Για αυτόν που θα πραγματοποιεί τις επιθέσεις (**attacker**) θα χρησιμοποιήσουμε τα **kali-linux-2019.1a-i386**.

Δημιουργία Μηχανημάτων Σε Vmware

- **Snort-Router** - ubuntu-16.04.1-desktop-amd64. Το μηχάνημα θα παίζει τον ρόλο του router στο οποίο θα εγκαταστήσουμε το snort και από εκεί θα περνάει η κίνηση από το εξωτερικό δίκτυο στο εσωτερικό.
- **Victim** WebServer/Database - ubuntu-16.04.1-desktop-amd64. Το μηχάνημα μας θα βρίσκεται στο εσωτερικού του δικτύου και θα περιέχει έναν webserver στον οποίο θα είναι δυνατό να συνδεθεί κάποιος μέσα από το εσωτερικό δίκτυο.
- **Attacker** - **kali-linux-2019.1a-i386** θα δημιουργήσουμε ένα μηχάνημα το οποίο θα πραγματοποιήσει κάποιες επιθέσεις με σκοπό την επαλήθευση των κανόνων που θα εφαρμόσουμε στον ανιχνευτή snort

Παραπάνω έχουμε ένα σχέδιο με τα ονόματα του δικτύου καθώς και τις ip διευθύνσεις που θα έχει το κάθε μηχάνημα. Το μηχάνημα **Snort-Router** δεν θα έχει τις default gateway ip διευθύνσεις για να λειτουργεί σαν το router του δικτύου.

Θα χρειαστεί να δημιουργήσουμε δύο εικονικά δίκτυα

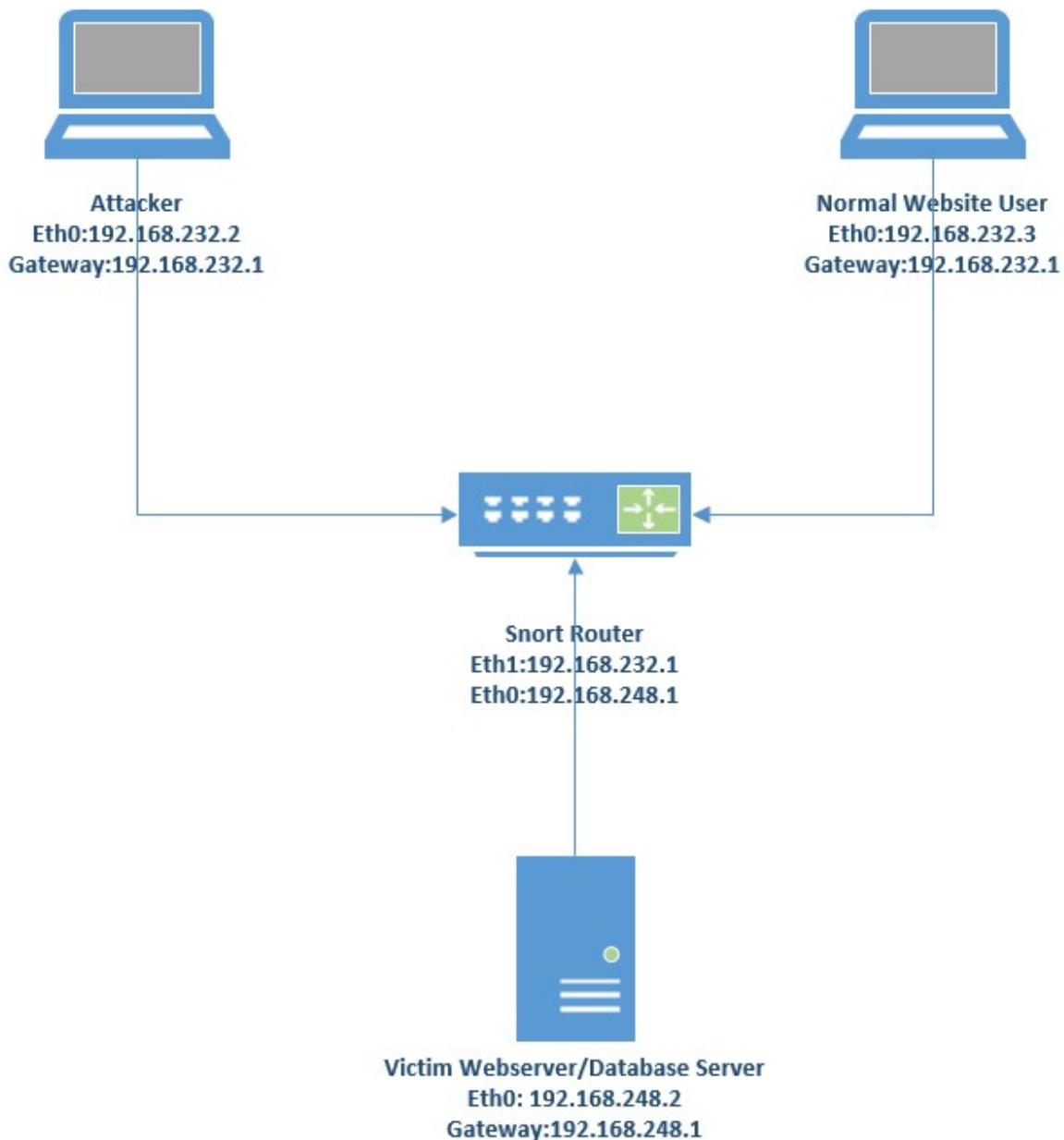
- Vmnet1 – Host only network with Subnet 192.168.248.0/24 **Εικονικό Εσωτερικό Δίκτυο**
- Vmnet2 – Host only network with Subnet 192.168.232.0/24 **Εικονικό Εξωτερικό Δίκτυο**

Ip διευθύνσεις που θα δοθούν:

Snort-Router – ip1:192.168.248.1, ip2:192.168.232.1, nat network για επικοινωνία με κανονικό internet

Victim – ip:192.168.248.2 (Το μηχάνημα αυτό δεν θα έχει nat καθώς δεν θα έχει την δυνατότητα να συνδεθεί στο internet θα είναι μέρος του εσωτερικού δικτύου θα επικοινωνεί με το router με το εξωτερικό δίκτυο)

Attacker – ip:192.168.232.2, nat network για σύνδεση στο internet.



Snort Nids (Network intrusion Detection System)

Victim (Εσωτερικό Δίκτυο)

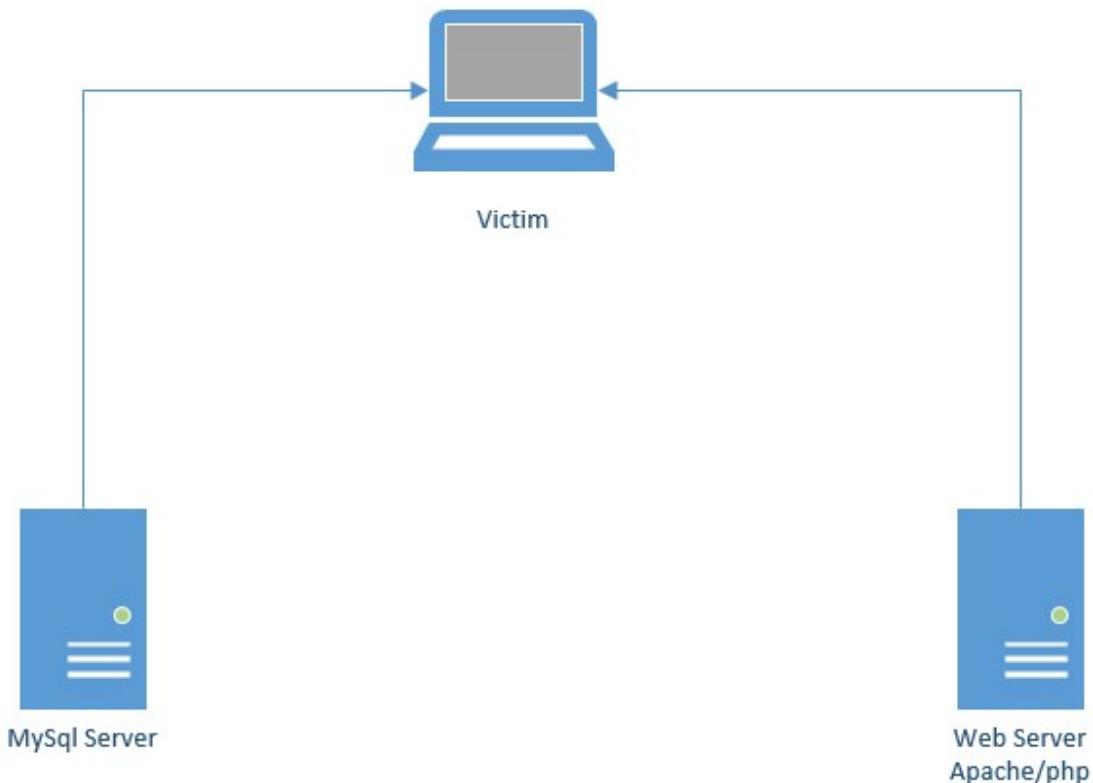
Webservers - Victim

Τι θα περιέχει το Εσωτερικό Δίκτυο

- **Apache2 WebServer**
 - Enable Http Basic Authentication



- Ενεργοποίηση της HTTP βασικής αυθεντικοποίησης (HTTP BASIC AUTHENTICATION)
- Δημιουργία από http διαπιστευτήρια (http credentials Username=web και password=xxx)
- **Web Application Server DVWA** δημιουργία ιστοσελίδας με την βοήθεια έτοιμου πλαισίου στο οποίο θα μπορεί να συνδέονται εξωτερικοί χρήστες μέσω του εξωτερικού δικτύου. Για την ιστοσελίδα θα χρησιμοποιήσουμε το **DVWA damn vulnerable web application** η οποίο είναι μια **php/mysql web application**. Θα χρησιμοποιήσουμε αυτήν την ιστοσελίδα για να μπορέσουμε να πραγματοποιήσουμε επιθέσεις με τον Attacker ο οποίος και θα συνδέεται στην ιστοσελίδα.
- **Php7.3**
- **MySQL Server**



Τι θα χρησιμοποιηθεί για την ιστοσελίδα

Snort



Θα χρησιμοποιήσουμε το snort για να ελέγχουμε τα παρακάτω Web Server

- Mysql Server
 - Ειδοποιήσεις για πιθανές Sql επιθέσεις
 - Ειδοποιήσεις για πιθανές επιθέσεις σύνδεσης

Ειδοποιήσεις σε Host scans

Ειδοποιήσεις σε Port scans

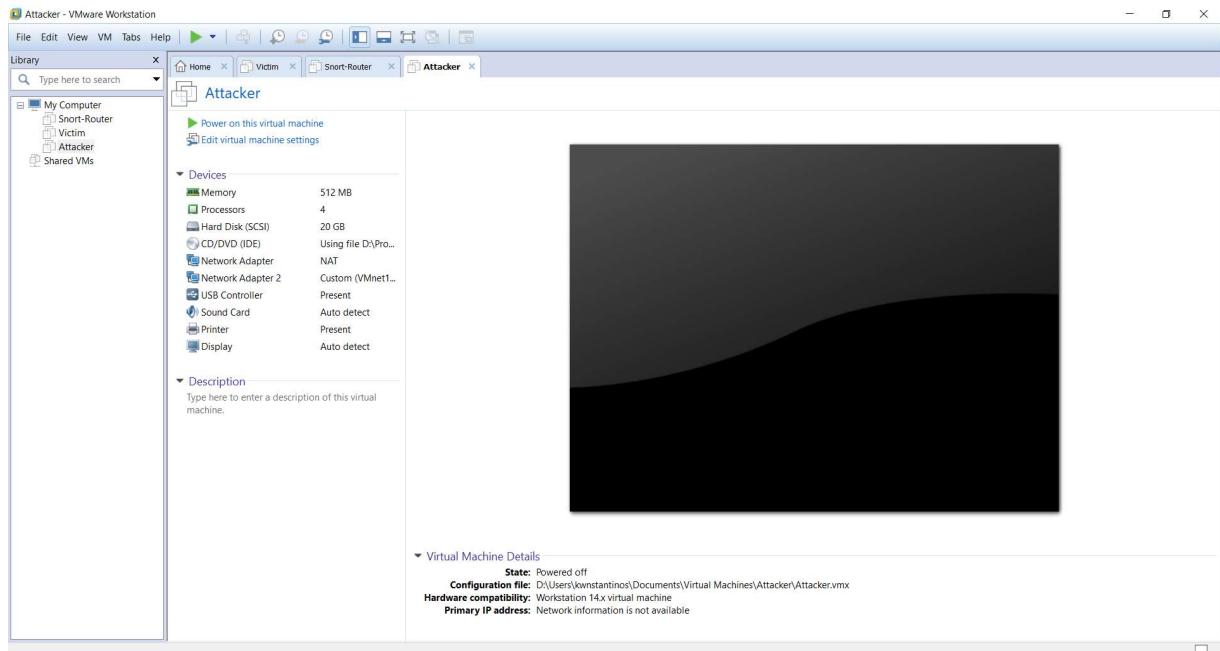
Ειδοποιήσεις σε Cross Site Scripting

Ειδοποιήσεις σε επιθέσεις Denial of Service



Β) Υλοποίηση και έλεγχος εφαρμογής της πολιτικής ανίχνευσης εισβολών (5 μονάδες)

1. Διαμόρφωση συστήματος ανίχνευσης εισβολών (IDS). Να διαμορφώσετε κατάλληλα το σύστημα snort ώστε να υλοποιεί την Πολιτική Ανίχνευσης Δικτυακών Εισβολών, όπως ορίστηκε στο Α μέρος. Να δημιουργηθεί και να παραδοθεί αναλυτικό εγχειρίδιο εγκατάστασης και παραμετροποίησης του λογισμικού. (3 μονάδες)



Αρχικά θα ορίσουμε το δίκτυο μας



Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.131.0
VMnet2	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.40.0
VMnet11	Custom	-	-	-	192.168.248.0
VMnet12	Custom	-	-	-	192.168.232.0

Add Network... Remove Network... Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)
Bridged to:

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet11

Use local DHCP service to distribute IP address to VMs

Subnet IP: Subnet mask:

⚠ Administrator privileges are required to modify the network configuration.

Restore Defaults

Το πρώτο δίκτυο που έχουμε θα είναι το 192.168.248.0 που θα λειτουργεί σαν εσωτερικό δίκτυο.



Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.131.0
VMnet2	Host-only	-	Connected	Enabled	192.168.226.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.40.0
VMnet11	Custom	-	-	-	192.168.248.0
VMnet12	Custom	-	-	-	192.168.232.0

Add Network... Remove Network... Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to:

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet12

Use local DHCP service to distribute IP address to VMs

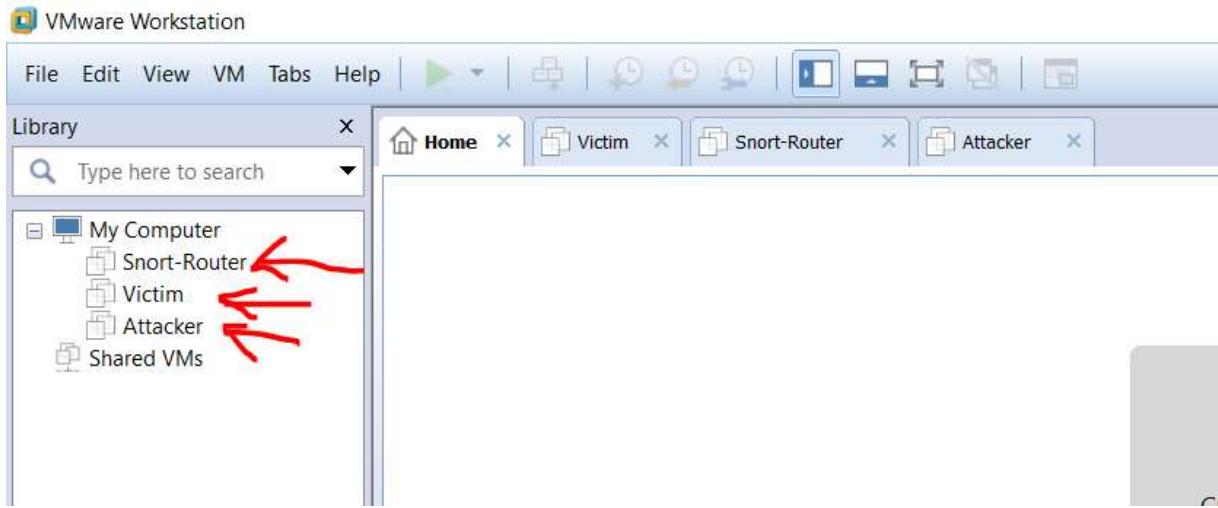
Subnet IP: Subnet mask:

! Administrator privileges are required to modify the network configuration.

Restore Defaults

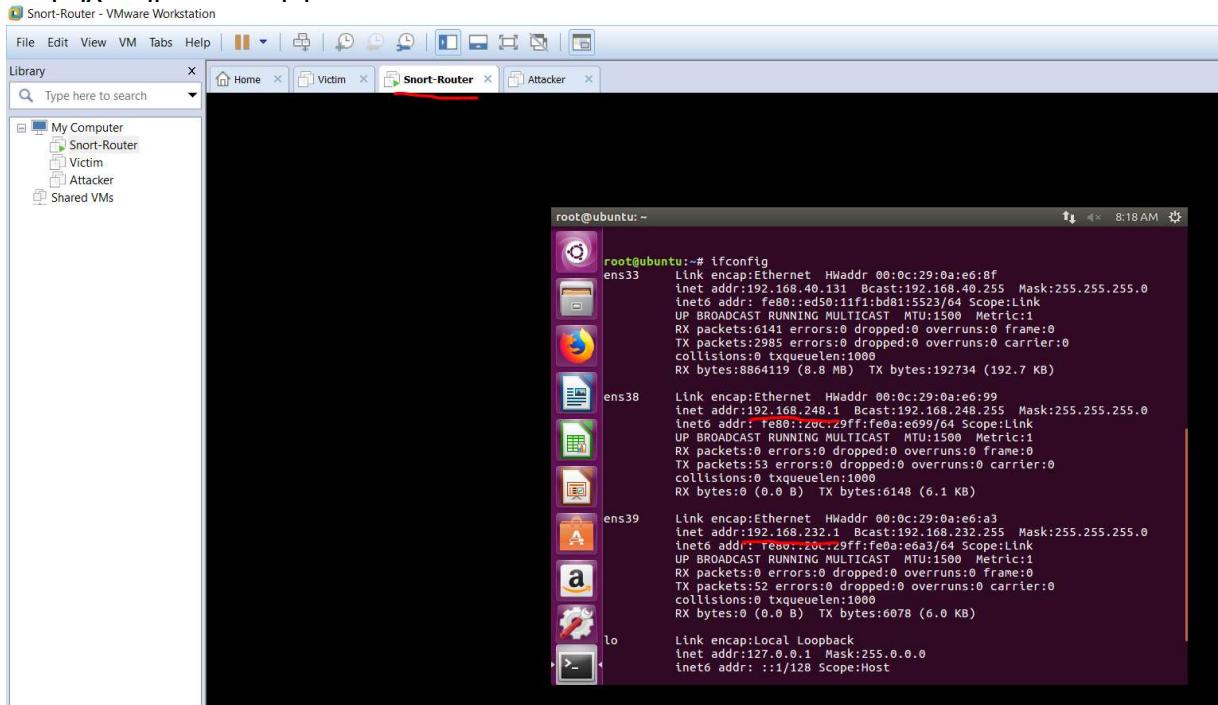
Το πρώτο δίκτυο που έχουμε θα είναι το 192.168.232.0 που θα λειτουργεί σαν εξωτερικό δίκτυο.

Στην συνέχεια θα δημιουργήσουμε τα μηχανήματα μας με τις αντίστοιχες ιρ όπως ορίσαμε στο προηγούμενο ερώτημα.



Snort – Router

Το μηχάνημα λειτουργεί σε ubuntu



Όπως βλέπουμε έχουμε ορίσει τις δυο ίρ διευθύνσεις
Ip: 192.168.248.1
Ip: 192.168.232.1
Και έχουμε και μια nat ip: 192.168.40.131 η οποία λειτουργεί σαν internet



Για να ρυθμίσουμε τις ίρ πηγαίνουμε στο path **/etc/networks** και με την εντολή **gedit Interfaces** και δηλώνουμε τις ίρ μας.

```
@ubuntu: /etc/network
root@ubuntu:/etc/network# gedit interfaces

interfaces (/etc/network) - gedit
Open ▾  interfaces
Save
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

#interface for internal network containing victim
auto ens38
iface ens38 inet static
address 192.168.248.1
netmask 255.255.255.0

#interface for external network containing attacker
auto ens39
iface ens39 inet static
address 192.168.232.1
netmask 255.255.255.0
```

Όπως βλέπουμε έχουμε δημιουργήσει το περιβάλλον για το εσωτερικό με ip 192.168.248.1 και το εξωτερικό δίκτυο με ip 192.168.232.1

Ενεργοποίηση ip forwarding στον host

- Στο αρχείο `/etc/sysctl.conf` βρίσκουμε τη γραμμή: `net.ipv4.ip_forward=1` και την ενεργοποιούμε.
- Από shell εκτελούμε: `sysctl -p`.

```
host.conf          ppp      zsh_command_not_found
hostname          printcap
root@ubuntu:/etc# gedit sysctl.conf
```



```
sysctl.conf (/etc) - gedit
Open ▾ Save
sysctl.conf
/etc
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
#
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
#
# Uncomment the next line to enable packet forwarding for IPv6
Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS
```

Αφαιρούμε το σχόλιο `#net.ipv4.ip_forward=1`

```
root@ubuntu: /etc#
root@ubuntu:/etc# sysctl -p
net.ipv4.ip_forward = 1
root@ubuntu:/etc#
```

Επιβεβαίωση ότι πραγματοποιήθηκε το προώθηση της ip

Αφού έχουμε ορίσει το δίκτυο στην συνέχεια πάμε να εγκαταστήσουμε τον ανιχνευτή snort.

Πρώτα συνδεόμαστε σαν διαχειριστές για να έχουμε δικαιώματα εγκατάστασης.

```
root@ubuntu: ~
root@ubuntu:~# apt-get install snort
```



```
root@ubuntu:/etc/snort
root@ubuntu:/etc/snort# ls
classification.config    reference.config    snort.debian.conf
community-sid-msg.map    rules               threshold.conf
gen-msg.map                snort.conf         unicode.map
root@ubuntu:/etc/snort#
```

Όπως βλέπουμε έχουν δημιουργηθεί οι φάκελοι για να επιβεβαιώσουμε θα πληκτρολογήσουμε την εντολή snort.

```
root@ubuntu:/etc/snort
root@ubuntu:/etc/snort# snort -q -A console -i ens38 -c /etc/snort/snort.conf
```

Snort -q -A console -i ens38 -c /etc/snort/snort.conf

-q -A console είναι για να εμφανίζονται τα μηνύματα στην οθόνη
-i ens38 επιλέγουμε το δίκτυο που θέλουμε να προστατεύσουμε δηλαδή το εσωτερικό μας δίκτυο 192.168.248.0.

```
root@ubuntu:/etc/snort
root@ubuntu:/etc/snort# ifconfig
ens33      Link encap:Ethernet HWaddr 00:0c:29:0a:e6:8f
            inet addr:192.168.40.131 Bcast:192.168.40.255 Mask:255.255.255.0
            inet6 addr: fe80::ed50:11f1:bd81:5523/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                      RX packets:73863 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:38254 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:109429829 (109.4 MB) TX bytes:2316202 (2.3 MB)

ens38      Link encap:Ethernet HWaddr 00:0c:29:0a:e6:99
            inet addr:192.168.40.131 Bcast:192.168.40.255 Mask:255.255.255.0
            inet6 addr: fe80::20c:29ff:fe0a:e699/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:6924 (6.9 KB)

ens39      Link encap:Ethernet HWaddr 00:0c:29:0a:e6:a3
            inet addr:192.168.232.1 Bcast:192.168.232.255 Mask:255.255.255.0
            inet6 addr: fe80::20c:29ff:fe0a:e6a3/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:0 (0.0 B) TX bytes:6854 (6.8 KB)
```

-c /etc/snort/snort.conf είναι το configuration file από το οποίο θα χρησιμοποιεί ο ανιχνευτής μας το snort.

```
root@ubuntu:/etc/snort
root@ubuntu:/etc/snort# snort -q -A console -i ens38 -c /etc/snort/snort.conf
```



Όπως μπορούμε να δούμε ο ανιχνευτής λειτουργεί κανονικά.
Αφού ορίσουμε το δίκτυο μας και όλους τους Server θα ορίσουμε τους κανόνες μας.



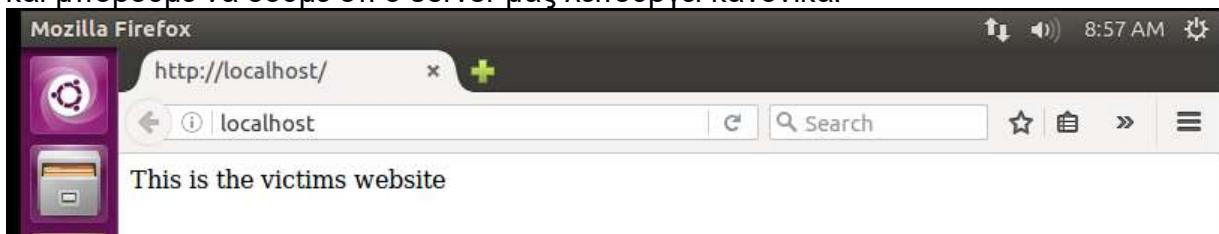
Στην συνέχεια δημιουργούμε το επόμενο χρήστη ο οποίος και θα είναι στο εσωτερικό μας δίκτυο **Victim**.

Το μηχάνημα λειτουργεί σε ubuntu και αρχικά πριν του ορίσω την ip με default gateway θα κατεβάσω τους απαραίτητους servers που θα χρειαστώ.

Apache Server

```
Terminal
root@ubuntu: ~
root@ubuntu:~# apt-get install apache2
```

Εγκατάσταση του apache2 μόλις κάνουμε εγκατάσταση επιλέγουμε localhost στον Firefox και μπορούμε να δούμε ότι ο server μας λειτουργεί κανονικά.



Έχει δημιουργηθεί και μια ιστοσελίδα η οποία μας ενημερώνει σε ποιόν ανήκει η ιστοσελίδα.

```
root@ubuntu: /var/www/html
root@ubuntu:/var/www/html# gedit index.html
```

Η ιστοσελίδα δημιουργήθηκε από το path /var/www/html



This is the victim's website.



HTTP SERVER AUTHORIZATION

Στην συνέχεια θα δημιουργήσουμε όπως ορίστηκε αυθεντικοποίηση http server ζητώντας έτσι από όποιον συνδέεται στην ιστοσελίδα ένα username και Password.

Επιλέγουμε το αντίστοιχο path gedit /etc/apache2/sites-available /

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory "/var/www/html">
        AuthType Basic
        AuthName "Restricted Content"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>
```

Θα το ορίσουμε ως περιορισμένο περιεχόμενο που μόνο όσοι θα δίνουν τα στοιχεία username και password θα μπορούν να συνδεθούν.

Στην συνέχεια θα ορίσουμε το username και το password με το οποίο θα συνδέονται όλοι οι χρήστες στην ιστοσελίδα.

```
root@ubuntu:~$ htpasswd -c etc/apache2/.htpasswd web
```

Username: web

Στην συνέχεια θα ορίσουμε το password.

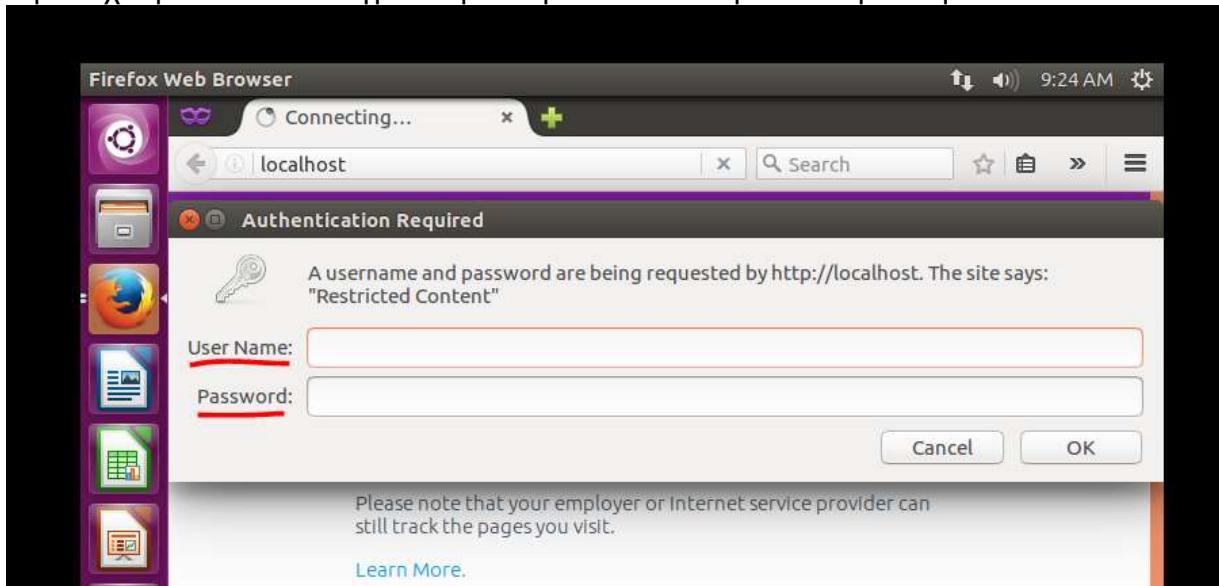


```
root@ubuntu:/# htpasswd -c etc/apache2/.htpasswd web
New password:
Re-type new password:
Adding password for user web
root@ubuntu:/#
```

Για λόγους προσομοίωσης έχουμε ορίσει το password :web

Username: web
password: web

Αφού έχουμε τελειώσει πιηγαίνουμε στην ιστοσελίδα για επαλήθευση.



Όπως μπορούμε να δούμε για να συνδεθεί κάποιος πρέπει να εισάγει τα στοιχεία username και password του http server



Εγκατάσταση Web Server στο Victim

Επειδή θα χρειαστεί στο επόμενο ερώτημα να πραγματοποιηθούν επίθεσης για επαλήθευση ότι το σύστημα snort λειτουργεί κανονικά. Θα επιλέξουμε να δημιουργήσουμε μια web εφαρμογή η οποία είναι κατάλληλη για τέτοιες επιθέσεις.

DVWA Damn Vulnerable Web Application

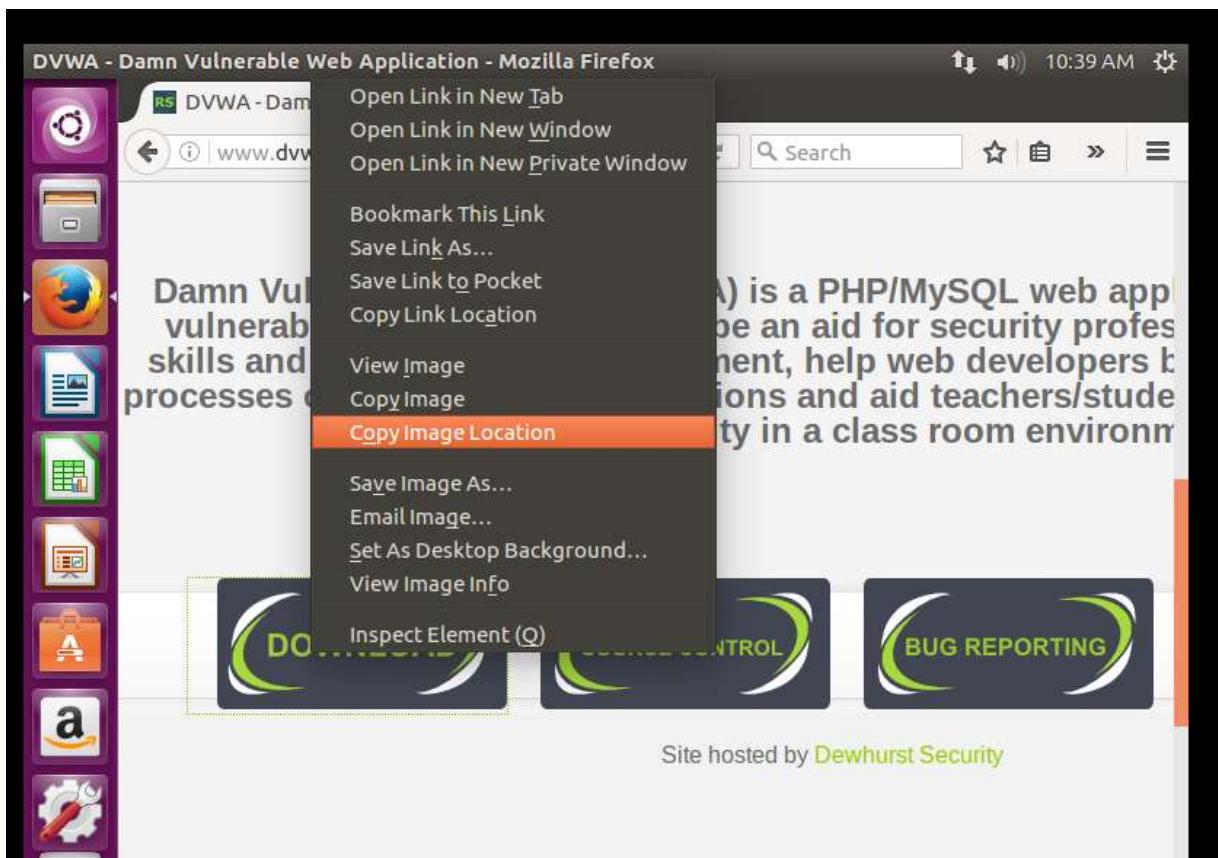
The screenshot shows a browser window with the title bar "DVWA - Damn Vulnerable Web App". The address bar shows "Not secure | www.dvwa.co.uk". The main content area displays the DVWA logo and the title "Damn Vulnerable Web Application (DVWA)". A sidebar on the left lists various attack modules: Brute Force, Command Execution, CSRF, File inclusion, SQL injection, XSS injection (Blind), Union, XXE reflected, Blind, DVWA Security, PHP info, and About. A "Logout" link is also present. The central panel contains a "Welcome to Damn Vulnerable Web App!" message, a "WARNING!" section, and a "Disclaimer". A "Help" button is at the bottom. The footer indicates "Damn Vulnerable Web Application (DVWA) v1.0.2".

Έχουμε την δυνατότητα να κατεβάσουμε την εφαρμογή στο ubuntu μηχάνημα μας και να την χρησιμοποιήσουμε σαν μια web application η οποία θα περιέχει την Mysql Database Sever και την Php.

Στην συνέχεια θα κάνουμε εγκατάσταση τις ιστοσελίδας στο μηχάνημα μας και θα είναι δική μας ιστοσελίδα με την δική μας ip.

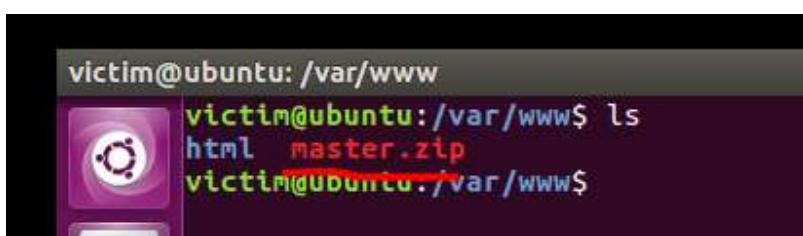
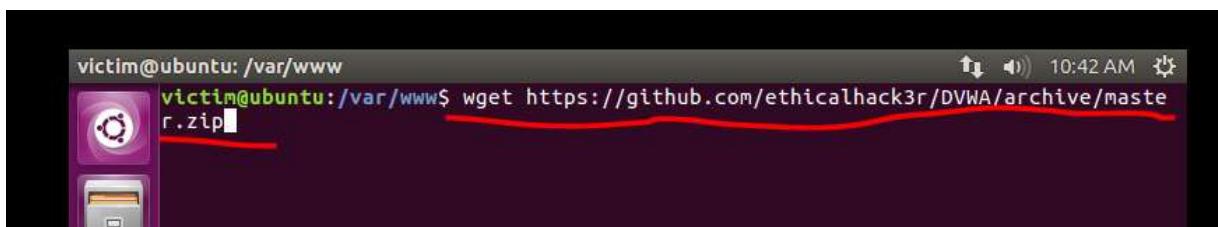
The screenshot shows a Mozilla Firefox browser window titled "DVWA - Damn Vulnerable Web Application - Mozilla Firefox". The address bar shows "www.dvwa.co.uk". The DVWA logo is visible on the left side of the page. The main content area displays the DVWA homepage with the same layout and information as the previous screenshot.

Πηγαίνουμε στην επίσημη ιστοσελίδα www.dvwa.co.uk και επιλέγουμε με δεξί κλικ save address και στην πηγαίνουμε στο command line και κάνουμε download το zip αρχείο master.zip



Με

Wget <https://github.com/ethicalhack3r/DVWA/archive/master.zip>



Κάνουμε unzip το αρχείο μέσα στο html αρχείο.

Χρειάζεται να κάνουμε μια τροποποίηση της ονομασίας ενός configuration καθώς θα έχει θέμα η ιστοσελίδα μας.

Πάμε στο path var/www/html/dvwa/config και κάνουμε rename το αρχείο μας σε config.inc.php



```
victim@ubuntu: /var/www/html/dvwa/config
victim@ubuntu: /var/www/html/dvwa/config$ ls
config.inc.php
victim@ubuntu: /var/www/html/dvwa/config$
```

Στην συνέχεια θα κάνουμε install Mysql Server και Php

```
victim@ubuntu: /
victim@ubuntu: $ apt-get install mysql
```

Ορίζουμε ένα κωδικό στην βάση μας τον οποίο και ορίζουμε και στο configuration file tou dvwa.

```
config.inc.php (/var/www/html/dvwa/config) - gedit
Open ▾ config.inc.php
/var/www/html/dvwa/config
Save
<?php
#
# If you are having problems connecting to the MySQL database and all of the
variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a
problem due to sockets.
#   Thanks to @digininja for the fix.

#
# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

#
# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED
during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a
dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = 'pappoulis13';

#
# Only used with PostgreSQL/PGSQL database selection.
$_DVWA[ 'db_port' ] = '5432';

#
# ReCAPTCHA settings
```

Ελέγχουμε ότι η βάση μας λειτουργεί κανονικά με την εντολή
mysql -u root -p και πληκτρολογούμε τον κωδικό



```
root@ubuntu:/var/www/html/dvwa# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.26-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2019, 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> 
```

Η βάση μας λειτουργεί κανονικά.

Στην συνέχεια κάνουμε εγκατάσταση της Php

```
root@ubuntu:/var/www/html/dvwa
root@ubuntu:/var/www/html/dvwa# apt-get install php 
```

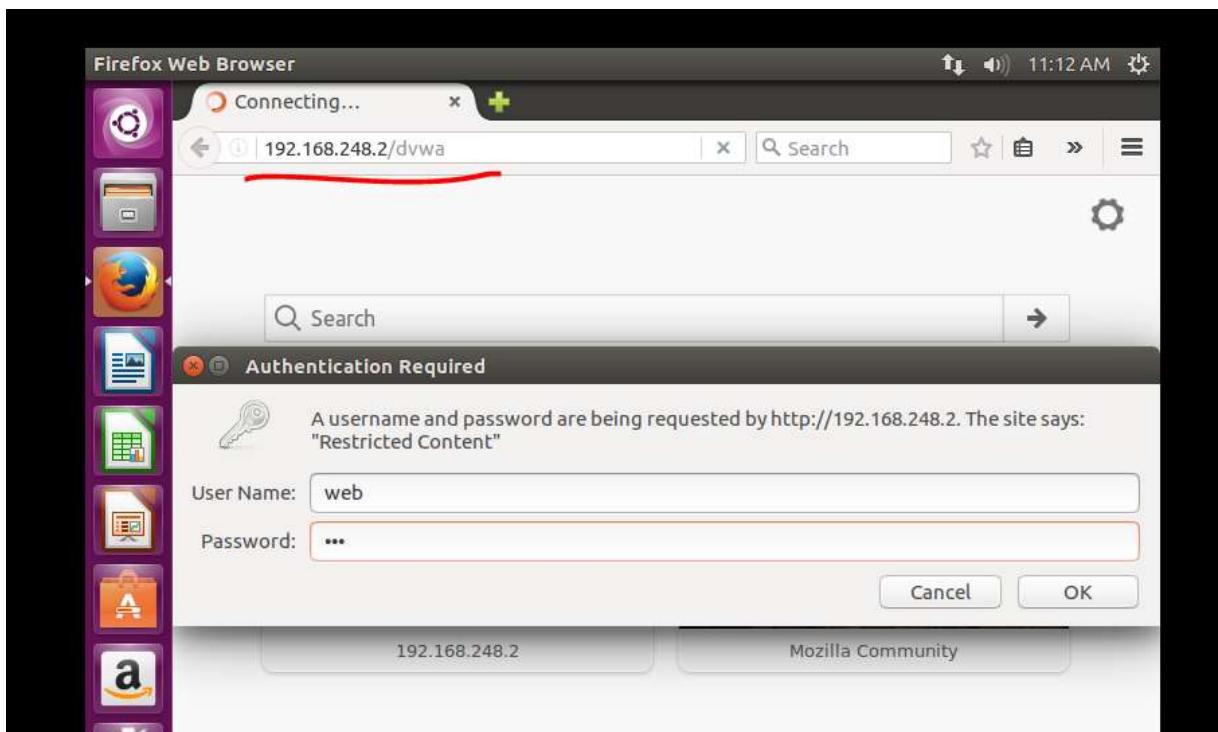
Apt-get install php

Αφού κάνουμε εγκατάσταση επιλέγουμε php -v

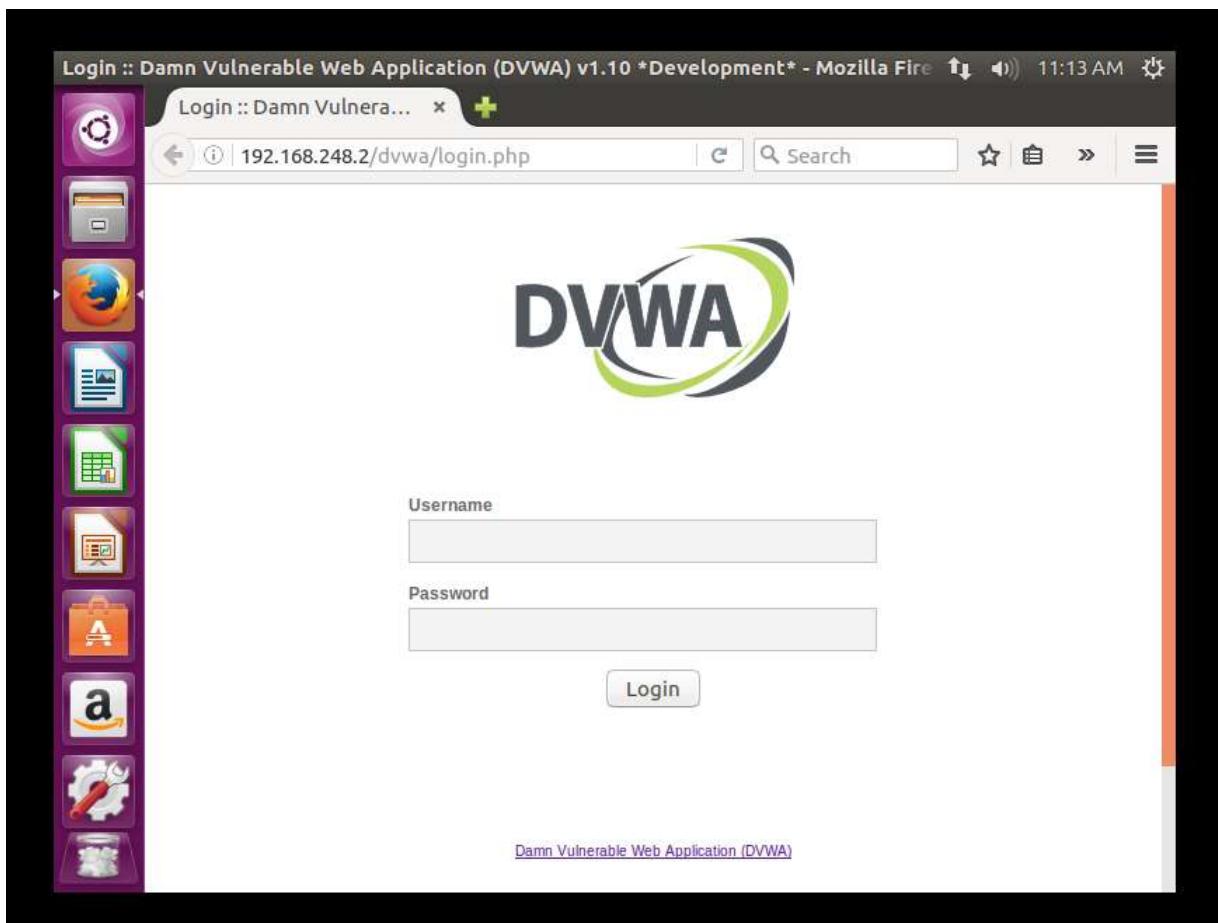
```
root@ubuntu:/var/www/html/dvwa
root@ubuntu:/var/www/html/dvwa# php -v
PHP 5.6.40-8+ubuntu16.04.1+deb.sury.org+1 (cli)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
root@ubuntu:/var/www/html/dvwa# 
```

PHP 5.6.40

Στην συνέχεια πηγαίνουμε στην ιστοσελίδα dvwa με το να πληκτρολογήσουμε στον firefox την ίρ που έχει το μηχάνημα και το dvwa και θα δούμε ότι μπορούμε να συνδεθούμε στην ιστοσελίδα.



Πληκτρολογούμε όνομα χρήστη και κωδικό και συνδέομαι.



Όπως βλέπουμε η ιστοσελίδα είναι λειτουργική

Username: admin

Password: password

Username

Password



The screenshot shows the DVWA homepage. The URL in the address bar is 192.168.248.2/dvwa/index.php. The DVWA logo is at the top right. A sidebar on the left lists various attack modules: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area contains a brief introduction to DVWA and a section titled "General Instructions".

Welcome :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla 11:14 AM

Welcome :: Damn V... 192.168.248.2/dvwa/index.php Search DVWA

Welcome to Damn Vulnerable Web Ap

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application the goal is to be an aid for security professionals to test their skills and tools in a le developers better understand the processes of securing web applications and to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every selecting any module and working up to reach the highest level they can before is not a fixed object to complete a module; however users should feel that they system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability w intentional. You are encouraged to try and discover as many issues as possible

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be increase the difficulty. This will demonstrate how adding another layer of securi actions. Note, there are also various public methods at bypassing these protec extension for more advanced users!

Έχουμε μια πλήρως λειτουργική ιστοσελίδα.



Ορισμός IP διεύθυνσης για το εσωτερικό δίκτυο.

Για να ρυθμίσουμε τις IP πηγαίνουμε στο path **/etc/networks** και με την εντολή **gedit Interfaces** και δηλώνουμε τις IP μας.

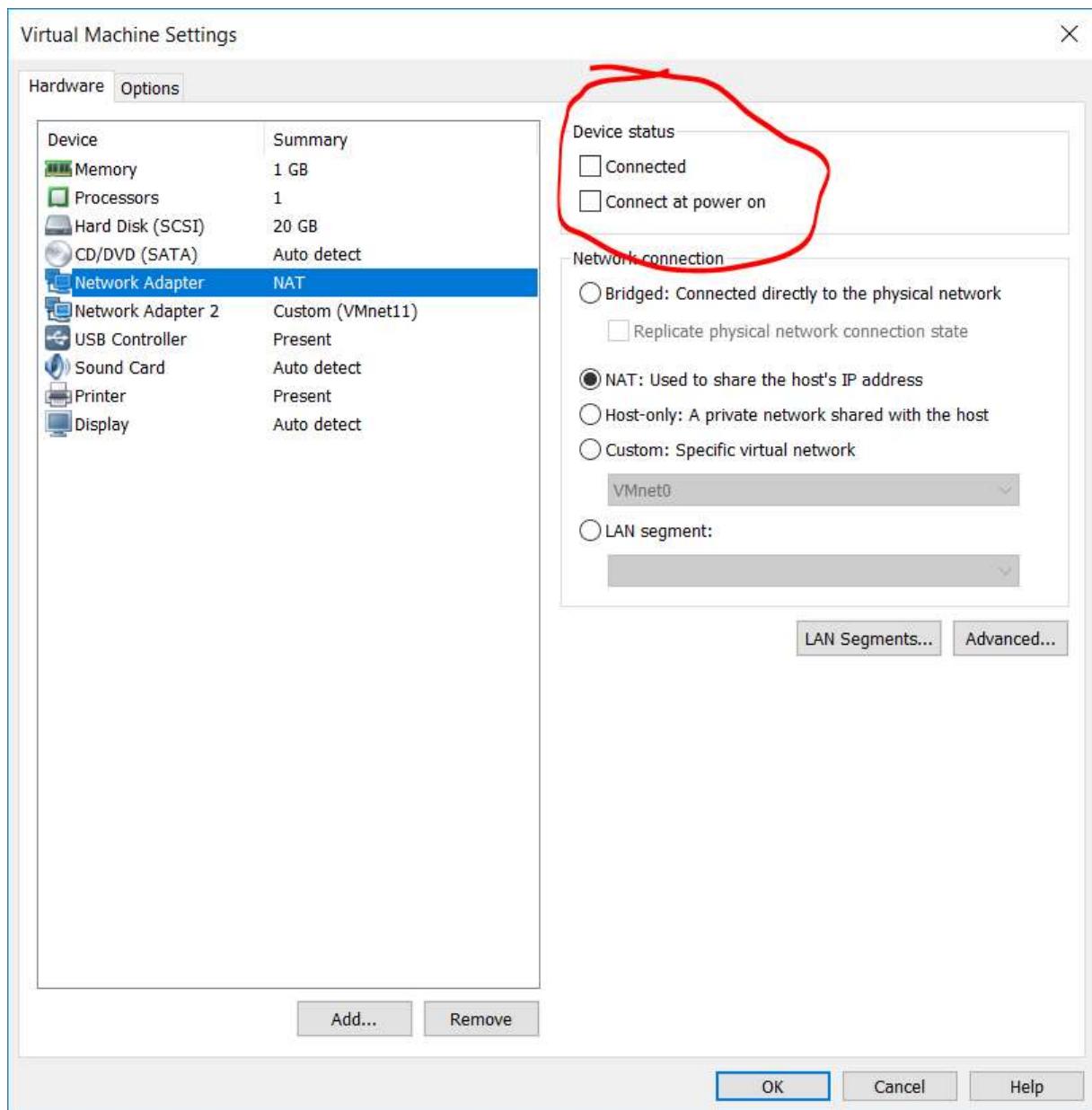
```
root@ubuntu:/etc/network
root@ubuntu:/etc/network# gedit interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens38
iface ens38 inet static
    address 192.168.248.2
    netmask 255.255.255.0
    gateway 192.168.248.1
```

Ορίζω τις IP διευθύνσεις 192.168.248.2 με default gateway 192.168.248.1 με σκοπό να έχει ως default gateway την 192.168.248.1 που έχει το router μας και από εκεί και μόνο να περνάει η σύνδεση.

Στην συνέχεια θα απενεργοποιήσω την NAT σύνδεση επιλέγω στις ρυθμίσεις του **Victim**.



Όπως μπορούμε να δούμε device status τα έχουμε απενεργοποιημένα.



Στην συνέχεια πληκτρολογούμε ifconfig για να δούμε τις διαθέσιμες ip.

```
victim@ubuntu: ~
victim@ubuntu:~$ ifconfig
ens33      Link encap:Ethernet HWaddr 00:0c:29:c0:f0:7a
           UP BROADCAST MULTICAST MTU:1500 Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B)   TX bytes:0 (0.0 B)

ens38      Link encap:Ethernet HWaddr 00:0c:29:c0:f0:84
           inet addr:192.168.248.2 Bcast:192.168.248.255 Mask:255.255.255.0
             inet6 addr: fe80::20c:29ff:fe00:84/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           RX packets:14 errors:0 dropped:0 overruns:0 frame:0
           TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1660 (1.6 KB)   TX bytes:9313 (9.3 KB)

lo         Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
             inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING MTU:65536 Metric:1
           RX packets:4420 errors:0 dropped:0 overruns:0 frame:0
           TX packets:4420 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1
           RX bytes:353086 (353.0 KB)   TX bytes:353086 (353.0 KB)

victim@ubuntu:~$
```

Επιβεβαιώνουμε ότι είναι ενεργοποιημένη μόνο η Ip 192.168.248.2



Εγκατάσταση ssh server και ενεργοποίηση πόρτας 22

```
[root@ubuntu ~]# apt-get install openssh-server
```

Ελέγχουμε αν η πόρτα έχει ανοίξει.

```
[root@ubuntu ~]# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2019-06-23 06:43:33 PDT; 3h 11min ago
    Process: 1399 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
    Process: 1387 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Process: 1146 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1157 (sshd)
      CGroup: /system.slice/ssh.service
              └─1157 /usr/sbin/sshd -D

Jun 23 06:43:34 ubuntu systemd[1]: Reloading OpenBSD Secure Shell server.
Jun 23 06:43:34 ubuntu sshd[1157]: Received SIGHUP; restarting.
Jun 23 06:43:34 ubuntu sshd[1157]: Server listening on 0.0.0.0 port 22.
Jun 23 06:43:34 ubuntu sshd[1157]: Server listening on :: port 22.
Jun 23 06:43:34 ubuntu systemd[1]: Reloaded OpenBSD Secure Shell server.
Jun 23 06:43:35 ubuntu systemd[1]: Reloading OpenBSD Secure Shell server.
Jun 23 06:43:35 ubuntu sshd[1157]: Received SIGHUP; restarting.
Jun 23 06:43:35 ubuntu sshd[1157]: Server listening on 0.0.0.0 port 22.
Jun 23 06:43:35 ubuntu sshd[1157]: Server listening on :: port 22.
Jun 23 06:43:35 ubuntu systemd[1]: Reloaded OpenBSD Secure Shell server.
lines 1-20/20 (END)...skipping...
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2019-06-23 06:43:33 PDT; 3h 11min ago
    Process: 1399 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
    Process: 1387 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Process: 1146 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1157 (sshd)
      CGroup: /system.slice/ssh.service
              └─1157 /usr/sbin/sshd -D

Jun 23 06:43:34 ubuntu systemd[1]: Reloading OpenBSD Secure Shell server.
Jun 23 06:43:34 ubuntu sshd[1157]: Received SIGHUP; restarting.
Jun 23 06:43:34 ubuntu sshd[1157]: Server listening on 0.0.0.0 port 22.
Jun 23 06:43:34 ubuntu sshd[1157]: Server listening on :: port 22.
Jun 23 06:43:34 ubuntu systemd[1]: Reloaded OpenBSD Secure Shell server.
Jun 23 06:43:35 ubuntu systemd[1]: Reloading OpenBSD Secure Shell server.
Jun 23 06:43:35 ubuntu sshd[1157]: Received SIGHUP; restarting.
Jun 23 06:43:35 ubuntu sshd[1157]: Server listening on 0.0.0.0 port 22.
Jun 23 06:43:35 ubuntu sshd[1157]: Server listening on :: port 22.
Jun 23 06:43:35 ubuntu systemd[1]: Reloaded OpenBSD Secure Shell server.
~
```



Δημιουργία Μηχανήματος Επιτιθέμενου Kali Linux

Με την ίδια διαδικασία θα ορίσουμε το δίκτυο μας να έχει το μηχάνημα την ip του εξωτερικού δικτύου ip 192.168.232.2 με default gateway 192.168.232.1.
Αντίστοιχα **etc/networks/ gedit interfaces**



```
root@kali:/etc/network
File Edit View Search Terminal Help
root@kali:/etc/network# gedit interfaces
```

Thu 14:26

interfaces
/etc/network

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth1
iface eth1 inet static
address 192.168.232.2
netmask 255.255.255.0
gateway 192.168.232.1
```

Ελέγχουμε την IP με την εντολή ifconfig

Thu 14:29

root@kali:/etc/network

```
File Edit View Search Terminal Help
root@kali:/etc/network# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.40.130 netmask 255.255.255.0 broadcast 192.168.40.255
inet6 fe80::20c:29ff:fe5:f00a prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:e5:f0:0a txqueuelen 1000 (Ethernet)
RX packets 1034 bytes 1294011 (1.2 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 39 bytes 3240 (3.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
device interrupt 19 base 0x2000

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.232.2 netmask 255.255.255.0 broadcast 192.168.232.255
inet6 fe80::20c:29ff:fe5:f014 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:e5:f0:14 txqueuelen 1000 (Ethernet)
RX packets 6 bytes 360 (360.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 682 bytes 40381 (39.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
device interrupt 16 base 0x2080

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 22 bytes 1194 (1.1 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 22 bytes 1194 (1.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

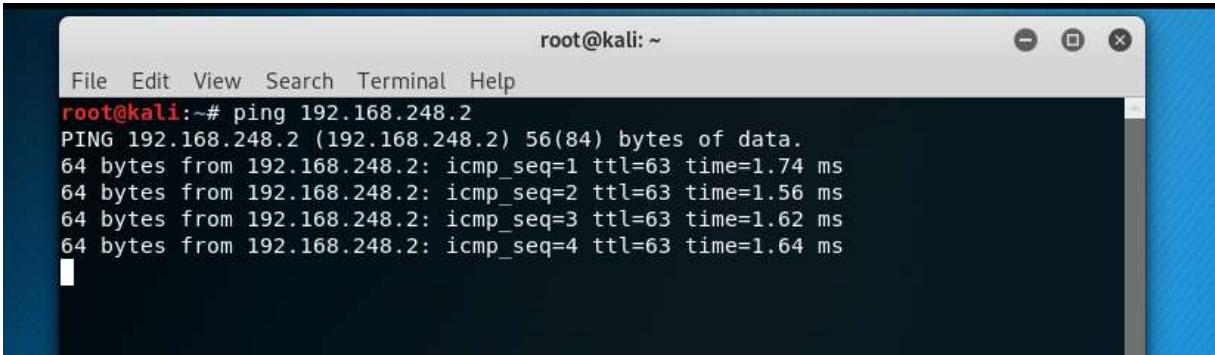
root@kali:/etc/network#
```



Όπως βλέπουμε έχουμε ορίσει την ip για το εξωτερικό δίκτυο 192.168.232.2 με default gateway.

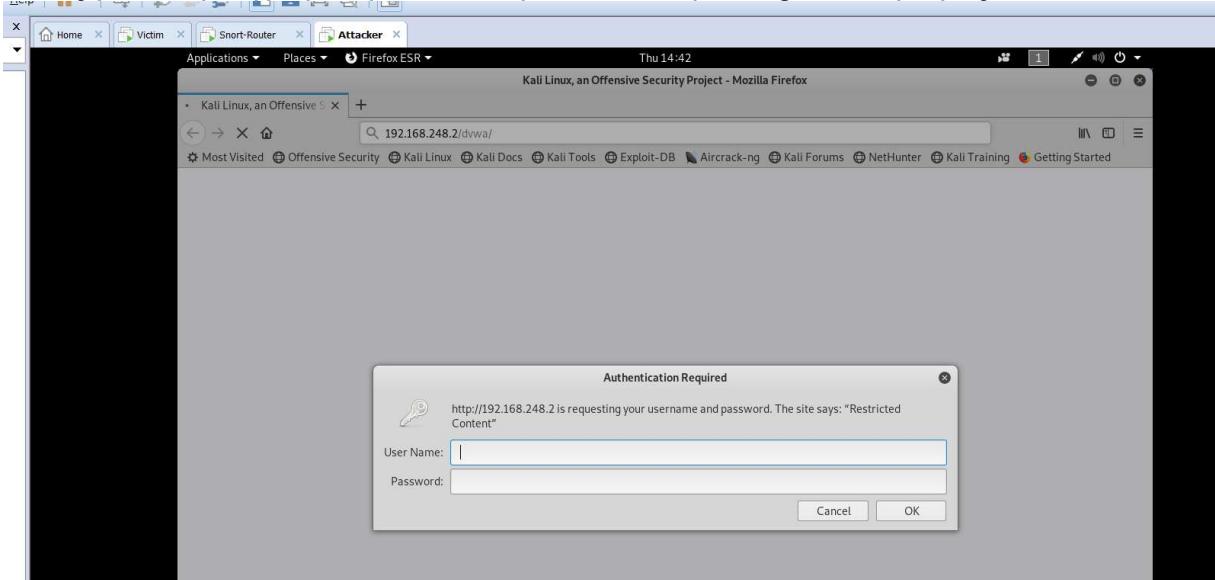
Επιπλέον καθώς εδώ θεωρούμαστε εξωτερικό δίκτυο θα κρατήσουμε την nat σύνδεση με ip 192.168.40.130 με σκοπό να συνδεόμαστε και στο internet.

Στην συνέχεια θα δοκιμάσουμε να κάνουμε ping στον Victim και να συνδεθούμε στη ιστοσελίδα.



```
root@kali:~# ping 192.168.248.2
PING 192.168.248.2 (192.168.248.2) 56(84) bytes of data.
64 bytes from 192.168.248.2: icmp_seq=1 ttl=63 time=1.74 ms
64 bytes from 192.168.248.2: icmp_seq=2 ttl=63 time=1.56 ms
64 bytes from 192.168.248.2: icmp_seq=3 ttl=63 time=1.62 ms
64 bytes from 192.168.248.2: icmp_seq=4 ttl=63 time=1.64 ms
```

Όπως βλέπουμε έχουμε την δυνατότητα να κάνουμε Ping στο θύμα μας.



Και μπορούμε όπως βλέπουμε να συνδεθούμε και στην ιστοσελίδα.

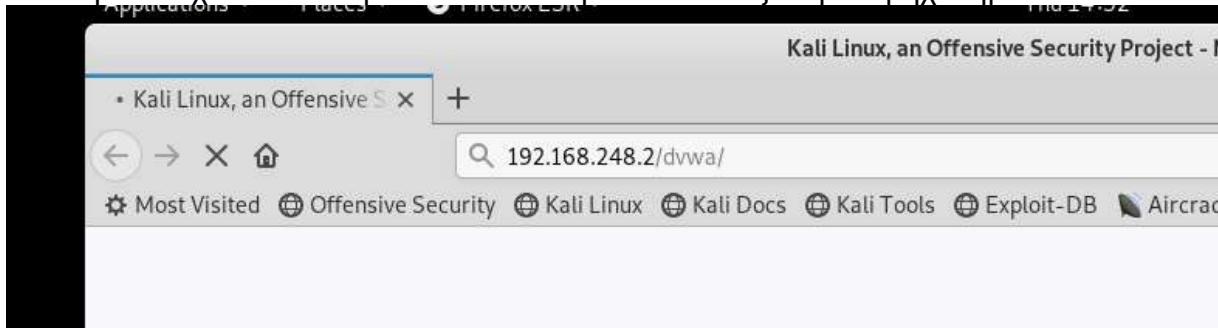
Σε περίπτωση που βάλουμε στον Victim ένα σχόλιο εκεί που έχουμε ορίσει το default gateway και δεν ορίζουμε ότι θέλουμε να περνάει από το router τότε δεν θα μπορούμε να κάνουμε Ping και να συνδεθούμε στην ιστοσελίδα.



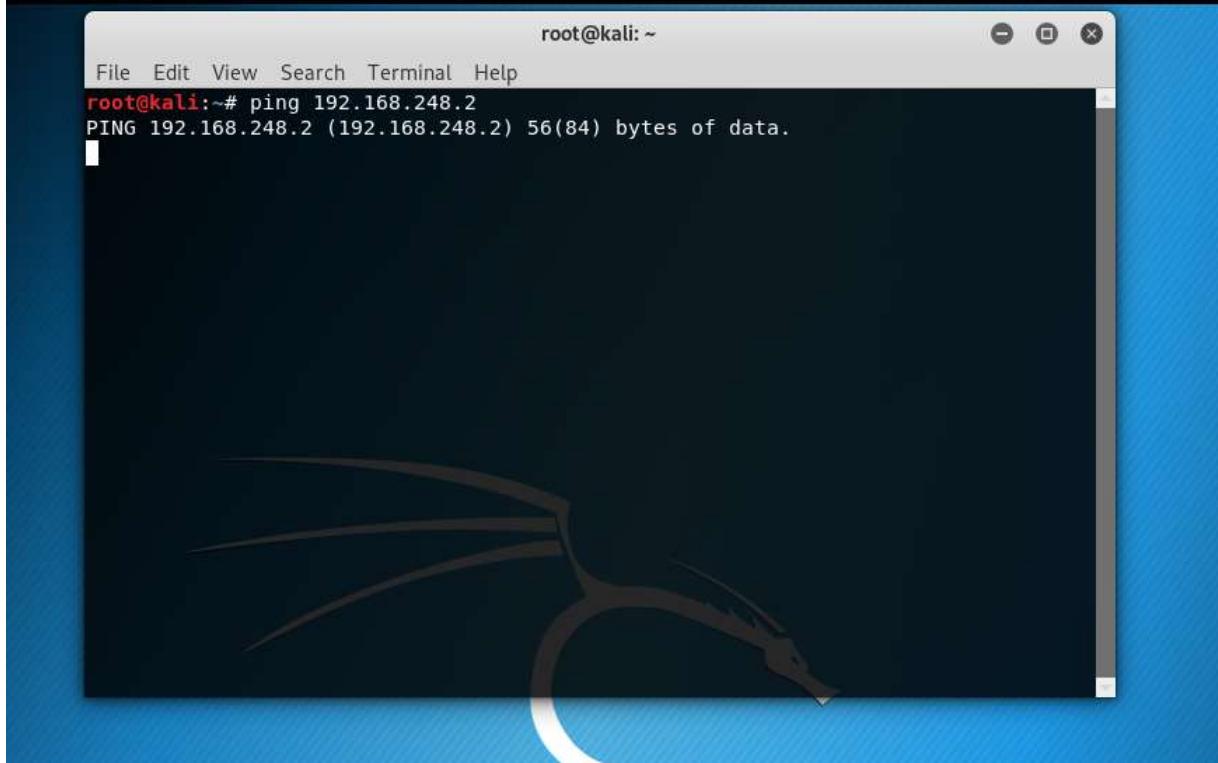
```
*interfaces (/etc/network) - gedit
Open ▾ Save
*interfaces
/etc/network
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens38
iface ens38 inet static
address 192.168.248.2
netmask 255.255.255.0
#gateway 192.168.248.1
```

Βάλλαμε το σχόλιο και πάμε να συνδεθούμε από τον εξωτερικό μηχάνημα.



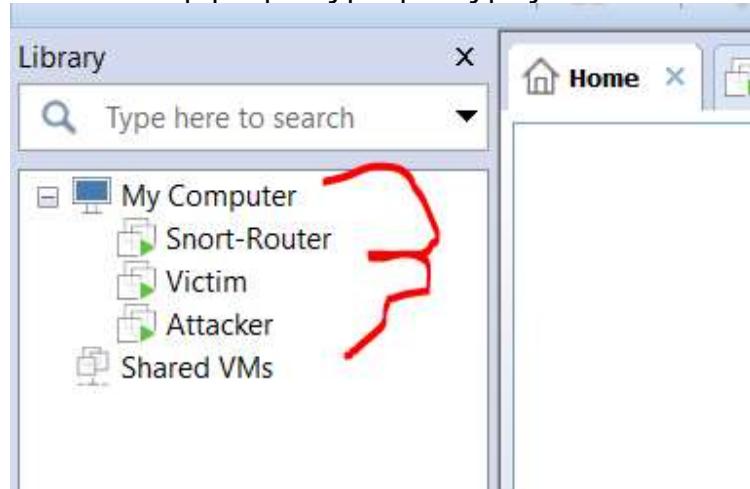
Όπως βλέπουμε δεν μπορεί να συνδεθεί αφού πλέον δεν επικοινωνεί με την default gateway.



Ενώ όπως βλέπουμε δεν μπορούμε να κάνουμε ping.



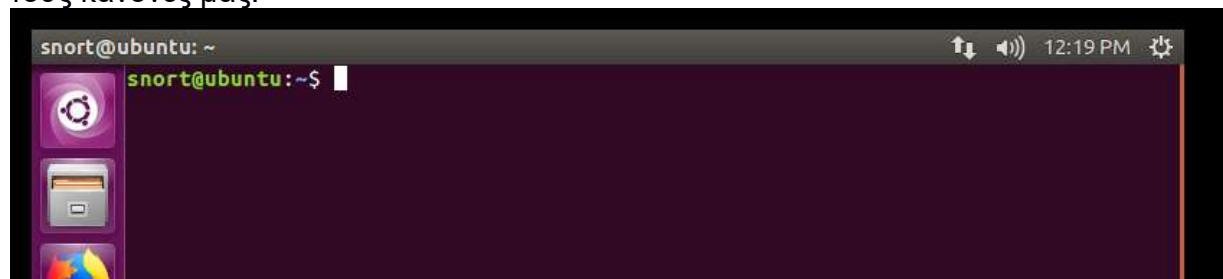
Ξανά επαναφέρουμε τις ρυθμίσεις μας.



Το δίκτυο μας έχει ολοκληρωθεί.

Ορισμός Κανόνων στον ανιχνευτή snort

Πηγαίνουμε στο μηχάνημα που έχουμε κάνει εγκατάσταση στο snort-router και γράφουμε τους κανόνες μας.



Σε πρώτο επίπεδο θα βάλουμε σε σχόλια όλους του κανόνες που έχει ορίσει από μόνου του το snort και στην συνέχεια θα γράψουμε τους δικούς μας κανόνες.





snort.conf (/etc/snort) - gedit

Open ▾ snort.conf /etc/snort Save

```
#include $RULE_PATH/exploit.rules
#include $RULE_PATH/file-executable.rules
#include $RULE_PATH/file-flash.rules
#include $RULE_PATH/file-identify.rules
#include $RULE_PATH/file-image.rules
#include $RULE_PATH/file-multimedia.rules
#include $RULE_PATH/file-office.rules
#include $RULE_PATH/file-other.rules
#include $RULE_PATH/file-pdf.rules
#include $RULE_PATH/finger.rules
#include $RULE_PATH/ftp.rules
#include $RULE_PATH/icmp-info.rules
#include $RULE_PATH/icmp.rules
#include $RULE_PATH/imap.rules
#include $RULE_PATH/indicator-compromise.rules
#include $RULE_PATH/indicator-obfuscation.rules
#include $RULE_PATH/indicator-shellcode.rules
#include $RULE_PATH/info.rules
#include $RULE_PATH/malware-backdoor.rules
#include $RULE_PATH/malware-cnc.rules
#include $RULE_PATH/malware-other.rules
#include $RULE_PATH/malware-tools.rules
#include $RULE_PATH/misc.rules
#include $RULE_PATH/multimedia.rules
#include $RULE_PATH/mysql.rules
#include $RULE_PATH/netbios.rules
#include $RULE_PATH/nntp.rules
#include $RULE_PATH/oracle.rules
#include $RULE_PATH/os-linux.rules
#include $RULE_PATH/os-other.rules
```

Plain Text ▾ Tab Width: 8 ▾ Ln 621, Col 2 ▾ INS

Βάζουμε σε όλους κανόνες σχόλιο το μόνο αρχείο που θα αφήσουμε είναι αυτό με τους local.rules το οποίο είναι από μόνο του κενό θα μπορούσαμε να ορίσουμε ένα νέο αρχείο με κατάληξη rules αλλά στην πραγματικότητα είναι η ίδια διαδικασία.

snort.conf (/etc/snort) - gedit

Open ▾ snort.conf /etc/snort Save

```
# If you install the official VRT Sourcefire configuration file and re-enable (remove the '#' character) the '# rules files that are available in your system (in the /etc/snort/rules directory)
# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:
```



```
root@ubuntu: /etc/snort/rules
root@ubuntu: /etc/snort/rules#
```

```
local.rules (/etc/snort/rules) - gedit
local.rules
Save
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

#for ftp buffer overflow
alert tcp $EXTERNAL_NET any ->$HOME_NET 21 (msg:"FTP NLST overflow attempt";content:"NLST";sid:1000007;)

#for brute force attack
alert tcp $EXTERNAL_NET any ->$HOME_NET 21 (msg:"FTP brute force failed login unicode attempt";content:"PASS";sid:1000008;)

alert tcp any any -> any 80 (msg:"Login attempt on webserver";content:"Authorization"; threshold:type limit, track_by_src,count 5,
seconds 10;sid:1000001;)

alert icmp any any -> any any (msg:"Someone is pinging";sid:1000002;)

alert tcp any any -> any 22 (msg:"SSH connection detected";sid:1000003;)

alert tcp any any -> any 80 (msg:"Possible sql injection";content:@"%27";sid:1000004;)

alert tcp any any -> any 80 (msg:"Possible sql injection";content:@"%22";sid:1000005;)

alert tcp any any -> any 80 (msg:"XSS Attack";content:"alert(1)";sid:1000006;)

alert tcp any any -> any 80 (msg:"Login attempt on webserver";content:"Authorization"; threshold:type limit, track_by_src,count 5,
seconds 10;sid:1000001;)
```

alert tcp any any -> any 80 (msg:"Login attempt on webserver";content:"Authorization"; threshold:type limit, track by_src,count 5, seconds 10;sid:1000001);

Ειδοποίηση για προσπάθειες σύνδεσης πάνω στην ιστοσελίδα βάζοντας όριο να εμφανιστεί η ειδοποίηση 5 φορές αν συμβεί μέσα σε λιγότερο από 10 δευτερόλεπτα.

```
alert icmp any any -> $HOME_NET any (msg:"Someone is pinging";threshold:type limit, track by_src,count 5, seconds 60;sid:1000002;)
```

alert tcp any any -> any 80 (msg:"Login attempt on webserver";content:"Authorization"; threshold:type limit, track by_src,count 5, seconds 10;sid:1000001);

Ειδοποίηση αν κάποιος πραγματοποιεί κάποιο ping στο μηχάνημα μας και εδώ σε αυτήν την περίπτωση οι ειδοποιήσεις λαμβάνονται ανά 5 μέσα σε 10 δευτερόλεπτα για το ίδιο alert.

```
alert tcp any any -> $HOME_NET 22 (msg:"SSH connection detected";threshold:type limit, track by_src,count 5, seconds 30;sid:1000003;)
```



```
alert tcp any any -> $HOME_NET 22 (msg:"SSH connection detected";threshold:type limit, track by_src,count 5, seconds 30;sid:1000003;)
```

Το SSH (Secure Shell) παρέχει στους χρήστες ένα ασφαλές, κρυπτογραφημένο μηχανισμό για σύνδεση, εκτέλεση εντολών και μεταφορά αρχείων. Το SSH χρησιμοποιεί συνήθως την θύρα 22 για την σύνδεση ανάμεσα στον υπολογιστή σας και έναν άλλο υπολογιστή ή συσκευή στο Internet ή στο τοπικό σας δίκτυο.

Ειδοποίηση για SSH συνδέσεις στην πόρτα 22.

```
alert tcp any any -> $HOME_NET 22 (msg:"Nmap xmas scan";sid:1000009;)
```

```
alert tcp any any -> $HOME_NET 22 (msg:"Nmap xmas scan";sid:1000009;)
```

Ειδοποιήσεις αν γίνεται έλεγχος θυρών.

```
alert tcp any any -> $HOME_NET 80 (msg:"Possible sql injection for single quotes ";content:"%27";sid:1000004;)
```

```
alert tcp any any -> $HOME_NET 80 (msg:"Possible sql injection for single quotes ";content:"%27";sid:1000004;)
```

Ειδοποιήσεις σε sql injection από url με sigle quotes attacks

```
alert tcp any any -> $HOME_NET 80 (msg:"Possible sql injection for double quotes";content:"%22";sid:1000005;)
```

```
alert tcp any any -> $HOME_NET 80 (msg:"Possible sql injection for double quotes";content:"%22";sid:1000005;)
```

Ειδοποιήσεις σε sql injection από url με double quotes attacks

```
alert tcp any any -> $HOME_NET 80 (msg:"Cross site scripting attack";pcre:"/((\%3C)|<)((\%2F)|\\/*[a-zA-Z0-9\%]+((\%3E)|>)/i";sid:1000006;)
```

```
alert tcp any any -> $HOME_NET 80 (msg:"Cross site scripting attack";pcre:"/((\%3C)|<)((\%2F)|\\/*[a-zA-Z0-9\%]+((\%3E)|>)/i";sid:1000006;)
```

Cross site scripting attack χρήση του url με την βοήθεια παραμέτρων για να περάσουν scripts

Ειδοποιήσεις πάνω σε cross site scripting attacks

```
alert tcp any any -> $HOME_NET 3306 (msg:"MYSQL root login attempt"; flow:to_server,established; sid:1000010;)
```



```
alert tcp any any -> $HOME_NET 3306 (msg:"MYSQL root login attempt";  
flow:to_server,established; sid:1000010;)
```

Ειδοποιήσεις αν γίνει προσπάθεια κάποιος να συνδεθεί κάποιος σαν διαχειριστής στην βάση.

Giana diabasome ta log files

```
** (gedit:4112): WARNING **: Set document metadata failed: Setting attr  
root@ubuntu:/var/log/snort# tcpdump -r snort.log.1560671185  
reading from file snort.log.1560671185, link-type EN10MB (Ethernet)  
00:46:30.760614 IP 192.168.248.2 > 192.168.222.2: TCMP 192.168.248.2 udo
```



2. Έλεγχος καταγραφής επιθέσεων. Από έναν άλλο κόμβο που θα έχει το ρόλο του επιτιθέμενου να δοκιμάσετε να προσομοιώσετε κάποιες επιθέσεις χρησιμοποιώντας κατάλληλα εργαλεία. Παράδειγμα επίθεσης μπορεί να περιλαμβάνουν χαρτογράφηση του δικτύου, επιθέσεις άρνησης υπηρεσίας (Denial of Service attacks) κτλ. Να αναλύσετε τα αρχεία καταγραφής του εργαλείου (2 μονάδες)

Αρχικά θα πάμε στο snort – router και θα ενεργοποιήσουμε το snort με την παρακάτω εντολή.

```
snort@ubuntu:~  
[root@snort ~]# snort -q -A console -i ens38 -c /etc/snort/snort.conf
```

snort -q -A console -i ens38 -c /etc/snort.conf



Nmap brute force attack

Χρησιμοποιήστε την ακόλουθη εντολή Nmap για να πραγματοποιήσετε τον έλεγχο κωδικού πρόσβασης με **brute force attack** για έναν βασικό έλεγχο ταυτότητας που προστατεύεται από το πρωτόκολλο HTTP

```
root@kali:~# nmap -script http-brute -p 80 192.168.248.2
Starting Nmap 7.70 ( https://nmap.org/ ) at 2019-06-23 10:28 EDT
Nmap scan report for 192.168.248.2
Host is up (0.00099s latency).
```

```
root@kali:~# nmap -script http-brute -p 80 192.168.248.2
Starting Nmap 7.70 ( https://nmap.org/ ) at 2019-06-23 10:28 EDT
Nmap scan report for 192.168.248.2
Host is up (0.00099s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-brute:
|_ Accounts:
|   web:web - Valid credentials
|_ Statistics: Performed 45010 guesses in 41 seconds, average tps: 1088.4

Nmap done: 1 IP address (1 host up) scanned in 42.41 seconds
root@kali:~#
```

Βλέπουμε ότι το όνομα χρήστη και ο κωδικός είναι σχετικά απλός **username:web** και **password :web**

Στην συνέχεια πηγαίνουμε και ελέγχουμε την κονσόλα στο snort.



root@ubuntu:/# snort -q -A console -i ens38 -c /etc/snort/snort.conf

```

06/23-07:28:19.924516 [**] [1:1000002:0] Someone is pinging [*] [Priority: 0] [ICMP] 192.168.232.2 -> 192.168.248.2
06/23-07:28:19.924603 [**] [1:1000002:0] Someone is pingng [*] [Priority: 0] [ICMP] 192.168.232.2 -> 192.168.248.2
06/23-07:28:19.924886 [**] [1:1000002:0] Someone is pinging [*] [Priority: 0] [ICMP] 192.168.248.2 -> 192.168.232.2
06/23-07:28:19.924971 [**] [1:1000002:0] Someone is pinging [*] [Priority: 0] [ICMP] 192.168.248.2 -> 192.168.232.2
06/23-07:28:20.170470 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52196 -> 192.168.248.2:80
06/23-07:28:20.171079 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52198 -> 192.168.248.2:80
06/23-07:28:20.171293 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52208 -> 192.168.248.2:80
06/23-07:28:20.171690 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52208 -> 192.168.248.2:80
06/23-07:28:20.172231 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52204 -> 192.168.248.2:80
06/23-07:28:30.003366 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:43978 -> 192.168.248.2:80
06/23-07:28:30.003416 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:43982 -> 192.168.248.2:80
06/23-07:28:30.003445 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:43982 -> 192.168.248.2:80
06/23-07:28:30.003449 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:43984 -> 192.168.248.2:80
06/23-07:28:40.003861 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:39318 -> 192.168.248.2:80
06/23-07:28:40.003908 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:39320 -> 192.168.248.2:80
06/23-07:28:40.003912 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:39322 -> 192.168.248.2:80
06/23-07:28:40.003941 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:39324 -> 192.168.248.2:80
06/23-07:28:40.004000 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:39326 -> 192.168.248.2:80
06/23-07:28:50.005105 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:33046 -> 192.168.248.2:80
06/23-07:28:50.005135 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:33050 -> 192.168.248.2:80
06/23-07:28:50.005138 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:33052 -> 192.168.248.2:80
06/23-07:28:50.006904 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:33056 -> 192.168.248.2:80
06/23-07:29:00.002747 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54530 -> 192.168.248.2:80
06/23-07:29:00.003369 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54532 -> 192.168.248.2:80
06/23-07:29:00.003432 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54534 -> 192.168.248.2:80
06/23-07:29:00.003511 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54536 -> 192.168.248.2:80
06/23-07:29:00.003609 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54538 -> 192.168.248.2:80

```

Όπως μπορούμε να παρατηρήσουμε μας έχει έρθει ειδοποίηση στην αρχή ότι κάποιος έχει κάνει ping με ip 192.168.232.2 η ip του attacker πάνω στην ip του Victim με ip 192.168.248.2

```

06/23-07:28:19.924886 [**] [1:1000002:0] Someone is pingting [*] [Priority: 0] [ICMP] 192.168.248.2 -> 192.168.232.2
06/23-07:28:19.924971 [**] [1:1000002:0] Someone is pingng [*] [Priority: 0] [ICMP] 192.168.248.2 -> 192.168.232.2
06/23-07:28:20.170470 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:52196 -> 192.168.

```

Στην συνέχεια βλέπουμε ότι η ίδια ip για πολλές φορές έχει προσπαθήσει να είσει στον web server και καθώς το έχουμε ορίσει εμείς από τους κανόνες μας αφού η επίθεση είχε διάρκεια 43 δευτερόλεπτα και έχουμε 20 και ειδοποιήσεις πράγμα που είναι φυσιολογικό καθώς εμφανίζονται στην κονσόλα 5 ειδοποιήσεις ανά δέκα δευτερόλεπτα.

```

06/23-07:29:00.003369 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54532 -> 192.168.248.2:80
06/23-07:29:00.003432 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54534 -> 192.168.248.2:80
06/23-07:29:00.003511 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54536 -> 192.168.248.2:80
06/23-07:29:00.003609 [**] [1:1000001:0] Login attempt on webserver [*] [Priority: 0] [TCP] 192.168.232.2:54538 -> 192.168.248.2:80

```

```

root@ubuntu: /var/log/snort
root@ubuntu:/var/log/snort# tcpdump -r snort.log.1561300093

```



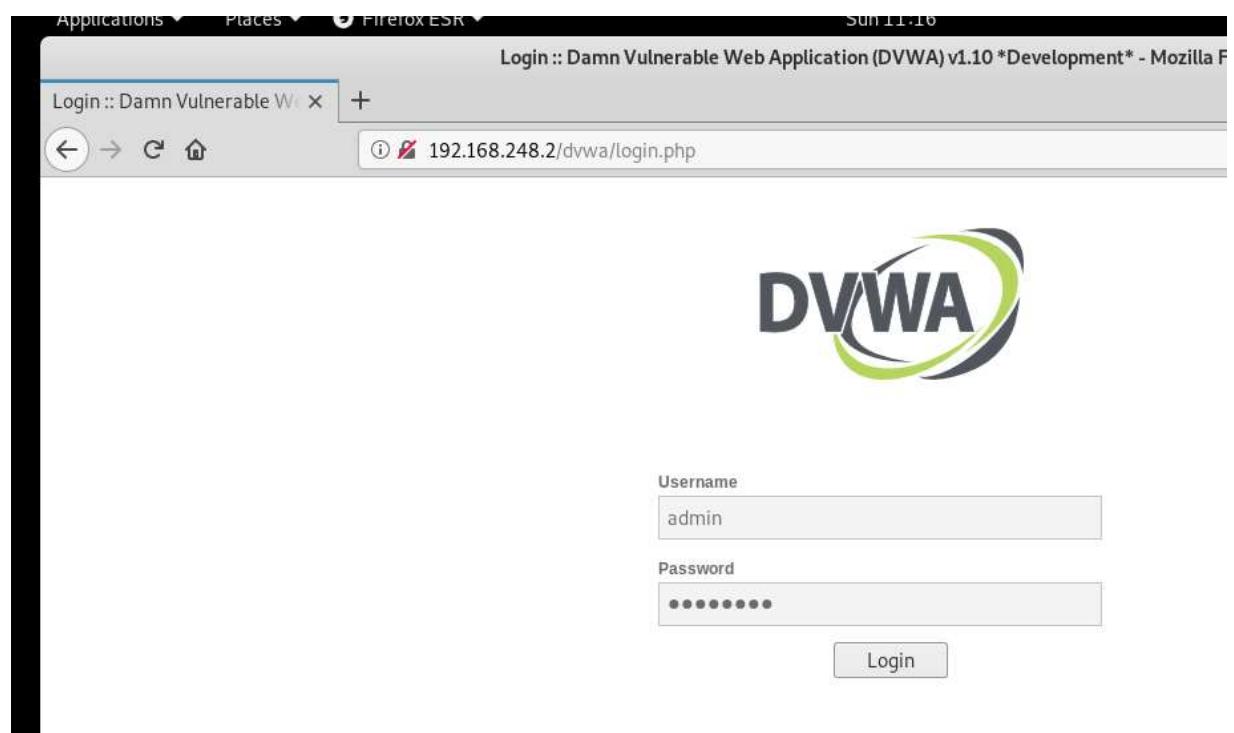
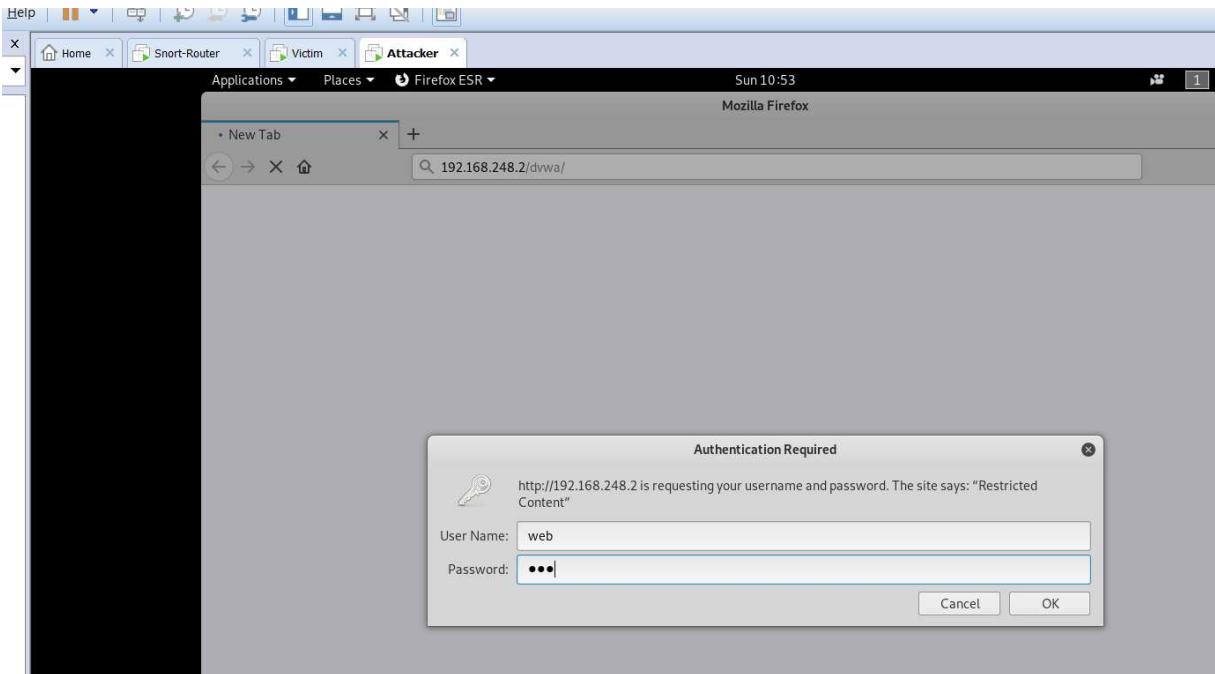
```
root@ubuntu:/var/log/snort# tcpdump -r snort.log.1561306169
reading from file snort.log.1561306169, link-type EN10MB (Ethernet)
09:09:35.260619 IP 192.168.232.2 > 192.168.248.2: ICMP echo request, id 895, seq 0, length 8
09:09:35.260727 IP 192.168.232.2 > 192.168.248.2: ICMP time stamp query id 49199 seq 0, length 20
09:09:35.260897 IP 192.168.248.2 > 192.168.232.2: ICMP echo reply, id 895, seq 0, length 8
09:09:35.260979 IP 192.168.248.2 > 192.168.232.2: ICMP time stamp reply id 49199 seq 0: org 00:00:00.000, recv 16:09:37.006, xmit 16:09:37.006, length 20
09:09:35.449537 IP 192.168.232.2.57994 > 192.168.248.2:http: Flags [P.], seq 4201182765:4201182951, ack 719411421, win 457, options [nop,nop,TS val 349505115 ecr 2118322], length 186: HTTP: GET / HTTP/1.1
09:09:35.449706 IP 192.168.232.2.57996 > 192.168.248.2:http: Flags [P.], seq 2931468811:2931469001, ack 1928639425, win 457, options [nop,nop,TS val 349505115 ecr 2118323], length 190: HTTP: GET / HTTP/1.1
09:09:35.449934 IP 192.168.232.2.57998 > 192.168.248.2:http: Flags [P.], seq 1974646917:1974647127, ack 1778680707, win 457, options [nop,nop,TS val 349505115 ecr 2118323], length 210: HTTP: GET / HTTP/1.1
09:09:35.450303 IP 192.168.232.2.58000 > 192.168.248.2:http: Flags [P.], seq 452681500:452681698, ack 2692661202, win 457, options [nop,nop,TS val 349505116 ecr 2118323], length 198: HTTP: GET / HTTP/1.1
09:09:35.450426 IP 192.168.232.2.58002 > 192.168.248.2:http: Flags [P.], seq 4120807882:4120808080, ack 282656802, win 457, options [nop,nop,TS val 349505116 ecr 2118323], length 198: HTTP: GET / HTTP/1.1
09:09:45.001325 IP 192.168.232.2.46706 > 192.168.248.2:http: Flags [P.], seq 1174019989:1174020183, ack 3414697727, win 457, options [nop,nop,TS val 349514665 ecr 2120708], length 194: HTTP: GET / HTTP/1.1
09:09:45.001483 IP 192.168.232.2.46708 > 192.168.248.2:http: Flags [P.], seq 82143284:82143490, ack 2994498777, win 457, options [nop,nop,TS val 349514666 ecr 2120709], length 206: HTTP: GET / HTTP/1.1
09:09:45.001496 IP 192.168.232.2.46710 > 192.168.248.2:http: Flags [P.], seq 3210486575:3210486773, ack 1091475594, win 457, options [nop,nop,TS val 349514666 ecr 2120709], length 198: HTTP: GET / HTTP/1.1
09:09:45.001574 IP 192.168.232.2.46712 > 192.168.248.2:http: Flags [P.], seq 2872904744:2872904942, ack 1221119099, win 457, options [nop,nop,TS val 349514666 ecr 2120709], length 198: HTTP: GET / HTTP/1.1
09:09:45.001582 IP 192.168.232.2.46714 > 192.168.248.2:http: Flags [P.], seq 1646081558:1646081756, ack 2673562883, win 457, options [nop,nop,TS val 349514666 ecr 2120710], length 198: HTTP: GET / HTTP/1.1
09:09:55.000324 IP 192.168.232.2.33270 > 192.168.248.2:http: Flags [P.], seq 1438502911:1438503101, ack 107431627, win 457, options [nop,nop,TS val 349524665 ecr 2123210], length 190: HTTP: GET / HTTP/1.1
09:09:55.000372 IP 192.168.232.2.33272 > 192.168.248.2:http: Flags [P.], seq 1757689730:1757689932, ack 2587530602, win 457, options [nop,nop,TS val 349524665 ecr 2123210], length 202: HTTP: GET / HTTP/1.1
09:09:55.000376 IP 192.168.232.2.33274 > 192.168.248.2:http: Flags [P.], seq 3065446087:3065446281, ack 3682728315, win 457, options [nop,nop,TS val 349524665 ecr 2123211], length 194: HTTP: GET / HTTP/1.1
09:09:55.000422 IP 192.168.232.2.33276 > 192.168.248.2:http: Flags [P.], seq 3212365845:3212366039, ack 3243604972, win 457, options [nop,nop,TS val 349524665 ecr 2123211], length 194: HTTP: GET / HTTP/1.1
09:09:55.000428 IP 192.168.232.2.33278 > 192.168.248.2:http: Flags [P.], seq 1042623455:1042623649, ack 2655196012, win 457, options [nop,nop,TS val 349524665 ecr 2123211], length 194: HTTP: GET / HTTP/1.1
09:10:05.002154 IP 192.168.232.2.49456 > 192.168.248.2:http: Flags [P.], seq 1437968282:1437968476, ack 3393665689, win 457, options [nop,nop,TS val 349534666 ecr 2125711], length 194: HTTP: GET / HTTP/1.1
09:10:05.027732 IP 192.168.232.2.49458 > 192.168.248.2:http: Flags [P.], seq 3566814261:3566814451, ack 1463645644, win 457, options [nop,nop,TS val 349534666 ecr 2125711], length 194: HTTP: GET / HTTP/1.1
```

Ανοίγουμε το αρχείο από τα logs.



Sql Injection

Στην συνέχεια θα συνδεθούμε στο αφού γνωρίζουμε τον κωδικό πρόσβασης του http server μας και θα πραγματοποιήσουμε από τον dvwa damn vulnerable web application sql injection επιθέσεις.





Username:admin
Password:password

The screenshot shows a Mozilla Firefox browser window with the title "Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox". The address bar shows the URL "192.168.248.2/dvwa/vulnerabilities/sqli/". The main content area displays the DVWA logo and the heading "Vulnerability: SQL Injection". On the left, there is a sidebar menu with various vulnerability categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (selected), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, and PHP Info. Below the menu, a "More Information" section lists several links related to SQL injection. A form at the bottom allows users to enter a "User ID" and click "Submit".

Επιλέγουμε sql injection και κάνουμε την επίθεση.

The screenshot shows the DVWA SQL Injection page again. The "User ID" field contains the value "1' or 1=1". The "Submit" button is visible next to the input field. This represents a common SQL injection exploit where a single quote character is followed by "or 1=1" to bypass the database's security checks.

Πληκτρολογούμε 1' or 1=1



Vulnerability: SQL Injection

User ID:

ID: 'or 1='1
First name: admin
Surname: admin

ID: 'or 1='1
First name: Gordon
Surname: Brown

ID: 'or 1='1
First name: Hack
Surname: Me

ID: 'or 1='1
First name: Pablo
Surname: Picasso

ID: 'or 1='1
First name: Bob
Surname: Smith

Όπως βλέπουμε μας επιστρέφει εγγραφές από την βάση δεδομένων

```
06/23-08:45:55.040985 [**] [1:1000004:0] Possible sql injection for single quotes [**] [Priority: 0] {TCP} 192.168.232.2:52474 -> 192.168.248.2:80
```

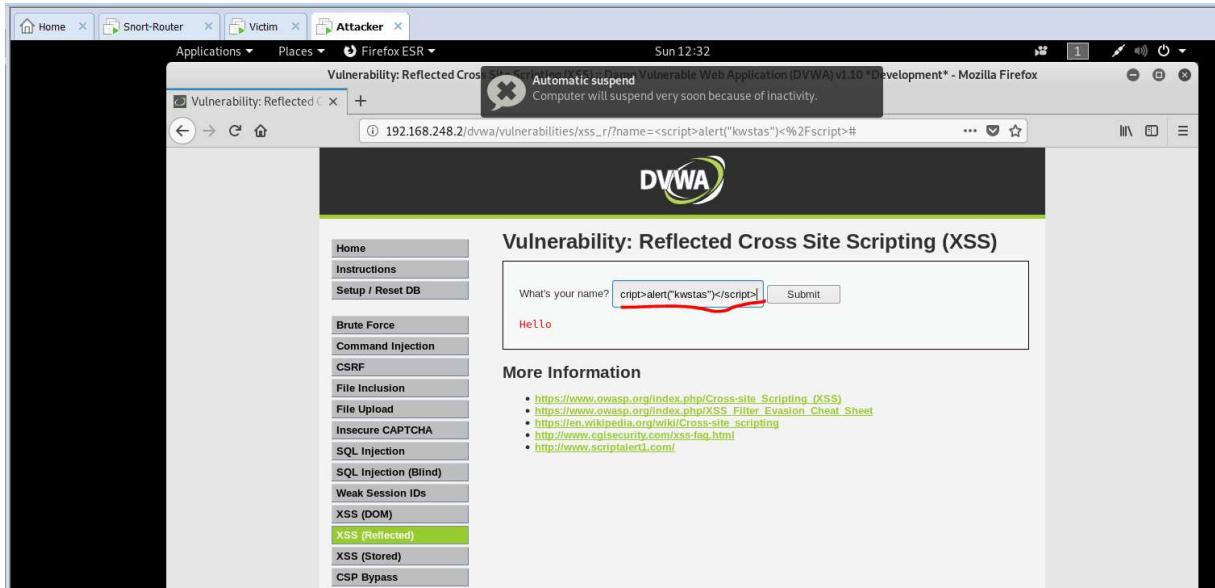
Πάμε στην κονσόλα να δούμε της ειδοποιήσεις από το snort μας εμφανίζεται ότι έχει γίνει sql injection με μονό αυτάκι στο url.

```
root@ubuntu:/var/log/snort# tcpdump -r snort.log.1561305704
reading from file snort.log.1561305704, link-type EN10MB (Ethernet)
09:02:00.764113 IP 192.168.232.2.52532 -> 192.168.248.2.http: Flags [P.], seq 3477171872:3477172400, ack 3598219829, win 457, options [nop,nop,TS val 349050426 ecr 2004651], length 528: HTTP: GET /dvwa/vulnerabilities/sql_injection/ HTTP/1.1
09:02:00.764113 IP 192.168.232.2.52532 -> 192.168.248.2.http: Flags [P.], seq 0:528, ack 1, win 457, options [nop,nop,TS val 349050426 ecr 2004651], length 528: HTTP: GET /dvwa/vulnerabilities/sql_injection/ HTTP/1.1
09:02:01.636632 IP 192.168.232.2.52532 -> 192.168.248.2.http: Flags [P.], seq 528:1024, ack 1776, win 547, options [nop,nop,TS val 349051299 ecr 2004674], length 496: HTTP: GET /dvwa/vulnerabilities/sql_injection/ HTTP/1.1
09:02:06.230803 IP 192.168.232.2.52532 -> 192.168.248.2.http: Flags [P.], seq 1024:1546, ack 3548, win 638, options [nop,nop,TS val 349055893 ecr 2004873], length 522: HTTP: GET /dvwa/vulnerabilities/sql_injection/?id=%27or+1%3D%271&Submit=Submit HTTP/1.1
09:02:06.230803 IP 192.168.232.2.52532 -> 192.168.248.2.http: Flags [P.], seq 1024:1546, ack 3548, win 638, options [nop,nop,TS val 349055893 ecr 2004873], length 522: HTTP: GET /dvwa/vulnerabilities/sql_injection/?id=%27or+1%3D%271&Submit=Submit HTTP/1.1
root@ubuntu:/var/log/snort#
```

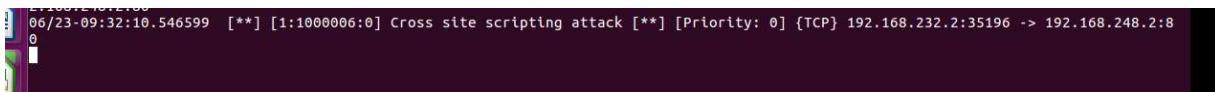
Όπως μπορούμε να δούμε στα logs βλέπουμε την sqli εντολή και περιέχει το single quote %27.



Cross site Scripting Attack



Πηγαίνουμε στην κονσόλα του snort και όπως μπορούμε να δούμε η επίθεση έχει πραγματοποιηθεί.



Ανοίγουμε τα log files

```
root@ubuntu:/var/log/snort# tcpdump -r snort.log.1561307509
reading from file snort.log.1561307509, link-type EN10MB (Ethernet)
09:32:05.019121 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 2712496402:2712496950, ack 2547467740, win 457, options [nop,nop,TS val 350854679 ecr 2455714], length 548: HTTP: GET /dvwa/vulnerabilities/xss_s/ HTTP/1.1
09:32:05.019121 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 0:548, ack 1, win 457, options [nop,nop,TS val 350854679 ecr 2455714], length 548: HTTP: GET /dvwa/vulnerabilities/xss_s/ HTTP/1.1
09:32:05.019121 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 0:548, ack 1, win 457, options [nop,nop,TS val 350854679 ecr 2455714], length 548: HTTP: GET /dvwa/vulnerabilities/xss_s/ HTTP/1.1
09:32:07.836067 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 1040:1588, ack 2046, win 547, options [nop,nop,TS val 350860207 ecr 2456423], length 548: HTTP: GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22kwstas%22%29%3C%2Fscript%3E HTTP/1.1
09:32:10.546599 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 1040:1588, ack 3799, win 638, options [nop,nop,TS val 350860207 ecr 2456423], length 548: HTTP: GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22kwstas%22%29%3C%2Fscript%3E HTTP/1.1
09:32:10.546599 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 1040:1588, ack 3799, win 638, options [nop,nop,TS val 350860207 ecr 2456423], length 548: HTTP: GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22kwstas%22%29%3C%2Fscript%3E HTTP/1.1
root@ubuntu:/var/log/snort#
```

Όπως βλέπουμε έχει γίνει επίθεση cross site scripting

09:32:10.546599 IP 192.168.232.2.35196 > 192.168.248.2.http: Flags [P.], seq 1040:1588, ack 3799, win 638, options [nop,nop,TS val 350860207 ecr 2456423], length 548: HTTP: GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22kwstas%22%29%3C%2Fscript%3E HTTP/1.1



Με την εντολή nmap -sX -p22 192.168.205.5 κάνουμε xmas scan.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# nmap -sX -p 22 192.168.248.2
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 13:12 EDT
Nmap scan report for 192.168.248.2
Host is up (0.0012s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
root@kali:~#
```

Router-Snort Console

```
root@ubuntu:/var/log/snort
root@ubuntu:/var/log/snort# snort -q -A console -i ens38 -c /etc/snort/snort.conf
06/23-10:13:37.455867 [**] [1:1000002:0] Someone is pinging [**] [Priority: 0] {ICMP} 192.168.232.2 -> 192.168.248.2
06/23-10:13:37.456007 [**] [1:1000002:0] Someone is pinging [**] [Priority: 0] {ICMP} 192.168.232.2 -> 192.168.248.2
06/23-10:13:37.456382 [**] [1:1000002:0] Someone is pinging [**] [Priority: 0] {ICMP} 192.168.248.2 -> 192.168.232.2
06/23-10:13:37.456467 [**] [1:1000002:0] Someone is pinging [**] [Priority: 0] {ICMP} 192.168.248.2 -> 192.168.232.2
06/23-10:13:37.545149 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:55658 -> 192.168.248.2:22
06/23-10:13:37.545149 [**] [1:1000009:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:55658 -> 192.168.248.2:22
06/23-10:13:37.645616 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:55659 -> 192.168.248.2:22
06/23-10:13:37.645616 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:55659 -> 192.168.248.2:22
```

Ειδοποιήσεις για nmap scan και ssh connections

Στην συνέχεια με το εργαλείο hydra θα κάνουμε brute force attack για να συνδεθούμε απομακρυσμένα από την πόρτα 22 με ssh

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hydra -l root -p root 192.168.248.2 -t 4 ssh
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-06-23 13:20:
53
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:0), ~1 try pe
r task
[DATA] attacking ssh://192.168.248.2:22/
1 of 1 target completed, 0 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-06-23 13:20:
56
root@kali:~#
```



```

06/23-10:13:37.545149 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:55658 -> 192.168.248.2:22
06/23-10:13:37.545149 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:55658 -> 192.168.248.2:22
06/23-10:13:37.645616 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:55659 -> 192.168.248.2:22
06/23-10:10:20:53.872916 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:55659 -> 192.168.248.2:22
06/23-10:20:53.872916 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.875795 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.875795 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.875795 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.875795 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.875914 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.949208 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.949208 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.950691 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.950691 [**] [1:1000003:0] SSH connection detected [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22
06/23-10:20:53.952495 [**] [1:1000009:0] Nmap xmas scan [**] [Priority: 0] {TCP} 192.168.232.2:52752 -> 192.168.248.2:22

```

Avoíγουμε τα logs

```

(gedit:3268): IBUS-WARNING **: The owner of /home/snort/.config/ibus/bus is not root!
root@ubuntu:[/var/log/snort# tcpdump -r snort.log.1561310013
reading from file snort.log.1561310013, link-type EN10MB (Ethernet)
10:13:37.455867 IP 192.168.232.2 > 192.168.248.2: ICMP echo request, id 42087, seq 0, length 8
10:13:37.456007 IP 192.168.232.2 > 192.168.248.2: ICMP time stamp query id 60026 seq 0, length 20
10:13:37.456382 IP 192.168.248.2 > 192.168.232.2: ICMP echo reply, id 42087, seq 0, length 8
10:13:37.456467 IP 192.168.248.2 > 192.168.232.2: ICMP time stamp reply id 60026 seq 0: org 00:00:00.0000, recv 17:13:39.190, xmit 17:13:39.190, length 20
10:13:37.545149 IP 192.168.232.2:55658 > 192.168.248.2:ssh: Flags [FPU], seq 3182025628, win 1024, urg 0, length 0
10:13:37.545149 IP 192.168.232.2:55658 > 192.168.248.2:ssh: Flags [FPU], seq 3182025628, win 1024, urg 0, length 0
10:13:37.645616 IP 192.168.232.2:55659 > 192.168.248.2:ssh: Flags [FPU], seq 3181960093, win 1024, urg 0, length 0
10:20:53.872916 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [S], seq 476038263, win 29200, options [mss 1460,sackOK,TS val 353783525 ecr 0,nop,wscale 6], length 0
10:20:53.872916 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [S], seq 476038263, win 29200, options [mss 1460,sackOK,TS val 353783525 ecr 0,nop,wscale 6], length 0
10:20:53.875795 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 3468240931, win 457, options [nop,nop,TS val 353783528 ecr 3187927], length 0
10:20:53.875795 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 1, win 457, options [nop,nop,TS val 353783528 ecr 3187927], length 0
10:20:53.875914 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 0:22, ack 1, win 457, options [nop,nop,TS val 353783528 ecr 3187927], length 22
10:20:53.875914 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 0:22, ack 1, win 457, options [nop,nop,TS val 353783528 ecr 3187927], length 22
10:20:53.949208 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 42, win 457, options [nop,nop,TS val 353783601 ecr 3187945], length 0
10:20:53.949208 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 42, win 457, options [nop,nop,TS val 353783601 ecr 3187945], length 0
10:20:53.950691 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 1018, win 487, options [nop,nop,TS val 353783603 ecr 3187946], length 0
10:20:53.950691 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 1018, win 487, options [nop,nop,TS val 353783603 ecr 3187946], length 0
10:20:53.952495 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 22:670, ack 1018, win 487, options [nop,nop,TS val 353783604 ecr 3187946], length 48
10:20:53.953410 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 670:718, ack 1018, win 487, options [nop,nop,TS val 353783605 ecr 3187946], length 48
10:20:53.9960391 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 718:734, ack 1394, win 518, options [nop,nop,TS val 353783606 ecr 3187948], length 16
10:20:53.998590 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 734:798, ack 1394, win 518, options [nop,nop,TS val 353783607 ecr 3187950], length 64
10:20:53.998590 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 734:798, ack 1394, win 518, options [nop,nop,TS val 353783607 ecr 3187950], length 64
10:20:53.999728 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 798:878, ack 1458, win 518, options [nop,nop,TS val 353783608 ecr 3187951], length 80
10:20:54.002086 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [P.], seq 878:942, ack 1538, win 518, options [nop,nop,TS val 353783609 ecr 3187951], length 64
10:20:54.002167 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [F.], seq 942, ack 1538, win 518, options [nop,nop,TS val 353783654 ecr 3187959], length 0
10:20:54.010024 IP 192.168.232.2:52752 > 192.168.248.2:ssh: Flags [.], ack 1539, win 518, options [nop,nop,TS val 353783662 ecr 3187961], length 0
10:20:54.216933 IP 192.168.232.2:52754 > 192.168.248.2:ssh: Flags [S], seq 4761298347, win 29200, options [mss 1460,sackOK,TS val 353783663 ecr 3187963], length 0

```

Βλέπουμε ότι έχουν γίνει ssh επιθέσεις.



Απομακρυσμένη σύνδεση στην βάση δεδομένων

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# mysql -h 192.168.248.2 -u root -p
```

Απομακρυσμένη σύνδεση στην βάση δεδομένων από την θύρα 3306

```
root@ubuntu:/etc/snort/rules# snort -q -A console -i ens38 -c /etc/snort/snort.conf
06/23-10:42:59.681152 [**] [1:10000:10:0] MYSQL root login attempt [**] [Priority: 0] {TCP} 192.168.232.2:57832 -> 192.168.248.2:3306
```

Προσπάθεια σύνδεσης στην βάση δεδομένων.



Οδηγίες Σύνδεσης Proxy

Με apt – get install apache2

Ενεργοποιούμε τον proxy στο apache2

```
<VirtualHost *:80>
ServerName localhost/dvwa
ProxyPreserveHost on
ProxyPass   / http://192.168.248.2/dvwa/
ProxyPassReverse / http://192.168.248.2/dvwa/
</VirtualHost>
```

Ενεργοποιούμε τον proxy και θέτουμε να επικοινωνεί με την ip του μηχανήματος **Victim** για να μπορούμε να συνδεθούμε στην ιστοσελίδα μέσω του **Router**.
Επανεκκίνηση του Apache2.

Βιβλιογραφία

[1] Andrew R. Baker, Joel Esler et al., “Snort IDS and IPS Toolkit”, Syngress Publishing, Inc., Elsevier, Inc., 2007

[2] Snort Intrusion Detection 2.02003, Pages 1-26

