ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ & ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ

Σχεδιασμός και Ανάπτυξη Μηχανισμού Αντιμετώπισης Κατανεμημένων Επιθέσεων Άρνησης Παροχής Υπηρεσίας σε Συνεργαζόμενα Δίκτυα Νέας Γενιάς

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μαρίνου Ι. Δημολιάνη

Επιβλέπων : Μάγκλαρης Βασίλειος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ & ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ

Σχεδιασμός και Ανάπτυξη Μηχανισμού Αντιμετώπισης Κατανεμημένων Επιθέσεων Άρνησης Παροχής Υπηρεσίας σε Συνεργαζόμενα Δίκτυα Νέας Γενιάς

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μαρίνου Ι. Δημολιάνη

Επιβλέπων : Βασίλειος Μάγκλαρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14η Ιουλίου 2017.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Βασίλειος Μάγκλαρης	Νεκτάριος Κοζύρης	Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2017

(Υπογραφή)

.....

ΜΑΡΙΝΟΣ ΔΗΜΟΛΙΑΝΗΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια της φοίτησής μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Θα ήθελα να ευχαριστήσω καταρχήν τον καθηγητή μου κ. Βασίλη Μάγκλαρη για την εμπιστοσύνη που μου έδειξε και την δυνατότητα που μου έδωσε να εκπονήσω την διπλωματική μου στο συγκεκριμένο πολύ ενδιαφέρον θέμα. Παράλληλα, θα ήθελα να ευχαριστήσω ιδιαιτέρως τον Αδάμ Παυλίδη για την αμέριστη βοήθεια και υποστήριξη κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Καταλήγοντας, θα ήθελα να ευχαριστήσω τους γονείς και τους φίλους μου για την συμπαράστασή και τη στήριξη τους όλα αυτά τα χρόνια.

Περίληψη

Οι δικτυακές επιθέσεις αποτελούν ένα από τα σημαντικότερα προβλήματα του Διαδικτύου. Συγκεκριμένα οι κατανεμημένες επιθέσεις άρνησης παροχής υπηρεσίας αποτελούν την πιο συνηθισμένη μορφή επίθεσης. Στόχος τους είναι να καταστήσουν έναν υπολογιστή ή μια υπηρεσία εκτός λειτουργίας. Συνεπώς οι επιθέσεις αυτές προκαλούν προβλήματα τόσο στις υπηρεσίες όσο και στο ίδιο το δίκτυο των Αυτόνομων Συστημάτων.

Κύριο αντικείμενο της παρούσης διπλωματικής εργασίας είναι η ανάπτυξη ενός μηχανισμού για την αντιμετώπιση δικτυακών επιθέσεων. Στόχος ήταν η αξιοποίηση των αμυντικών δυνατοτήτων του ίδιου του Αυτόνομου Συστήματος και η προσπάθεια για συνεργασία με άλλα Αυτόνομα Συστήματα με σκοπό την καταστολή της επίθεσης.

Ειδικότερα προτάθηκε ένας μηχανισμός επικοινωνίας-συνεργασίας μεταξύ Αυτόνομων Συστημάτων. Ο μηχανισμός αυτός βασίζεται στο πρωτόκολλο BGP και δίνει τη δυνατότητα για αίτηση συνεργασίας. Επιπροσθέτως, υλοποιήθηκε τρόπος παρακολούθησης του δικτύου μέσω του πρωτοκόλλου Netflow, με σκοπό την υποβοήθηση του συνεργατικού μοντέλου. Τέλος, αναλύθηκαν και υλοποιήθηκαν τεχνικές απόρριψης κίνησης σε δικτυακές αρχιτεκτονικές Ευφυών και Legacy Δικτύων όπως και σε συνδυασμό τους.

Ο προτεινόμενος μηχανισμός μπορεί να αποτελέσει βάση ανάπτυξης αλγορίθμων και εφαρμογής πολιτικών για διαχείριση των πόρων ενός αυτόνομου συστήματος με στόχο την αντιμετώπιση δικτυακών επιθέσεων. Η υλοποίηση του μηχανισμού είναι αμιγώς επεκτάσιμη. Για αυτό το λόγο καθίσταται δυνατός ο συνδυασμός του με τεχνικές πρόληψης και ανίχνευσης δικτυακών επιθέσεων , δίνοντας τη δυνατότητα κατασκευής ενός συνολικού εργαλείου για την συνολική καταπολέμηση τους.

Λέξεις Κλειδιά: Ευφυή-προγραμματιζόμενα δίκτυα, κατανεμημένες επιθέσεις άρνησης παροχής υπηρεσιών,Openflow, συνεργατική αντιμετώπιση, τεχνικές αντιμετώπισης επιθέσεων, Ελεγκτής Ryu

Abstract

Distributed denial of service attacks (DDoS) raise a significant threat to modern networks. Its aim is to make a machine or network service unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. These attacks can cause a lot of problems to Autonomous System's network and also to the services that are offered by.

The main purpose of the diploma thesis is the deployment of a network-attack mitigation mechanism. It takes advantage of the defensive mechanism of an Autonomous System and in parallel gives the capability for collaboration so as an attack to be distributively mitigated.

In particular, we proposed a communication-cooperation mechanism between Autonomous Systems. It is mainly based on BGP (Border Gateway Protocol) and gives the ability for requesting collaboration as far as the mitigation of a network-attack is concerned. In addition to that, we analyzed and implemented , a network monitoring technique by using Netflow protocol with a view to helping the collaboration mechanism. Finally, we delved into techniques for manipulating malicious traffic in hybrid Network Elements (Legacy and SDN environments) by exploiting each Autonomous System's capabilities.

The proposed mitigation mechanism can be used as basis for resource-centric mitigation algorithms so as an Autonomous System can defend itself against DDoS attacks. The development of the mitigation framework is also fully extendable and can be combined with detection and prevention mechanisms so that a complete self-defense tool can be deployed.

Keywords: DDoS, Openflow, SDN, Mitigation Techniques, Cooperative mitigation, Ryu Controller

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

1 Εισαγωγή

1.1 Ερευνητικό Πρόβλημα-Προσέγγιση

Ένα από τα σημαντικότερα είδη επιθέσεων που μαστίζουν σήμερα το διαδίκτυο είναι οι επιθέσεις άρνησης παροχής υπηρεσίας (DoS) και ειδικότερα οι κατανεμημένες επιθέσεις άρνησης υπηρεσίας (DDoS). Είναι ένα είδος κυβερνοεπίθεσης όπου ο δράστης προσπαθεί να βγάλει εκτός υπηρεσίας ένα μηχάνημα ή έναν δικτυακό πόρο, ώστε οι νόμιμοι (legitimate) χρήστες να μην μπορούν να το χρησιμοποιήσουν. Ο τρόπος με τον οποίο εκκινούνται επιθέσεις τέτοιου τύπου είναι η αποστολή κίνησης προς μία υπηρεσία ή έναν εξυπηρετητή μαζικά από ένα δίκτυο από bots. Ειδικότερα το δίκτυο αυτό αποτελείται από μηχανήματα που έχουν προσβληθεί από κακόβουλο λογισμικό και εν αγνοία τους συμμετέχουν στην επίθεση. Σύμφωνα με το [1] η συχνότητα των DDoS επιθέσεων έχει αυξηθεί 3 φορές τα τελευταία τρία χρόνια και ο μέσος όρος μεγέθους των επιθέσεων κυμαίνεται κοντά στο 1Gbps, αριθμός ικανός για να μπορούν να βγουν εκτός υπηρεσίας οι περισσότεροι οργανισμοί σήμερα.

Τα δίκτυα οριζόμενα από λογισμικό (SDN) διευκολύνουν τόσο τον εντοπισμό και την αναγνώριση αυτών των επιθέσεων αλλά και την αντιμετώπιση τους αφού μπορούν να επιβληθούν συγκεκριμένοι κανόνες απόρριψης των κακόβουλων ροών μέσω του ελεγκτή στον αντίστοιχο μεταγωγέα. Ωστόσο λόγω του μεγέθους των επιθέσεων είναι δύσκολη η αντιμετώπιση αυτών των επιθέσεων τόσο σε ένα στάδιο αλλά και μόνο από το αυτόνομο σύστημα που την δέχεται. Για αυτό το λόγο καθίσταται αναγκαία τόσο η συνεργασία μεταξύ των αυτόνομων συστημάτων όσο και μια πολυστρωματική (multi-layer) προσέγγιση για την αντιμετώπιση των επιθέσεων αυτών.

Ένα από τα κυριότερα ζητήματα που αφορούν τη συνεργασία μεταξύ των αυτόνομων συστημάτων είναι ο τρόπος με τον οποίο μπορεί να εξασφαλισθεί η ύπαρξη της επίθεσης σε πρώτο επίπεδο και σε δεύτερο επίπεδο ο τρόπος με τον οποίο μπορεί να αιτηθεί ένα αυτόνομο σύστημα τη βοήθεια κάποιου άλλου. Ακόμη οφείλει να ληφθεί υπόψιν ότι δεν έχουν υιοθετηθεί πλήρως και από όλους προγραμματιζόμενοι μεταγωγείς, για αυτό κρίνεται απαραίτητο να γίνεται δυνατή η συμμετοχή σε ένα συνεργατικό μηχανισμό αντιμετώπισης επιθέσεων και σε αυτόνομα συστήματα χωρίς την υποστήριξη τέτοιου είδους τεχνολογίας.

Στην παρούσα διπλωματική εργασία προτείνεται ένα αρχιτεκτονικό μοντέλο για την αντιμετώπιση των DDoS επιθέσεων όσο το δυνατό πιο κοντά στις πηγές του. Κύριος στόχος είναι η ανάπτυξη μιας εφαρμογής που θα εκτελείται στον ελεγκτή SDN και θα επιτελεί έναν αμυντικό μηχανισμό για αυτές τις επιθέσεις. Η εφαρμογή αυτή θα τρέχει σε έναν κεντρικό ελεγκτή του αυτόνομου συστήματος. Ως είσοδο σε αυτή θεωρούμε τις διευθύνσεις που στέλνουν κακόβουλη κίνηση. Κύριο μέλημα του ελεγκτή είναι η αξιοποίηση των δικτυακών συσκευών που καθιστούν δυνατή την απόρριψη κακόβουλης κίνησης. Εν συνεχεία στέλνεται αίτημα για βοήθεια σε γειτονικά συνεργαζομενα αυτόνομα συστήματα στέλνοντας σε αυτά μόνο τις διευθύνσεις οι οποίες τα αφορούν, δηλαδή τις διευθύνσεις εκείνες που αποτελούν διεύθυνση πηγής των

κακόβουλων πακέτων που διέρχονται ή που πηγάζουν από το αντίστοιχο αυτόνομο σύστημα.

1.2 Συνεισφορά εργασίας

Λόγω της ταχείας ανάπτυξης και υιοθέτησης των SDN δικτύων αλλά και της ενσωμάτωσης τους σε ήδη υπάρχοντα δίκτυα, επιχειρήθηκε η κατασκευή ενός μηχανισμού που εφαρμόζεται τόσο σε legacy αλλά και σε SDN δίκτυα, για την αντιμετώπιση DDoS επιθέσεων μέσω απόρριψης της κακόβουλης κίνησης. Κύρια προϋπόθεση για την υιοθέτηση του προτεινόμενου μηχανισμού είναι η ύπαρξη OpenFlow μεταγωγέων αλλά και δρομολογητών με συγκεκριμένα χαρακτηριστικά που αναλύονται παρακάτω.

Ειδικότερα, στην παρούσα διπλωματική εργασία, επιχειρείται αρχικά η επιλογή ενός αρχιτεκτονικού μοντέλου ικανό για την απόρριψη της κακόβουλης κίνησης με τη χρήση υπαρχουσών τεχνικών οι οποίες επεκτάθηκαν σε SDN τεχνολογία . Στη συνέχεια προτείνεται ένα σύστημα επικοινωνίας μεταξύ των ελεγκτών SDN , όσον αφορά τον τρόπο επικοινωνίας τους με σκοπό τη συνεργατική αντιμετώπιση της DDoS επίθεσης. Προτείνεται ένα σύστημα επικοινωνίας βασισμένο στο πρωτόκολλο BGP συνυφασμένο με τη χρήση κατάλληλης δομής που περιέχει λεπτομέρειες της επίθεσης. Τέλος υλοποιείται τεχνική για την παρακολούθηση της κίνησης στο δίκτυο μέσω του δρομολογητή άκρης με απώτερο σκοπό την αντιστοίχηση της κακόβουλης κίνησης με τα αυτόνομα συστήματα από τα οποία διέρχεται και είναι προσκείμενα στο αυτόνομο σύστημα που εκτελείται η εφαρμογή.

1.3 Δομή εργασίας

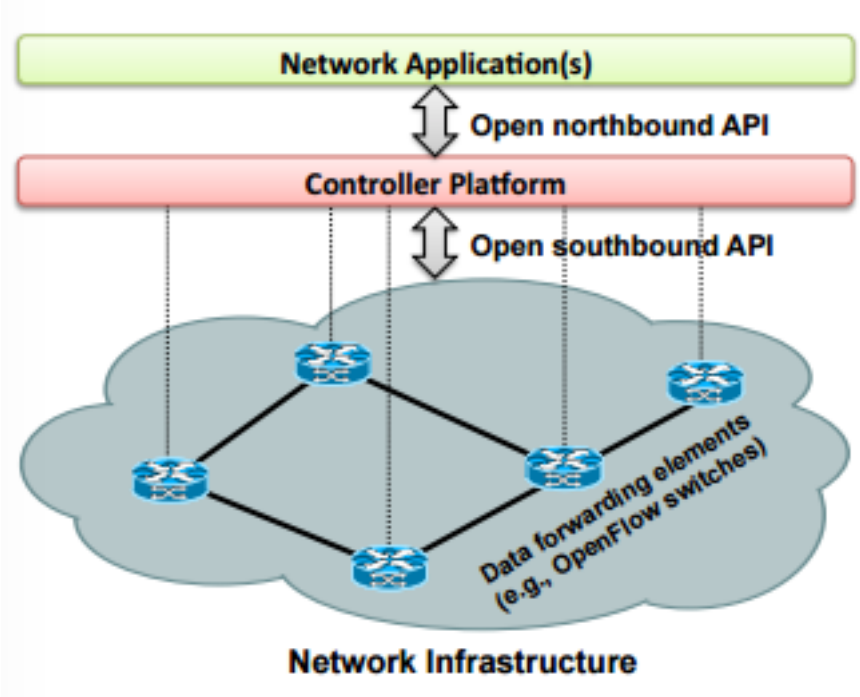
Η εργασία απαρτίζεται από εφτά κεφάλαια. Στο δεύτερο κεφάλαιο περιγράφονται συνοπτικά τα δίκτυα οριζόμενα από λογισμικό, το πρωτόκολλο Openflow καθώς και ο ελεγκτής SDN που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία στοιχεία απαραίτητα ως θεωρητικό υπόβαθρο για την κατανόηση της υπόλοιπης εργασίας. Στο τρίτο κεφάλαιο παρουσιάζονται αρχιτεκτονικά μοντέλα καθώς και τρόποι τόσο για την επικοινωνία όσο και για την συνεργασία μεταξύ αυτόνομων συστημάτων για την αντιμετώπιση DDoS επιθέσεων και τελικά προτείνεται ένα μοντέλο για την επικοινωνία μεταξύ συνεργαζόμενων αυτόνομων συστημάτων. Στο επόμενο κεφάλαιο αναλύονται ο τρόπος διεξαγωγής αλλά και τα είδη διαφόρων DDoS επιθέσεων, επίσης παρουσιάζονται και τεχνικές αντιμετώπισης τους. Στο τέλος του κεφαλαίου παρουσιάζεται η προτεινόμενη αμυντική αρχιτεκτονική. Ακολουθεί στο επόμενο κεφάλαιο ο συνολικός προτεινόμενος μηχανισμός αντιμετώπισης και αναλύονται τα χαρακτηριστικά του. Στα δύο τελευταία κεφάλαια αρχικά αναφέρονται διάφορα θέματα υλοποίησης και εν συνεχεία η πειραματική διαδικασία που ακολουθήθηκε καθώς και τα αποτελέσματα που εξάχθηκαν.

2 Θεωρητικό υπόβαθρο

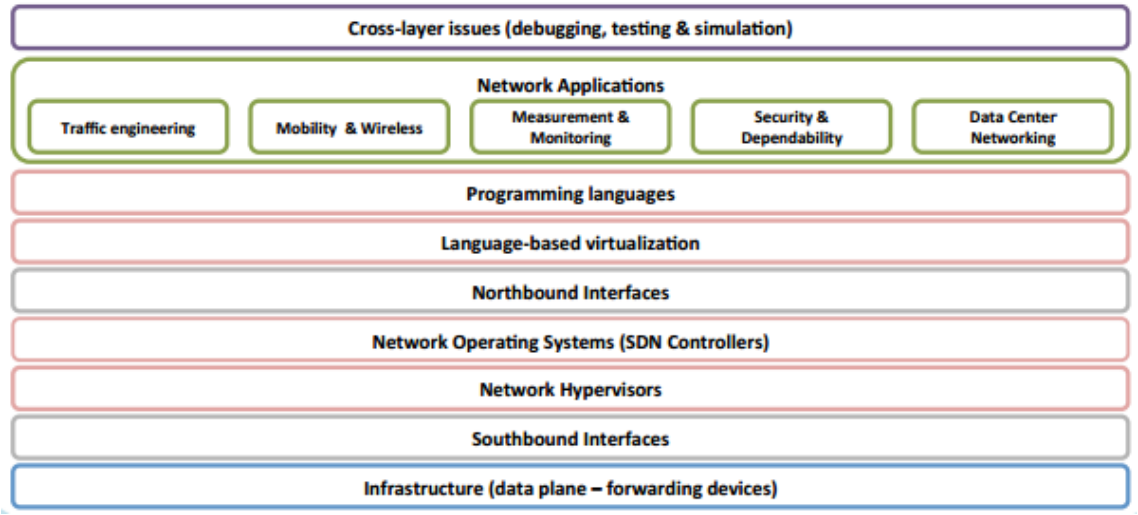
2.1 Δίκτυα οριζόμενα από λογισμικό (SDN)

Το διαδίκτυο έχει οδηγήσει στην δημιουργία ενός ψηφιακού κόσμου όπου σχεδόν τα πάντα συνδέονται και είναι προσβάσιμα σε και από οποιονδήποτε. Ωστόσο παρά την υιοθέτηση των κλασικών παραδοσιακών δικτύων IP, έχει προκύψει μεγάλη δυσκολία στη διαχείριση τους. Για να γίνει σαφέστερη η προηγούμενη δυσκολία, η ρύθμιση του δικτύου ώστε να υπακούει σε συγκεκριμένες πολιτικές και η συνεχής επαναρύθμιση του όταν αυτό είναι απαραίτητο χρήζει ιδιαίτερης προσοχής και ενέχει πολλές διαφορετικές δυσκολίες. Ο κύριος λόγος που γίνεται ακόμη πιο δύσκολη η παραπάνω διαδικασία είναι διότι οι συσκευές των δικτύων έχουν ενσωματωμένα στον τρόπο λειτουργίας τους τόσο το επίπεδο ελέγχου αλλά και το επίπεδο μεταφοράς δεδομένων. Εδώ υπεισέρχονται τα δίκτυα οριζόμενα από λογισμικό, όπου δομικά υπάρχει πλήρης διάκριση μεταξύ του επιπέδου μεταφοράς δεδομένων, από το επίπεδο ελέγχου, μεταφέροντας την ευφυΐα του δικτύου σε μία συσκευή με κεντρικοποιημένο έλεγχο και την πραγμάτωση της ευφυίας από δικτυακές συσκευές όπως δρομολογητές και μεταγωγείς. Αναφερόμενοι σε ευφυΐα στο SDN περιβάλλον, δίνεται η δυνατότητα προγραμματισμού των λειτουργιών του δικτύου έχοντας ως αποτέλεσμα την ευελιξία και την ευκολία διαχείρισης του δικτύου αλλά και την ταχύρρυθμη εξέλιξη του.

Η κύρια δομή των δικτύων οριζόμενων από λογισμικό απαρτίζεται από μία πλατφόρμα ελέγχου του δικτύου και τις συσκευές οι οποίες υλοποιούν τις εντολές που δίνονται από τον κεντρικό ελεγκτή, προωθώντας δεδομένα. Το συνηθέστερο πρωτόκολλο επικοινωνίας μεταξύ του ελεγκτή και των συσκευών αυτών είναι το πρωτόκολλο Openflow. Για να γίνει καλύτερα αντιληπτή η διάκριση του επιπέδου ελέγχου από το επίπεδο μεταφοράς των δεδομένων ο ελεγκτής επικοινωνεί με τις συσκευές του δικτύου (Openflow μεταγωγείς) μέσω μιας southbound διεπαφής προγραμματισμού εφαρμογών μέσω του πρωτοκόλλου Openflow (συνήθως) όπως αναφέρθηκε παραπάνω. Από την άλλη ο ελεγκτής επικοινωνεί με δικτυακές εφαρμογές μέσω της north bound διεπαφής προγραμματισμού. Παρακάτω παρατίθεται η αρχιτεκτονική των SDN δικτύων καθώς και μια bottom-up προσέγγιση των χαρακτηριστικών τους.



Σχήμα 1: Αρχιτεκτονική SDN δικτύων [2]



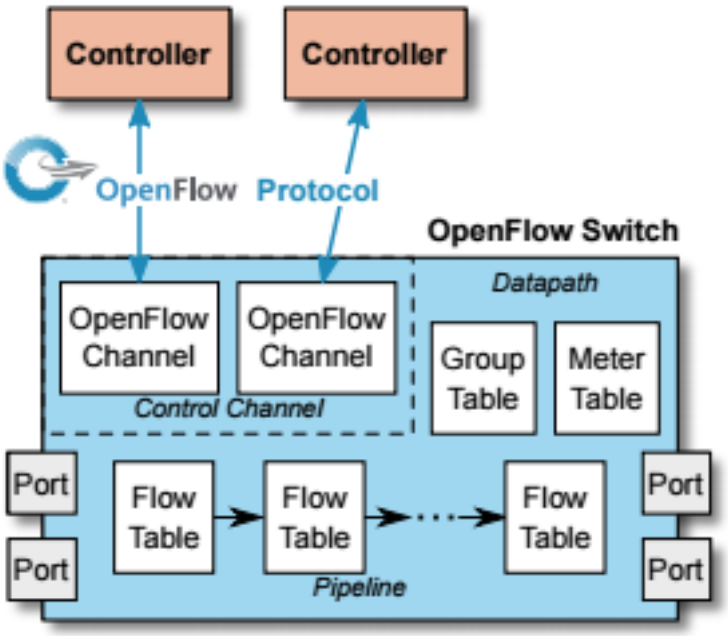
Σχήμα 2: Χαρακτηριστικά SDN δικτύων [2]

2.2 Το πρωτόκολλο OpenFlow

2.2.1 Στοιχεία του μεταγωγέα

Ο Openflow μεταγωγέας [3] αποτελείται από έναν ή περισσότερους πίνακες ροών (flow table) και έναν πίνακα ομάδας (group table) που εκτελούν την αναζήτηση και

προώθηση των πακέτων, καθώς και έναν ή περισσότερους διαύλους επικοινωνίας με έναν ή περισσότερους εξωτερικούς ελεγκτές. Ο μεταγωγέας επικοινωνεί με τον ελεγκτή και ο ελεγκτής διαχειρίζεται τον μεταγωγέα μέσω του πρωτοκόλλου Openflow. Μέσω αυτού ο ελεγκτής έχει τη δυνατότητα προσθήκης, ενημέρωσης και διαγραφής εγγραφών από τους πίνακες ροών. Κάθε πίνακας ροών περιέχει ένα σετ από εγγραφές, όπου κάθε εγγραφή αποτελείται από πεδία αντιστοίχισης (match fields), μετρητές (counters) και ένα σετ από οδηγίες (instructions) που εφαρμόζονται στα πακέτα που έχουν αντιστοιχιστεί με την εκάστοτε ροή. Η διαδικασία της αντιστοίχισης ξεκινάει στον πρώτο πίνακα ροής και μπορεί και να συνεχιστεί και στους υπόλοιπους πίνακες. Τα πακέτα αντιστοιχίζονται σε εγγραφές ροών των πινάκων ακολουθώντας σειρά προτεραιότητας, δηλαδή η πρώτη εγγραφή που αντιστοιχεί στο πακέτο που εισέρχεται χρησιμοποιείται για τη διευθέτηση του. Με τον όρο διευθέτηση εννοούμε μια σειρά οδηγιών (instructions) που μπορούν να εφαρμοσθούν στο αντίστοιχο πακέτο. Αν δεν βρεθεί κάποια εγγραφή στον πίνακα ροών τότε έχουμε table-miss και εξαρτάται από τη ρύθμιση που έχει γίνει, η κατάληξη του πακέτου. Συνήθως σε αυτή την περίπτωση το πακέτο προωθείται στον ελεγκτή μέσω του Openflow διαύλου και εκεί αποφασίζεται τι θα γίνει με το πακέτο.



Σχήμα 3: Κύρια χαρακτηριστικά του Openflow μεταγωγέα

Οι οδηγίες που σχετίζονται με κάθε εγγραφή του πίνακα ροών περιέχουν είτε (δράσεις) actions είτε τροποποιούν τη διαδικασία επεξεργασίας των πακέτων (αλλάζουν τη ροή του pipeline). Τα actions που μπορούν να εφαρμοσθούν είναι προώθηση, τροποποίηση καθώς και αποβολή του πακέτου. Από την άλλη πλευρά με την τροποποίηση του pipeline δίνεται η δυνατότητα, να σταλούν τα πακέτα σε άλλους πίνακες του μεταγωγέα όπου μπορούν να επεξεργασθούν καταλλήλως. Το pipeline του πίνακα ροών ολοκληρώνεται συνήθως με εγγραφή ροής, στην οποία δεν ορίζεται επόμενος πίνα-

κας επεξεργασίας και σε αυτό το σημείο το πακέτο τροποποιείται και προωθείται.

Οι εγγραφές ροών προωθούν τα πακέτα κατά κύριο λόγο σε μια port (θύρα) του μεταγωγέα. Οι Openflow ports είναι οι δικτυακές διεπαφές μέσω των οποίων υλοποιείται η επικοινωνία μεταξύ του μεταγωγεά και του υπόλοιπου δικτύου. Οι Openflow μεταγωγείς μπορούν να συνδεθούν μεταξύ τους για την προώθηση πακέτων όπως δύο Ethernet μεταγωγείς. Η διαδικασία που ακολουθείται συνήθως είναι η λήψη ενός πακέτου από μια port η οποία ονομάζεται ingress και η αποστολή του μέσω μιας πόρτας η οποία ονομάζεται output port. Οι ports του μεταγωγέα διακρίνονται σε 3 κατηγορίες:

- **Physical Ports:**Είναι ports ορισμένες από τον μεταγωγέα και αντικατοπτρίζονται σε πραγματική port του μεταγωγέα, λόγου χάρη σε έναν Ethernet μεταγωγέα, οι physical ports αντιστοιχίζονται μία προς μία στις Ethernet διεπαφές.
- **Logical Ports:**Είναι ports ορισμένες από τον μεταγωγέα που δεν αντικατοπτρίζονται κατευθείαν σε πραγματικό υλικό.
- **Reserved Ports:**Είναι ένα σύνολο δεσμευμένων ports που σχετίζονται με μία σειρά actions όπως αποστολή στον ελεγκτή, στην port από την οποία ήρθε το πακέτο, σε όλες τις ports κ. α.

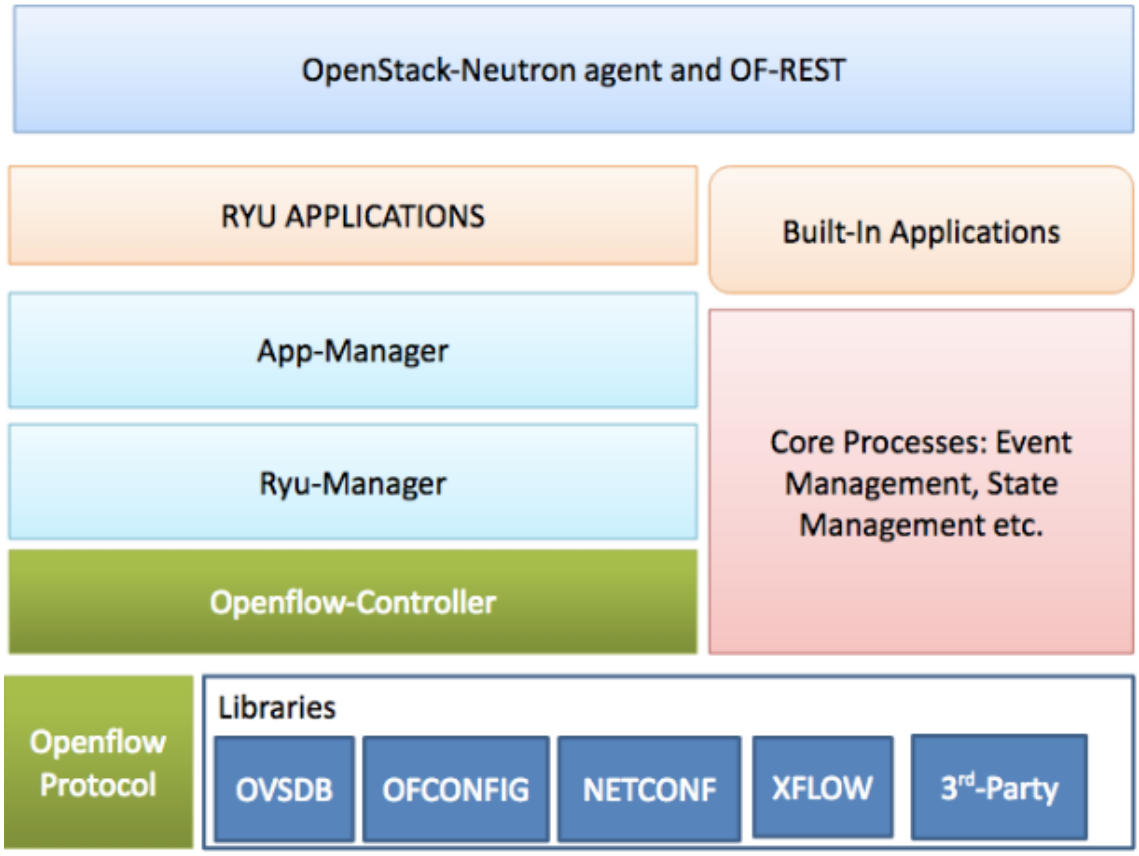
2.3 Ryu Controller

2.3.1 Γενικά Χαρακτηριστικά

Ο ελεγκτής Ryu αποτελεί μία ανοιχτού κώδικα εφαρμογή δικτύωσης που βασίζεται σε components και είναι πλήρως κατασκευασμένη σε Python. Όπως και σε άλλους ελεγκτές SDN, ο Ryu μας παρέχει μια σειρά από εφαρμογές με καλώς ορισμένες διεπαφές προγραμματισμού εφαρμογών (API) που δίνουν τη δυνατότητα στον προγραμματιστή να σχεδιάσει εύκολα νέες εφαρμογές που σχετίζονται με τη διαχείριση και τον ελέγχου του δικτύου. Ένα από τα κύρια πλεονεκτήματα του συγκεκριμένου ελεγκτή είναι η δυνατότητα υποστήριξης πολλαπλών southbound πρωτοκόλλων διαχείρισης συσκευών όπως το OpenFlow, το NETCONF, το OF-Config αλλά και άλλα.

2.3.2 Αρχιτεκτονική

Όπως ακριβώς και οποιοσδήποτε SDN ελεγκτής έτσι και ο Ryu έχει τη δυνατότητα να δημιουργεί και να στέλνει OpenFlow μηνύματα, να χειρίζεται ασύγχρονα συμβάντα όπως λόγου χάρη η αφαίρεση μιας ροής, αλλά και να αναλύει και να χειρίζεται εισερχόμενα πακέτα. Παρακάτω ακολουθεί η πλήρης αρχιτεκτονική του σκελετού του ελεγκτή Ryu:



Σχήμα 4: Ryu's Architecture

2.3.3 Βιβλιοθήκες

Ο Ryu έχει μια εντυπωσιακή συλλογή από βιβλιοθήκες που εκτείνεται από πολλαπλά southbound πρωτόκολλα όπως αναφέρθηκε και παραπάνω μέχρι διάφορες λειτουργίες επεξεργασίας πακέτων. Ο Ryu υποστηρίζει τα OF-Config, OVSDB, NETCONF, XFlow (Netflow και Sflow) και άλλα third-party πρωτόκολλα. Τα πρωτόκολλα Netflow και Sflow υποστηρίζουν λειτουργίες δειγματοληψίας και συνάθροισης πακέτων που χρησιμοποιούνται κυρίως για μετρήσεις στην κίνηση του δικτύου. Οι third-party βιβλιοθήκες που αναφέρθηκαν παραπάνω συμπεριλαμβάνουν το OpenVSwitch Python binding, τη βιβλιοθήκη για Oslo configuration και μία βιβλιοθήκη για NETCONF πελάτη. Τέλος δίνονται βιβλιοθήκες που βοηθούν στην ανάλυση και κατασκευή πακέτων όπως VLAN, MPLS, GRE και άλλα.

2.3.4 Ryu και Openflow

Ο ελεγκτής Ryu υποστηρίζει μέχρι και την έκδοση 1. 5 του πρωτοκόλλου Openflow και πέραν τούτου συμπεριλαμβάνεται στις βιβλιοθήκες του κωδικοποιητής (encoder) καθώς και αποκωδικοποιητής (decoder) για το πρωτόκολλο αυτό. Επιπρόσθετα, ένα

από τα κύρια χαρακτηριστικά του Ryu είναι ο ελεγκτής Openflow, ο οποίος είναι υπεύθυνος για τη διαχείριση των Openflow μεταγωγέων (switches) και χρησιμοποιείται για την ρύθμιση ροών (flows configuration) καθώς και για τη διαχείριση ασύγχρονων συμβάντων (asynchronous events).

2.3.5 Manager και Διεργασίες Πυρήνα

Ο manager του Ryu είναι το κύριο εκτελέσιμο αρχείο, το οποίο κατά την εκτέλεση του ακούει στην διεύθυνση IP που τρέχει ο ελεγκτής και στη θύρα 6633. Σύμφωνα με αυτό οποιοσδήποτε Openflow μεταγωγέας (είτε Open vSwitch είτε πραγματικό υλικό) μπορεί να συνδεθεί στον manager του Ryu. Συγκεκριμένα όλες οι εφαρμογές του ελεγκτή κληρονομούν από τον app manager την κλάση RyuApp. Οι διεργασίες πυρήνα χειρίζονται κατά κύριο λόγο τη διαχείριση συμβάντων, μηνυμάτων μεταξύ των εφαρμογών αλλά και την κατάσταση της μνήμης. Ένα από τα πιο ενδιαφέροντα στοιχεία της υπηρεσίας ανταλλαγής μηνυμάτων του Ryu είναι η υποστήριξη στοιχείων (components) που έχουν αναπτυχθεί σε άλλες γλώσσες προγραμματισμού πέραν της Python.

2.3.6 Ryu Northbound

Στο northbound στρώμα διεπαφής προγραμματισμού εφαρμογών (API), συμπεριλαμβάνεται ένα Openstack Neutron plug-in που υποστηρίζει και GRE αλλά και VLAN configurations. Ακόμη ο ελεγκτής υποστηρίζει μία εύχρηστη REST διεπαφή μέσω της οποίας μπορούν να υλοποιηθούν όλες οι OpenFlow λειτουργίες όπως φερειπείν η αποκόμιση των ροών του switch, η εισαγωγή κανόνων, η διαγραφή κανόνων και οποιαδήποτε άλλη ενέργεια που σχετίζεται με το πρωτόκολλο Openflow.

2.3.7 Εφαρμογές

Οι εφαρμογές του Ryu είναι μονονηματικές οντότητες που επικοινωνούν μεταξύ τους με ασύγχρονα events. Ειδικότερα κάθε εφαρμογή έχει μία FIFO ουρά λήψης events που διατηρεί τη σειρά των events. Το κύριο νήμα της εφαρμογής εξάγει events από την ουρά και καλείται η κατάλληλη συνάρτηση χειρισμού events. Όθεν μία τέτοιου είδους συνάρτηση κληθεί, δεν μπορούν να ληφθεί άλλο event από την ουρά και να επεξεργασθεί μέχρι να επιστρέψει ο έλεγχος στο κύριο νήμα (blocking fashion). Ο ελεγκτής διανέμεται με μία σειρά έτοιμων εφαρμογών οι οποίες αναφέρονται παρακάτω:

- simple_switch_1x απλός μεταγωγέας με υποστήριξη απο Openflow 1. 0 μέχρι 1. 5
- router

- firewall
- GRE tunnel
- topology (δίνει την τοπολογία των switches σε γραφικό περιβάλλον)
- simple_vlan
- BGP Speaker

2.4 Border Gateway Protocol (BGP)

Το πρωτόκολλο BGP αποτελεί τον λόγο που το Intenet λειτουργεί αφού μέσω αυτού γίνεται η δρομολόγηση μεταξύ των αυτόνομων συστημάτων [4]. Το πρωτόκολλο BGP είναι ένα τυποποιημένο και καλά εδραιωμένο πρωτόκολλο που ανήκει στην κατηγορία των πρωτοκόλλων διανύσματος μονοπατιού και επιτρέπει σε ένα αυτόνομο σύστημα να αναγγείλει την ύπαρξη του στο Διαδίκτυο. Οι αποφάσεις δρομολόγησης βασίζονται στα διαθέσιμα μονοπάτια που δημιουργούνται από την ανταλλαγή BGP μηνυμάτων. Για αυτόν το λόγο η κύρια λειτουργία ενός συστήματος που υποστηρίζει BGP είναι η ανταλλαγή πληροφορίας, που σχετίζεται με την προσβασιμότητα ενός δικτύου, με άλλα BGP συστήματα. Κρίνεται αναγκαίο να τονίσουμε ότι το λογισμικό που υλοποιεί το BGP δεν λαμβάνει αποφάσεις δρομολόγησης αλλά χρησιμοποιείται για την κατασκευή των πινάκων δρομολόγησης που στη συνέχεια χρησιμοποιούνται από τους δρομολογητές για την δρομολόγηση του δικτυακού φορτίου.

2.4.1 BGP Μηνύματα

Τα BGP μηνύματα στέλνονται πάνω από το πρωτόκολλο TCP και συγκεκριμένα η TCP συνεδρία που αφορά στο BGP γίνεται με τη χρήση της πόρτας 179. Υπάρχουν τέσσερα είδη μηνυμάτων που σχετίζονται με το BGP τα οποία αναλύουμε παρακάτω:

- **OPEN (1)**: Αφού εδραιωθεί η tcp σύνδεση μεταξύ δύο συστημάτων που υποστηρίζουν BGP το πρώτο μήνυμα που στέλνεται από κάθε πλευρά είναι ενα OPEN μήνυμα. Σε αυτό το μήνυμα περιέχονται οι εξής πληροφορίες:
  - Η έκδοση του πρωτοκόλλου που υποστηρίζεται.
  - Το αυτόνομο σύστημα του αποστολέα.
  - Ενα BGP αναγνωριστικό που συνήθως είναι η διεύθυνση IP του αποστολέα.
  - Ένα σύνολο προαιρετικών παραμέτρων. Θα αναφερθούμε στην προαιρετική παράμετρο Capabilities αφού θα μας απασχολήσει στην παρούσα διπλωματική [5] . Μέσω της συγκεκριμένης παραμέτρου (αν υποστηρίζεται) μπορεί να να ανακοινωθέι μία λίστα δυνατοτήτων που υποστηρίζεται απο

τον BGP speaker. Σε περίπτωση που υποστηρίζει ο έταιρος BGP speaker οποιαδήποτε δυνατότητα τότε σε απάντηση στο OPEN μήνυμα που έλαβε απαντάει με ένα OPEN μήνυμα με τα αντίστοιχα Capabilities που υποστηρίζει.

- **UPDATE (2):** Κύριος σκοπός για τον οποίο χρησιμοποιούνται αυτά τα μηνύματα είναι η μεταφορά πληροφορίας που σχετίζεται με τη δρομολόγηση μεταξύ BGP ομότιμων (peers). Η πληροφορία που μεταφέρεται μπορεί να χρησιμοποιηθεί για την κατασκευή ενός γράφου που περιγράφει τις σχέσεις μεταξύ των αυτόνομων συστημάτων μεταξύ τους. Τα UPDATE μηνύματα στέλνονται με σκοπό είτε την ανακοίνωση κάποιας εφικτής διαδρομής είτε την απόσυρση πολλαπλών μη εφικτών διαδρομών. Συνολικά η πληροφορία που περιέχεται σε ένα τέτοιου είδους μήνυμα είναι η εξής:
  - Οι διαδρομές που πρέπει να αποσυρθούν γιατί πλέον δεν είναι προσβάσιμες (Withdrawn Routes).
  - Ένα σύνολο ιδιοτήτων που σχετίζεται με το μονοπάτι από το οποίο έφθασε το UPDATE μήνυμα (PATH Attributes):
    * Η πηγή που μας έδωσε την πληροφορία για το μονοπάτι (IGP, EGP, άλλο).
    * Το μονοπάτι των αυτόνομων συστημάτων που διανύθηκε.
    * Το επόμενο βήμα για να φτάσουμε στη ανακοινωθείσα διεύθυνση.
    * Μια μετρική για την τοπική προτίμηση του αυτόνομου συστήματος που μας ανακοίνωσε τη διαδρομή (local preference).
    * Πληροφορία σχετικά με την επιλογή ειδικότερης διαδρομής απο αυτήν που ανακοινώθηκε.
  - Τις προσβάσιμες διευθύνσεις IP (Network Layer Information Reachability)
- **KEEPALIVE (3):** Είναι μηνύματα που στέλνονται ανά τακτά χρονικά διαστήματα μεταξύ των BGP ομότιμων και εξασφαλίζουν την ενεργή διατήρηση της BGP συνεδρίας.
- **NOTIFICATION (4):** Όταν ανιχνευθεί κάποιο λάθος που σχετίζεται με το BGP πρωτόκολλο, τότε στέλνεται ένα NOTIFICATION μήνυμα και η συνεδρία διακόπτεται αμέσως μετά την αποστολή του.

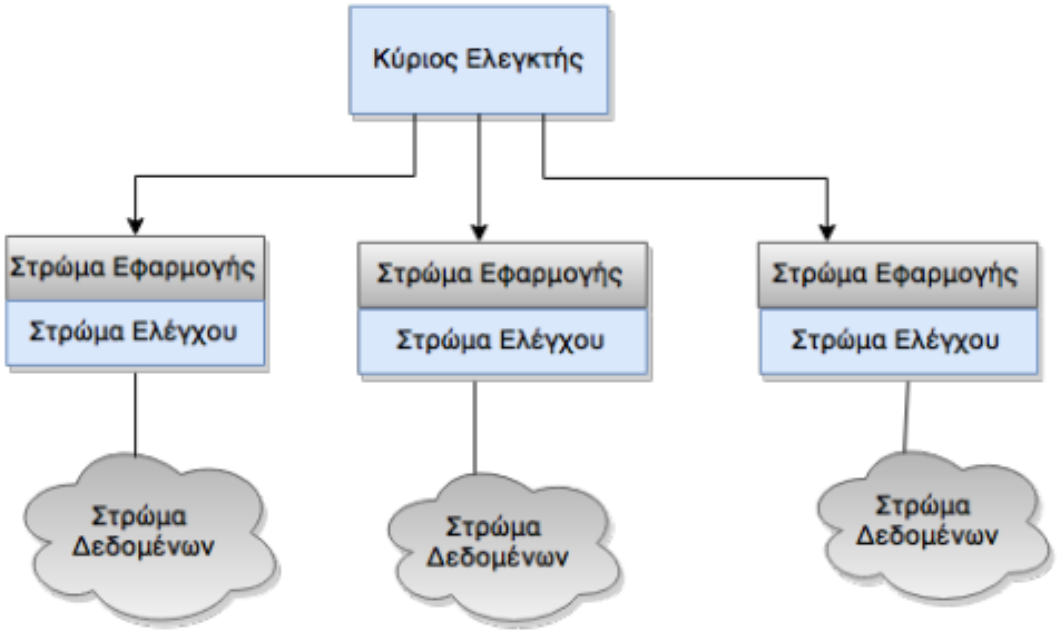
3 Επικοινωνία και συνεργασία μεταξύ αυτόνομων συστημάτων

Η επικοινωνία μεταξύ των αυτόνομων συστημάτων έχει απασχολήσει πολύ μεγάλο μέρος της ακαδημαϊκής κοινότητας. Πέραν τούτου, η εύρεση ενός μοντέλου επικοινωνίας μπορεί να δώσει τη δυνατότητα συνεργασίας μεταξύ των αυτόνομων συστημάτων. Είναι προφανές ότι ο λόγος στον οποίο αναφερόμαστε στα παραπάνω σχετίζεται με την συνεργατική αντιμετώπιση επιθέσεων. Δεν θα μπορούσε μια κατανεμημένη επίθεση άρνησης παροχής υπηρεσίας να αντιμετωπισθεί αποδοτικότερα από την συνεργασία των αυτόνομων συστημάτων από τα οποία ξεκινά, διέρχεται αλλά και καταλήγει. Συγκεκριμένα στην συνέχεια θα αναλυθούν τρόποι επικοινωνίας μεταξύ οντοτήτων των SDN δικτύων εντός και εκτός ενός αυτόνομου συστήματος.

3.1 Αρχιτεκτονικά μοντέλα

3.1.1 Το κάθετο αρχιτεκτονικό μοντέλο

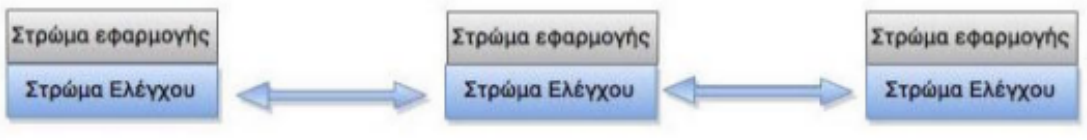
Στον κάθετο αρχιτεκτονικό μοντέλο έχουμε μία δενδρική δομή όπου ο πρόγονος των κόμβων αναλαμβάνει το διαχειριστικό συντονισμό τους. Παρόλο που αυτό το μοντέλο δίνει καλή εποπτεία του συστήματος έχει το μειονέκτημα ότι ο συνολικός συντονισμός αναλαμβάνεται από έναν κεντρικό συντονιστή. Στον SDN κόσμο, το μοντέλο αυτό θα μπορούσε να εφαρμοσθεί μόνο εντός ενός αυτόνομου συστήματος και όχι μεταξύ ”ξένων” SDN τομέων αφού δεν καθίσταται δυνατός ο ιεραρχικός συντονισμός των αυτόνομων συστημάτων από μια κοινή διαχειριστική οντότητα. Το συγκεκριμένο αρχιτεκτονικό μοντέλο εφαρμόζεται συνήθως εντός μικρών δικτύων λόγω του πλήθους των συσκευών και αποτελεί μια απλή και εύκολη λύση για τέτοιες περιπτώσεις. Παρόλα αυτά ο κίνδυνος που ελλοχεύει σε αυτή την αρχιτεκτονική είναι η αποτυχία ενός σημείου (single point failure). Αφού η διαχειριστική οντότητα είναι μοναδική οποιαδήποτε αποτυχία αυτής μπορεί να οδηγήσει στην συνολική αποτυχία του αυτόνομου συστήματος. Παρακάτω ακολουθεί σχηματικά το κάθετο αρχιτεκτονικό μοντέλο:



Σχήμα 5: Το κάθετο αρχιτεκτονικό μοντέλο

3.1.2 Το οριζόντιο αρχιτεκτονικό μοντέλο

Στο οριζόντιο αρχιτεκτονικό μοντέλο δεν υπάρχει μια ενιαία διαχειριστική οντότητα αλλά η επικοινωνία υλοποιείται σε οριζόντιο επίπεδο. Για να γίνει καλύτερα αντιληπτό αυτό, σε επίπεδο SDN τομέων οι ελεγκτές των αυτόνομων συστημάτων συμμετέχουν στο εκάστοτε αρχιτεκτονικό μοντέλο του αυτόνομου συστήματος αλλά διατηρούν και μεταξύ τους επικοινωνία ανά δυο . Η επικοινωνία αυτή μπορεί να παραλληλισθεί με την αντίστοιχη σχέση των αυτόνομων συστημάτων (πελάτης-πάροχος, πάροχος-πάροχος κ. ο. κ). Αυτό το αρχιτεκτονικό μοντέλο υπερέχει σε επίπεδο κλιμακωσιμότητας του προηγούμενου και σίγουρα δεν επηρεάζεται η αποτελεσματικότητα και η απόδοση του συνολικού συστήματος. Το οριζόντιο αρχιτεκτονικό μοντέλο μπορεί να απεικονισθεί ως εξής:



Σχήμα 6: Το οριζόντιο αρχιτεκτονικό μοντέλο

3.2 Παρακολούθηση κίνησης στο δίκτυο

3.2.1 Netflow

Το Netflow είναι ένα πρωτόκολλο που αρχικά προτάθηκε από την Cisco στους δρομολογητές της, με σκοπό τη συλλογή της κίνησης που εξέρχεται ή εισέρχεται από ένα ή σε ένα interface του δρομολογητή. Ωστόσο πλέον έχει προτυποποιηθεί και υποστηρίζεται κι από άλλους κατασκευαστές. Τα δεδομένα που παρέχονται στον διαχειριστή του δικτύου μέσω του Netflow αφορούν λεπτομέρειες της κίνησης του δικτύου στις οποίες θα αναφερθούμε αναλυτικότερα παρακάτω και βοηθούν στην παρακολούθηση του. Το σύνηθες σενάριο ενεργοποίησης του Netflow, είναι η ενεργοποίηση του στα interfaces των δρομολογητών, στη συνέχεια η εξαγωγή των δεδομένων αυτών στο συλλέκτη της κίνησης και τέλος η ανάλυση της προκύπτουσας πληροφορίας. Συνεπώς η απαραίτητη αρχιτεκτονική προϋποθέτει μία συσκευή μέσω της οποίας εξάγεται η πληροφορία μέσω του πρωτοκόλλου Netflow, ένας συλλέκτης για την λήψη των δεδομένων, μία βάση δεδομένων στην οποία αποθηκεύονται τα δεδομένα και μια εφαρμογή μέσω της οποίας διαχειριζόμαστε τα δεδομένα προς όφελός μας.

Η ενεργοποίηση του Netflow γίνεται ανά δρομολογητή ανά interface και με τον ίδιο τρόπο ορίζεται και η διεύθυνση καθώς και η θύρα στην οποία θα σταλούν τα αντίστοιχα δεδομένα. Συνήθως η μεταφορά των δεδομένων γίνεται πάνω από το πρωτόκολλο UDP, ωστόσο λόγω της φύσης του (μη ενημέρωση για την απώλεια πακέτων) σε κάποιες σύγχρονες υλοποιήσεις χρησιμοποιείται το πρωτόκολλο SCTP (Stream Control Transmission Protocol), για την αποφυγή απώλειας πακέτων. Η εξαγωγή της πληροφορίας στον συλλέκτη γίνεται είτε όταν μία ροή (flow) είναι ανενεργή για κάποιο χρονικό διάστημα, είτε ενεργή για μεγαλύτερο από κάποιο ορισμένο χρονικό διάστημα. Τα όρια αυτά μπορούν να ρυθμιστούν στον δρομολογητή και όταν η αντίστοιχη πληροφορία σταλεί, διαγράφεται οριστικά από την μνήμη του δρομολογητή. Η μνήμη στην οποία αποθηκεύονται οι πληροφορίες που σχετίζονται με το Netflow είναι μία μνήμη cache στον δρομολογητή με χαρακτηριστικά που μπορούν να ρυθμιστούν και αυτά από τον διαχειριστή του δικτύου.

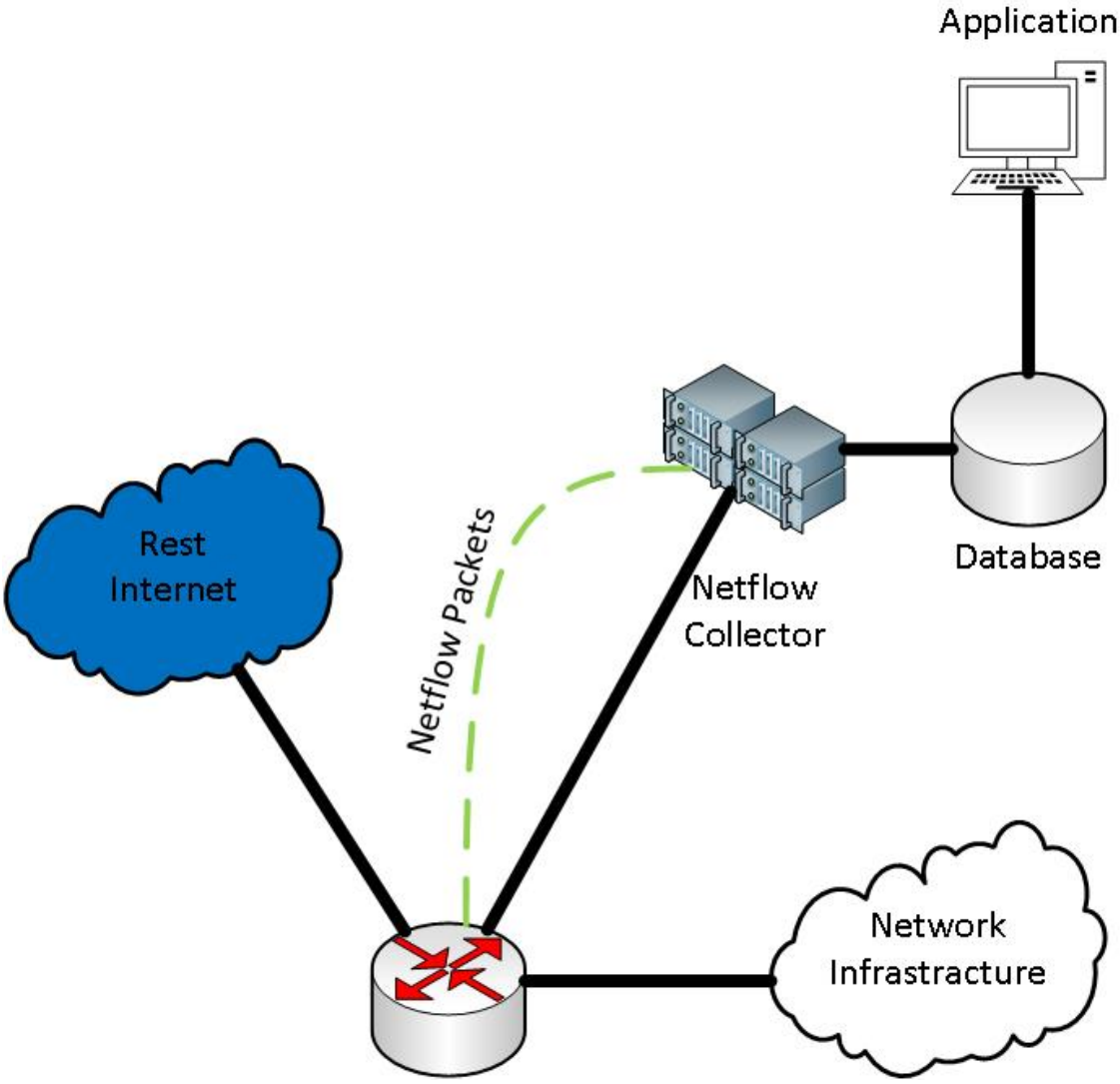
Οι πιο διαδεδομένες εκδόσεις Netflow είναι η έκδοση 5 και η έκδοση 9, ωστόσο στην παρούσα διπλωματική εργασία θα αναφερθούμε αναλυτικά στην έκδοση 5, αφού αυτή χρησιμοποιήθηκε. Περισσότερες πληροφορίες σχετικά με την έκδοση 9 μπορούν να βρεθούν στο [18]. Η δομή του πακέτου Netflow διακρίνεται σε δύο στοιχεία στην επικεφαλίδα και στις εγγραφές ροών. Συγκεκριμένα η επικεφαλίδα του πακέτου περιέχει πληροφορίες σχετικά με:

- Την έκδοση του πρωτοκόλλου
- Τον αριθμό ακολουθίας του πακέτου
- Τον χρόνο εξαγωγής του πακέτου όπως αυτός προέκυψε από τον δρομολογητή
- Τον αριθμό των ροών που ακολουθούν

Ενώ η εγγραφή για τις ροές περιέχει τα εξής στοιχεία:

- Το interface εισόδου όπως αυτό αναγράφεται στο πρωτόκολλο SNMP
- Το interface εξόδου ή τον αριθμό μηδέν αντί αυτού αν το πακέτο απορρίφθηκε
- Τις χρονικές στιγμές εκκίνησης και λήξης της ροής
- Τον αριθμό των πακέτων και των bytes που παρατηρήθηκαν στην αντίστοιχη ροή
- Τη διεύθυνση πηγής και προορισμού
- Τον τύπο και τον κωδικό του ICMP
- το πρωτόκολλο IP και την τιμή TOS (Type of Service)
- Τις θύρες πηγής και προορισμού των πρωτοκόλλων TCP, UDP, SCTP
- Αν πρόκειται για TCP ροές, περιέχονται και πληροφορίες για τις σημαίες των TCP πακέτων
- Δίνεται η δυνατότητα παροχής πληροφορίας σχετικά με το αυτόνομο σύστημα από το οποίο προήλθε το πακέτο (είτε το άμεσα γειτονικό, είτε το αυτόνομο σύστημα από το οποίο ξεκίνησε το πακέτο)

Παρακάτω ακολουθεί σχηματικά το αρχιτεκτονικό περιβάλλον στο οποίο μπορεί να χρησιμοποιηθεί το πρωτόκολλο Netflow:

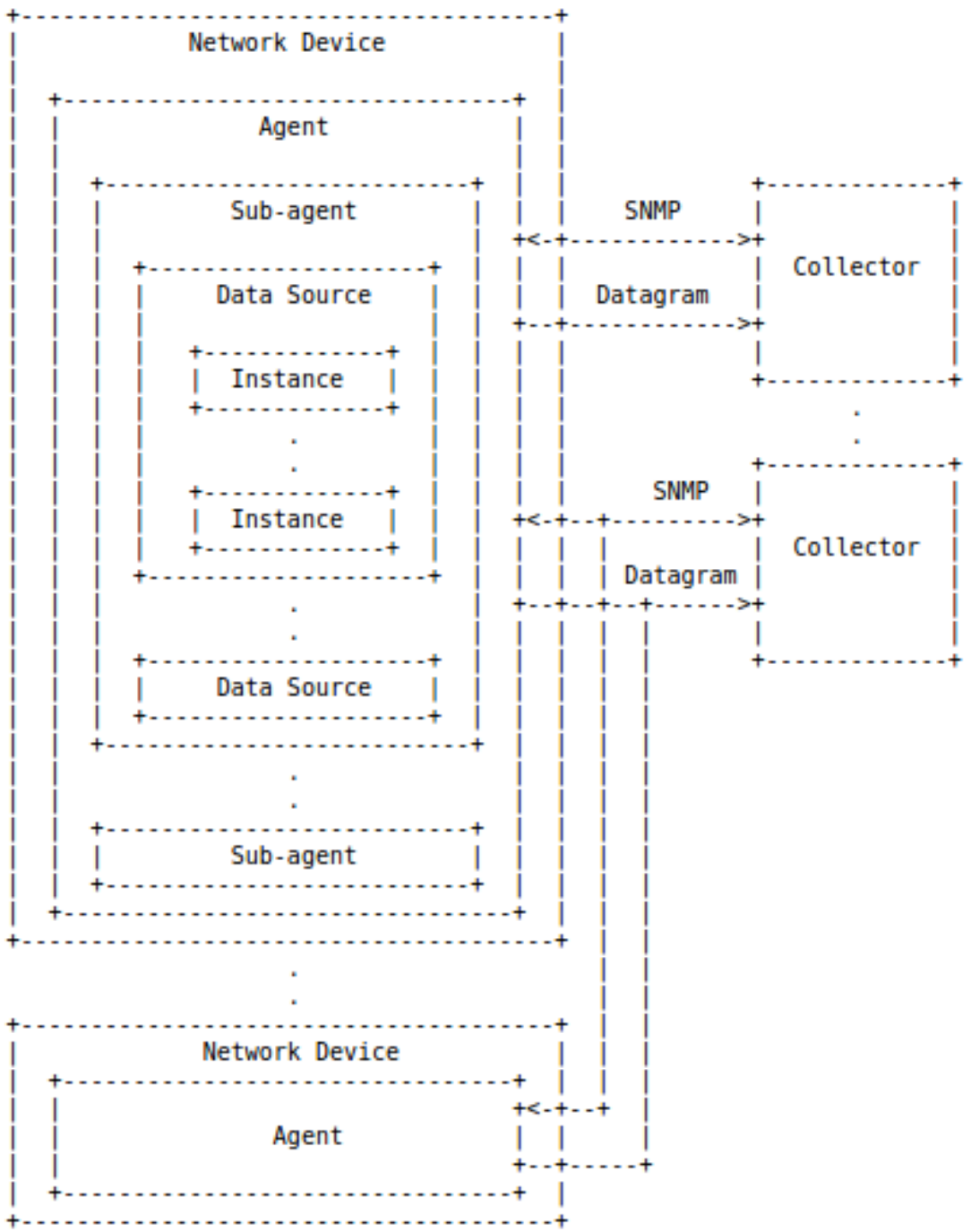


Σχήμα 7: Αρχιτεκτονική παρακολούθησης δικτύου με το πρωτόκολλο Netflow

3.2.2 sFlow

Το πρωτόκολλο sFlow (sampled flow) παρέχει όπως και το πρωτόκολλο Netflow τη δυνατότητα συλλογής πληροφοριών της κίνησης που διέρχεται από κάποια συσκευή που το υποστηρίζει (δρομολογητής, μεταγωγέας, Openflow μεταγωγέας κ. α.). Η βασική διαφορά με το πρωτόκολλο Netflow έγκειται στον τρόπο συλλογής δεδομένων αφού όπως αναφέρει και το όνομα του πρωτοκόλλου πρόκειται για δειγματοληπτική μέθοδο. Ο λόγος που χρησιμοποιείται η δειγματοληψία είναι για κλιμακωσιμότητα αφού δεν τίθεται δυνατή η παρακολούθηση πακέτων σε αναλογία 1:1 σε δίκτυα υψηλών ταχυτήτων (Gigabit /sec). Δεν κρίνεται απαραίτητο να αναφερθούμε παραπάνω στο πρωτόκολλο sFlow, αφού δεν χρησιμοποιήθηκε στην παρούσα διπλωματική ωστόσο μπορούν να βρεθούν περισσότερες λεπτομέρειες στο [19], οι οποίες δεν μας απασχόλησαν στο δικό μας αρχιτεκτονικό μοντέλο. Παρακάτω ακολουθεί η

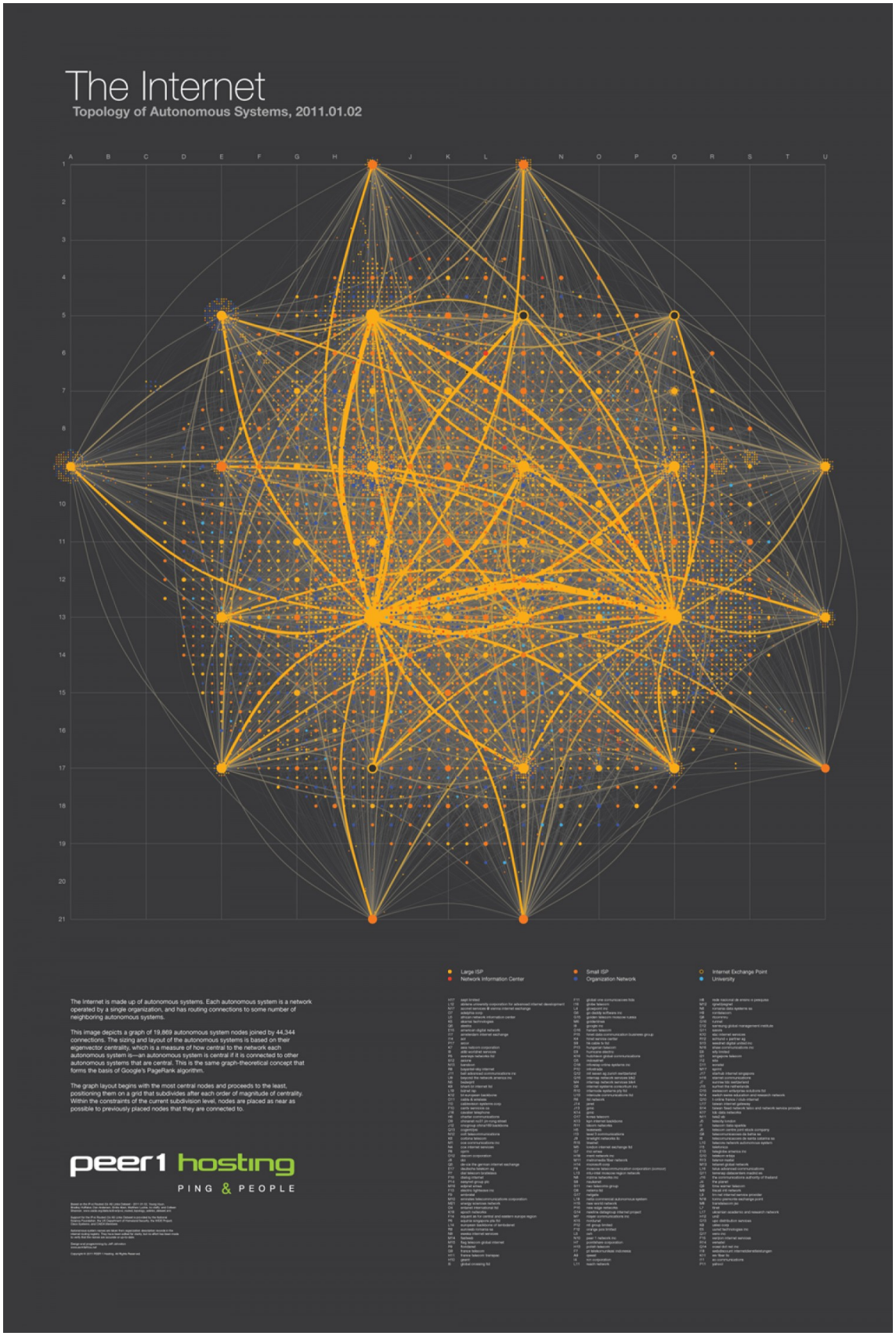
αρχιτεκτονική υλοποιήσης του sFlow:



Σχήμα 8: Αρχιτεκτονική παρακολούθησης δικτύου με το πρωτόκολλο sFlow

3.3 Συστήματα εμπιστοσύνης

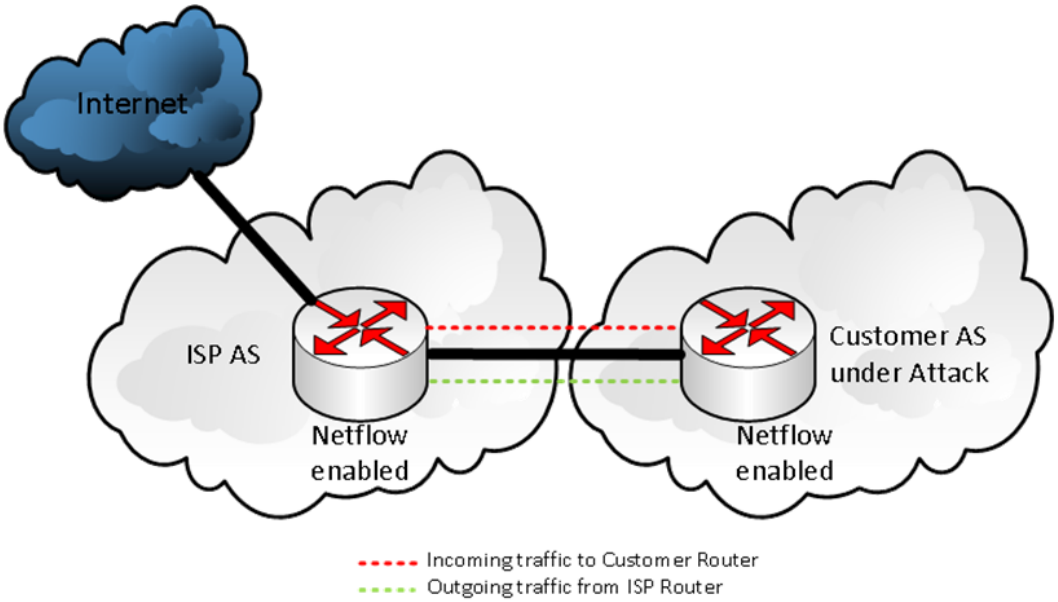
Η δομή του διαδικτύου, η ύπαρξη των αυτόνομων συστημάτων δηλαδή καθώς και οι σχέσεις που αυτά έχουν μεταξύ τους μπορεί να παραλληλισθεί με τα ομότιμο προς ομότιμο δίκτυα (p2p networks). Έχουν μελετηθεί κατ' εξοχήν τα δίκτυα αυτά ως προς την εύρεση των ιδανικότερων αλγορίθμων για την εμπιστοσύνη μεταξύ των ομότιμων αλλά και για τη δημιουργία διαφόρων μοντέλων φήμης όπως στο [28]. Παρόλα αυτά, αυτοί οι μηχανισμοί χρησιμοποιούνται κυρίως σε δίκτυα τα οποία έχουν ως κύριο σκοπό το διαμοιρασμό αρχείων. Παρότι έχουν γίνει προσπάθειες ώστε να μην μπορούν τέτοιου είδους συστήματα να χρησιμοποιηθούν κακοβούλως και συγκεκριμένα να διαμοιραστούν αρχεία με κακόβουλο λογισμικό, αφού οι μηχανισμοί αυτοί αποτελούν συστήματα εμπιστοσύνης-φήμης αν κάποιος ομότιμος αλλάξει άρδην στάση και αρχίσει να στέλνει κακόβουλο λογισμικό λόγω της "καλής" φήμης που έχει το αρχείο διαμοιράζεται. Παρακάτω παρατίθεται ένα μέρος του σημερινού διαδικτύου με 19, 689 αυτόνομα συστήματα που ενώνονται με 44, 344 συνδέσεις μεταξύ τους. Ο λόγος που το παρουσιάζουμε είναι για να είναι εμφανής η πολυπλοκότητα και η p2p φύση της δομής του.



Σχήμα 9: Η σημερινή τοπολογία του διαδικτύου σε αυτόνομα συστήματα

Στην προκείμενη περίπτωση, αναφερόμαστε σε ομότιμο προς ομότιμο SDN τομέα σε ξένα αυτόνομα συστήματα. Στην συγκεκριμένη περίπτωση η ανταλλαγή της πληροφορίας αφορά διευθύνσεις από τις οποίες δεχόμαστε κακόβουλη κίνηση. Θεωρώντας ότι χρησιμοποιούμε ένα σύστημα εμπιστοσύνης όπως στο [20] μπορούμε να έχουμε μια ένδειξη για τη φήμη του άλλου SDN τομέα, παρόλα αυτά ένας τέτοιος μηχανισμός μπορεί να αξιοποιηθεί κακοβούλως και βάσει φήμης να αιτηθεί την απόρριψη κίνησης από καλόβουλους χρήστες. Αν αυτός ο τομέας είναι έμπιστος τότε βάσει της μετρικής που έχουμε θα απορρίψουμε καλόβουλη κίνηση και όχι μόνο αυτό αλλά θα προωθήσουμε το αίτημα και περαιτέρω, με πιθανότητα να δημιουργηθεί μεγάλο πρόβλημα στο διαδίκτυο.

Είναι αναγκαίο σε μηχανισμούς όπου απαιτείται η συνεργασία μεταξύ ομότιμων να υπάρχει ένας τρόπος εξασφάλισης της εμπιστοσύνης που υπάρχει μεταξύ των συνεργαζόμενων μονάδων. Συγκεκριμένα έχουν προταθεί διάφορα συστήματα φήμης-εμπιστοσύνης για σενάρια παρόμοια με τον προτεινόμενο μηχανισμό. Ωστόσο οι προτάσεις αυτές κυρίως κινούνται στα πλαίσια της προαναφερθείσας λογικής που σχετίζεται με μοντέλα φήμης-εμπιστοσύνης. Για αυτό το λόγο προτείνεται η αξιοποίηση του Netflow στο δίκτυο. Ειδικότερα το Netflow χρησιμοποιείται ούτως η αλλιώς για την παρακολούθηση του δικτύου. Εκμεταλλευόμενοι την ύπαρξη του μπορούμε να δημιουργήσουμε μια δικλείδα ασφαλείας για την ύπαρξη της επίθεσης. Συγκεκριμένα είναι αναπόφευκτο η κίνηση που εξέρχεται από έναν δρομολογητή άκρης να μην καταλήξει στο δρομολογητή άκρης του συνδεόμενου σε αυτό αυτόνομου συστήματος. Συνεπώς μέσω του πρωτοκόλλου Netflow καθίσταται δυνατή η επιβεβαίωση της διέλευσης της κίνησης και δη της κακόβουλης, για την οποία ζητά το αιτών αυτόνομο σύστημα βοήθεια. Στην παρακάτω εικόνα γίνεται σαφέστερη η προσέγγιση μας:



Σχήμα 10: Επαλήθευση διέλευσης κίνησης με χρήση του Netflow

3.4 Προτυποποίηση δεδομένων

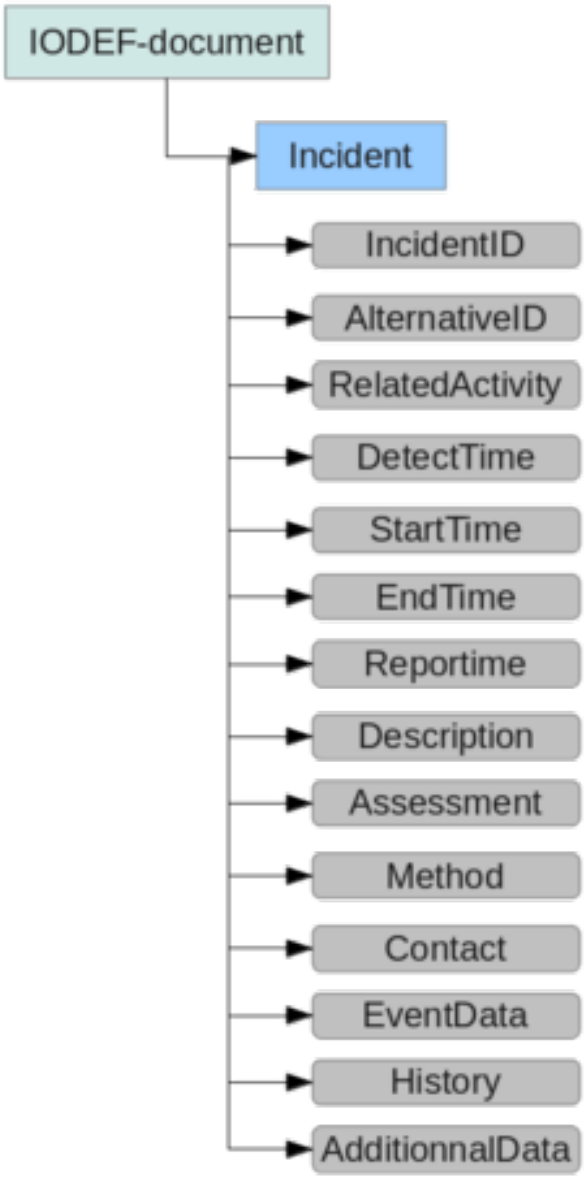
3.4.1 JSON (JavaScript Object Notation)

Το JSON αποτελεί ένα πρότυπο δεδομένων σε μορφή αρκετά φιλική και ευανάγνωστη από τον άνθρωπο. Η κύρια χρήση του είναι η μεταφορά δεδομένων δομημένα σε αντικείμενα υπό τη μορφή ζευγαριών γνωρίσματος-τιμής. Θεωρείται ο πιο σύγχρονος τρόπος ανταλλαγής δεδομένων για ασύγχρονη επικοινωνία μεταξύ περιηγητή και εξυπηρετητή και τα τελευταία χρόνια τείνει να αντικαταστήσει το πρότυπο XML. Ένα από τα κυριότερα πλεονεκτήματα του είναι ότι αποτελεί μια μορφή δεδομένων ανεξάρτητη από τη γλώσσα προγραμματισμού στην οποία χρησιμοποιείται. Παρότι υιοθετήθηκε από την γλώσσα προγραμματισμού Javascript, αρκετές γλώσσες προγραμματισμού περιέχουν έτοιμες βιβλιοθήκες που δίνουν τη δυνατότητα χειρισμού και επεξεργασίας δεδομένων σε JSON μορφή. Η δομή του JSON αποτελείται από μία σειρά από τέσσερις τύπους δεδομένων και δύο δομές δεδομένων. Συγκεκριμένα μπορούν να αναπαρασταθούν σύμφωνα με το [23] σε επίπεδο τύπων συμβολοσειρές, αριθμοί, boolean και το null ενώ σε επίπεδο δομών δεδομένων πίνακες και αντικείμενα. Η δομή JSON χρησιμοποιήθηκε εκτενώς στην συγκεκριμένη διπλωματική εργασία αφού τόσο η βάση δεδομένων που χρησιμοποιήθηκε αποθηκεύει τα δεδομένα της σε JSON format αλλά και η προτυποποίηση των δεδομένων που ανταλ-

λάσσονται μεταξύ των SDN τομέων των αυτόνομων συστημάτων χρησιμοποιούν μία αναφορά τύπου IODEF σε JSON μορφή.

3.4.2 IODEF (The Incident Object Description Format)

Η ύπαρξη μιας κοινής προτυποποιήσης δεδομένων για την περιγραφή συμβάντων που σχετίζονται με επιθέσεις στο διαδίκτυο καθίσταται αναγκαία. Συγκεκριμένα απαιτείται μία δομή δεδομένων η οποία θα παρέχει πληροφορίες για την επίθεση και η οποία θα μπορεί να προσπελαστεί μεθοδικά και να ανακτηθούν οι επιθυμητές πληροφορίες. Η αναφορά IODEF αποτελεί χαρακτηριστικό παράδειγμα, αφού είναι μια μορφή αναπαράστασης δεδομένων που σχετίζονται με θέματα ασφάλειας και ανταλλάσσονται από ομάδες CSIR (Computer Security Incident Response). Στα πλαίσια της παρούσας διπλωματικής επεκτάθηκε το πρότυπο της αναφοράς IODEF στις ανάγκες του προτεινόμενου μηχανισμού. Επίσης πέραν τούτου χρησιμοποιήθηκε η αναπαράσταση σε μορφή JSON [24] του προτύπου IODEF και όχι σε μορφή XML, λόγω του ότι η βάση δεδομένων που χρησιμοποιήθηκε δέχεται δεδομένα σε JSON μορφή και υπήρχε πλήρης αντιστοιχία του τρόπου αποθήκευσης και της αναφοράς IODEF. Ακολουθούν τα πεδία της IODEF αναφοράς και πως επεκτάθηκαν για τις ανάγκες του συστήματος που σχεδιάστηκε, αρχικά σε γενικό επίπεδο στην παρακάτω εικόνα και έπειτα αναλυτικότερα.



Σχήμα 11: Το πρότυπο IODEF

Τα πεδία από τα οποία απαρτίζεται η αναφορά IODEF είναι τα εξής:

- **IncidentID:** Το πεδίο αυτό περιέχει τον αριθμό που χρησιμοποιείται ως μοναδικό αναγνωριστικό της συγκεκριμένης αναφοράς. Για αυτό το λόγο απαιτείται η κατασκευή του συγκεκριμένου αρχείου να γίνεται με γνώμονα τη δημιουργία διαφορετικών IDs όταν πρόκειται για διαφορετικές αναφορές IODEF.
- **AlternativeID:** Το πεδίο αυτό βάσει του προτύπου IODEF χρησιμοποιείται για την εύρεση του ίδιου περιστατικού από μια άλλη ομάδα από αυτήν που έχει παράξει το αρχείο. Στην δική μας υλοποίηση χρησιμοποιείται για την μεταφορά

του αναγνωριστικού της αρχικής αναφοράς όπως αυτή παρήχθη από τον τομέα που αναγνώρισε την επίθεση και αιτήθηκε βοήθεια για την αντιμετώπιση της. Μπορεί να θεωρηθεί σαν ένας τρόπος ένδειξης της αρχικής επίθεσης αλλά όχι απόδειξης της.

- **Related Activity:** Το πεδίο αυτό σύμφωνα με το πρότυπο IODEF χρησιμοποιείται για την μεταφορά του URL κάποιου άλλου γεγονότος που σχετίζεται με το τρέχον συμβάν. Ωστόσο, στην δική μας προσέγγιση μεταφέρει το url του τομέα στον οποίο αν ζητήσουμε το AlternativeID θα λάβουμε την αρχική αναφορά.

- **Report Time:** Το πεδίο αυτό αναφέρει την χρονική στιγμή όπου παράχθηκε η συγκεκριμένη αναφορά. Συγκεκριμένα στην δική μας περίπτωση κατά την δημιουργία του αρχείο τοποθετούμε την τρέχουσα χρονική στιγμή που μας δίνεται μέσω της γλώσσας προγραμματισμού που χρησιμοποιήθηκε.

- **Description:**Όπως αναφέρει και το όνομα του πεδίου, εδώ περιγράφεται σε ελεύθερη μορφή οτιδήποτε χρίζει αναφοράς σχετικά με το συμβάν.

- **Assesment:** Το πεδίο αυτό περιγράφει τις επιπτώσεις που έχει το συμβάν που περιγράφεται στον τομέα ο οποίος σύνταξε την τρέχουσα αναφορά. Ωστόσο το πεδίο αυτό αποτελεί μια κλάση από στοιχεία τα οποία μπορούν αρκετές φορές να είναι χρήσιμα για την εκτενέστερη περιγραφή του συμβάντος. Συγκεκριμένα εμπεριέχονται τα χαρακτηριστικά:

  - **Impact:**Το πεδίο αυτό έχει πεδία που σχετίζονται με την επικινδυνότητα του συμβάντος, το αποτέλεσμα που είχε το συμβάν (πέτυχε ή απέτυχε), το είδος της επίθεσης και άλλα που δεν θα μας απασχολήσουν περαιτέρω.
  - **TimeImpact:**Όπως είναι εμφανές από το όνομα του πεδίου, εδώ περιγράφονται χαρακτηριστικά που σχετίζονται με τη χρονική επίδραση του συμβάντος όπως το χρόνο που διήρκεσε, το χρόνο που χρειάστηκε για να αντιμετωπισθεί καθώς και ένα πεδίο που αναφέρει τις μονάδες που προσμετρώνται η παραπάνω χρόνοι.
  - **MonetaryImpact:**Αυτό το πεδίο περιγράφει τις οικονομικές συνέπειες που είχε η επίθεση στο σύστημα που τη δέχθηκε και αναφέρεται η σοβαρότητα της επίθεσης αλλά και η ορθότητα της ανίχνευσης της επίθεσης.

- **Method:**Αυτό το πεδίο μεταφέρει τη μέθοδο που χρησιμοποιήθηκε για την επίθεση.

- **History:**Σύμφωνα με το RFC στο πεδίο αυτό χρησιμοποιείται για την ενημέρωση του παραλήπτη σχετικά με μια σειρά ενεργειών-γεγονότων που έχουν προηγηθεί στο παρελθόν για την αντιμετώπιση του συμβάντος.

- **EventData:**Η κλάση αυτή περιγράφει ένα συγκεκριμένο event του περιστατικού για ένα πεπερασμένο αριθμό hosts ή δικτύων. Η περιγραφή αυτή συμπεριλαμβάνει τα συστήματα από τα οποία προέρχεται η συγκεκριμένη δραστηριότητα αλλά και οι στόχοι της. Στην περίπτωση μας χρησιμοποιήθηκε για την μεταφορά των διευθύνσεων που βρέθηκαν από το σύστημα ανίχνευσης του δικτύου

μας, ότι σχετίζονται με την επίθεση και ειδικότερα είναι οι πηγές της. Παρόλα αυτά ανάλογα με την επίθεση υπάρχει πιθανότητα να σταλεί πέραν της διεύθυνσης πηγής και η διεύθυνση του στόχου, για περιπτώσεις όπου χρειάζονται ειδικότεροι κανόνες απόρριψης πακέτων. Να τονίσουμε σε αυτό το σημείο ότι η μεταφορά πληροφορίας στο πεδίο EventData μπορεί να προσαρμοσθεί ανάλογα με τα ιδιαίτερα χαρακτηριστικά και να μεταφερθούν και άλλα στοιχεία, όπως τύπος πρωτοκόλλου, θύρα εισόδου-εξόδου κ. α.

```
from bson import ObjectId
from pymongo import MongoClient
from time import gmtime, strftime

class IodefJson():
    ID=0
    def __init__(self,alternative_id,related_activity,description,
                 method,datetime,additional_data,ip_sources):
        IodefJson.ID=IodefJson.ID+1
        self.id=IodefJson.ID
        self.aid=alternative_id
        self.related_activity=related_activity
        self.report_time=strftime("%Y-%m-%d %H:%M:%S", gmtime())
        self.description=description
        self.method=method
        self.datetime=datetime
        self.additional_data=additional_data
        self.ip_sources=ip_sources
    def toJson(self):
        iodefjson={
            'ID' : self.id,
            'AlternativeID': self.aid,
            'RelatedActivity':self.related_activity,
            'ReportTime':self.report_time,
            'Description':self.description,
            'Method':self.method
            'History':[{'DateTime':self.datetime,
                        'AdditionalData':self.additional_data}],
            'EventData':self.ip_sources


        }
        return iodefjson
```

Σχήμα 12: Η υλοποίηση του προτύπου iodef σε json format στη γλώσσα Python

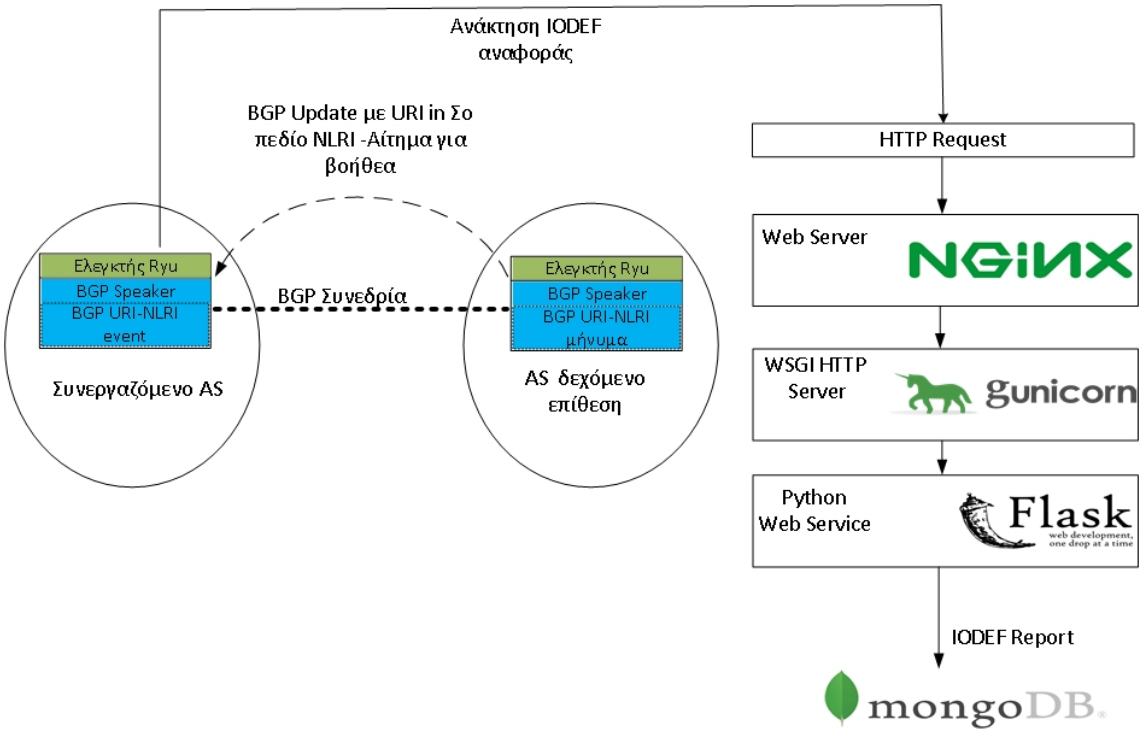
3.5 Προτεινόμενο αρχιτεκτονικό μοντέλο επικοινωνίας

Βασιζόμενοι στο οριζόντιο αρχιτεκτονικό μοντέλο που περιγράφηκε παραπάνω αλλά και στα [20], [21], [22] καταλήξαμε στο παρακάτω αρχιτεκτονικό μοντέλο για

την επικοινωνία-συνεργασία μεταξύ ελεγκτών διαφορετικών αυτόνομων συστημάτων.

Προϋπόθεση για την υλοποίηση αυτού του μοντέλου είναι η υποστήριξη BGP Speaker από τον εκάστοτε ελεγκτή. Οι ελεγκτές δημιουργούν μεταξύ τους μια BGP συνεδρία, στην οποία όμως δεν ανταλλάσσουν μηνύματα που αφορούν στην προσβασιμότητα του δικτύου αλλά σχετίζονται με αίτηση βοήθειας για την αντιμετώπιση μιας κατανεμημένης επίθεσης άρνησης παροχής υπηρεσιών. Ο τρόπος με τον οποίο μπορεί να υλοποιηθεί αυτό είναι η επέκταση του πρωτοκόλλου BGP ώστε να υποστηρίζει τη μεταφορά URI's στο πεδίο NLRI όπως αυτό έχει προταθεί στο [22]. Ειδικότερα οι ελεγκτές δημιουργούν μεταξύ τους μια BGP συνεδρία μέσω της οποίας υποστηρίζεται η μεταφορά uri μέσω πακέτων BGP Update και συγκεκριμένα στο πεδίο NLRI. Για να καταλάβει ο ομότιμος BGP ότι αφορά τέτοιου είδους μήνυμα απαιτείται τα πεδία Tolal Path Attribute Length και Withdrawn Routes Length να έχουν μέγεθος 0. Κατά αυτόν τον τρόπο αντιλαμβάνεται ο ομότιμος ότι πρόκειται για αίτημα βοήθειας.

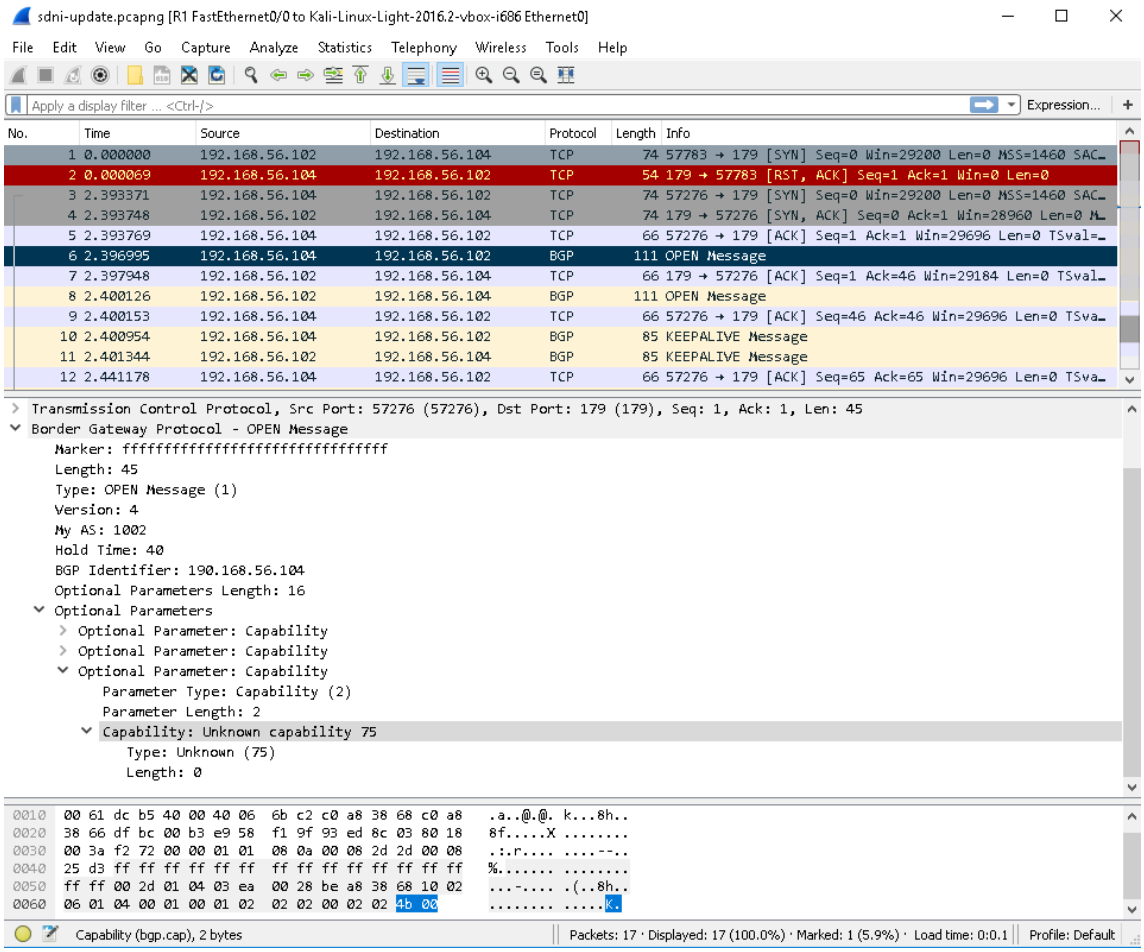
Στην προτεινόμενη αρχιτεκτονική το uri που στέλνεται στο αυτόνομο σύστημα από το οποίο ζητείται βοήθεια, δείχνει σε μία αναφορά IODEF όπως αυτή περιγράφηκε πρωθύστερα. Στην αναφορά αυτή περιέχονται οι διευθύνσεις IP που αφορούν το κάθε αυτόνομο σύστημα. Οι διευθύνσεις IP που σχετίζονται με ένα αυτόνομο σύστημα κατά τη διάρκεια της επίθεσης είναι αρχικά οι διευθύνσεις που αφορούν το ίδιο το αυτόνομο σύστημα (δημόσιες διευθύνσεις που υπάρχουν σε αυτό) και οι διευθύνσεις IP της κίνησης που διέρχεται από αυτό με προορισμό το θύμα. Ο τρόπος με τον οποίο γίνεται αυτή η αντιστοίχιση βασίζεται στα δεδομένα που συλλέγονται μέσω του πρωτοκόλλου NetFlow από τους δρομολογητές άκρης. Ακολουθεί ο τρόπος που υλοποιήθηκε το προτεινόμενο μοντέλο επικοινωνίας-συνεργασίας



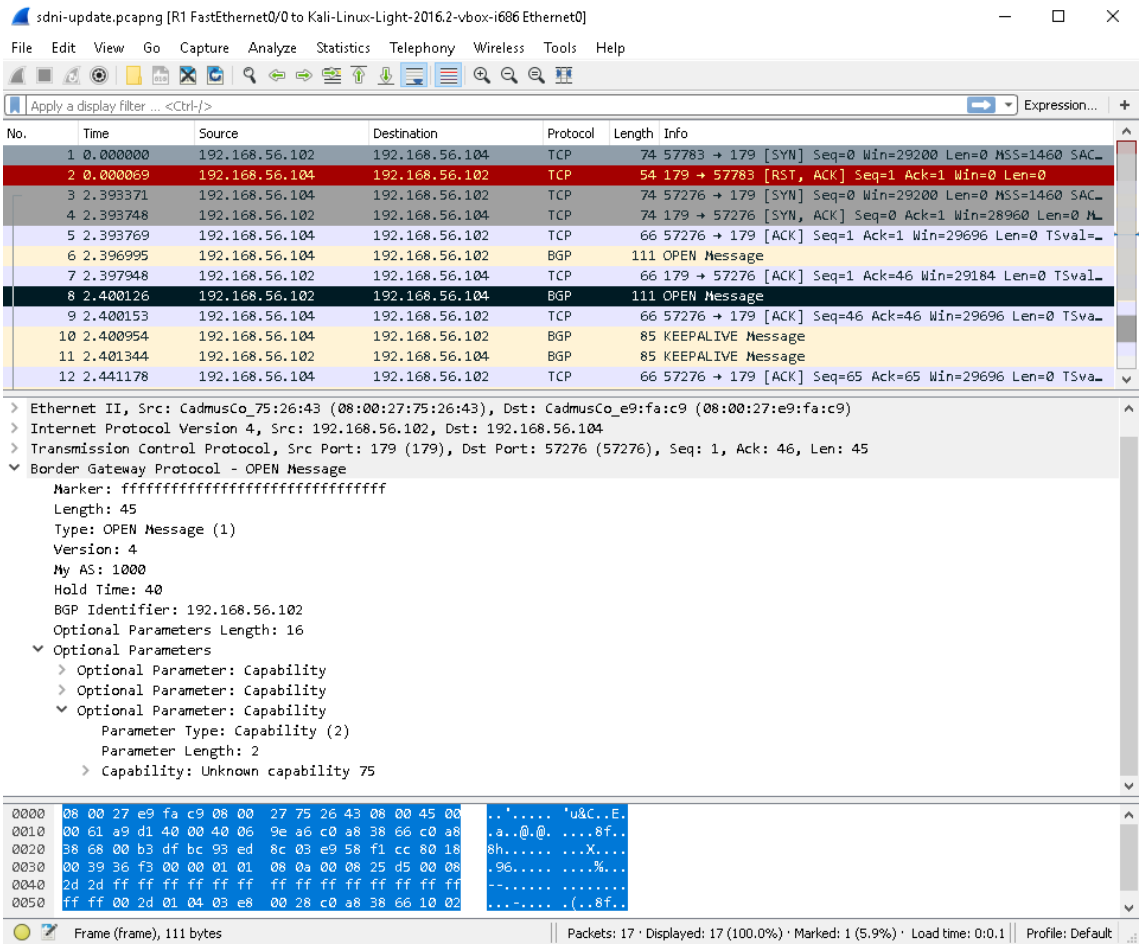
Σχήμα 13: Αίτημα για βοήθεια και απόκτηση της IODEF αναφοράς

3.6 Επέκταση του ελεγκτή Ryu για υποστήριξη του προτεινόμενου μοντέλου επικοινωνίας

Βάσει του προτεινόμενου μοντέλου επικοινωνίας μεταξύ των SDN τομέων διαφορετικών αυτόνομων συστημάτων, που περιγράφηκε παραπάνω επεκτάθηκε ο ελεγκτής Ryu, ώστε να υποστηρίζεται η συγκεκριμένη προσέγγιση. Αρχικά προστέθηκε στην εφαρμογή του ελεγκτή το capability που επιτρέπει τη διέλευση μηνυμάτων, μεταξύ BGP ομότιμων, που μεταφέρουν στο πεδίο NLRI του μηνύματος BGP Update ένα uri μήνυμα. Συγκεκριμένα επεκτάθηκε ο BGP Speaker του Ryu ώστε να υποστηρίζει τέτοια μηνύματα. Ακόμη, σεβόμενοι τον τρόπο λειτουργίας του ελεγκτή δημιουργήσαμε μία εφαρμογή για τον χειρισμό αυτών των ιδιαίτερων μηνυμάτων. Επί της ουσίας κατασκευάσθηκε μια εφαρμογή για τον χειρισμό αυτών των μηνυμάτων ασύγχρονα σαν event. Παρακάτω ακολουθεί τόσο ο κώδικας για τον χειρισμό τέτοιων events αλλά και στιγμιότυπα από το Wireshark που δείχνουν την υποστήριξη του προτεινόμενου feature.

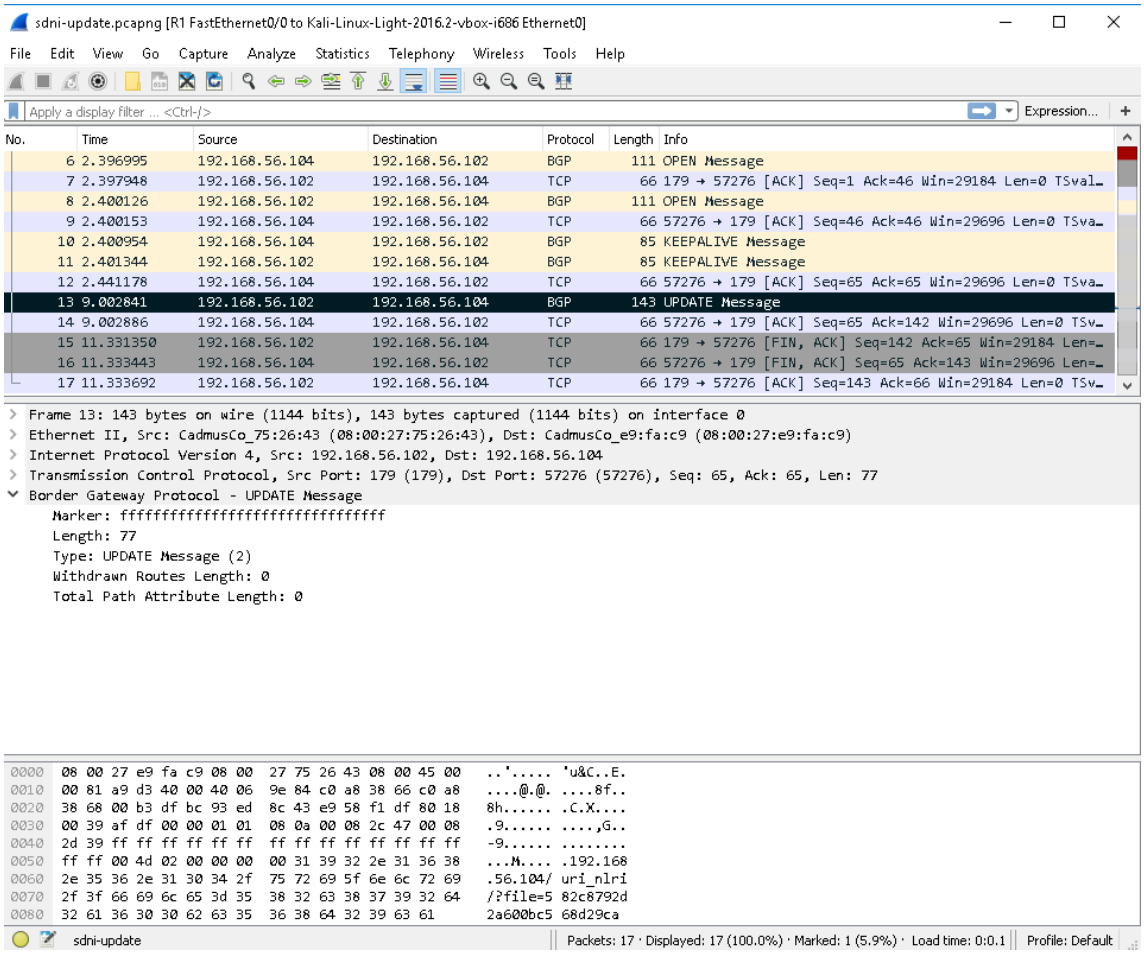


Σχήμα 14: Εκκίνηση BGP συνεδρίας μεταξύ των ελεγκτών με ενεργό το συγκεκριμένο capability



Σχήμα 15: Αποδοχή BGP συνεδρίας μεταξύ των ελεγκτών με ενεργό το συγκεκριμένο capability

Στο παράρτημα δίνονται τα αρχεία κώδικα που επεκτάθηκαν για την υποστήριξη τόσο του BGP Capability αλλά και την μεταφορά του uri μέσα από μήνυμα BGP Update.



Σχήμα 16: BGP Update μήνυμα-αίτηση για βοήθεια με uri που δείχνει σε IODEF αναφορά

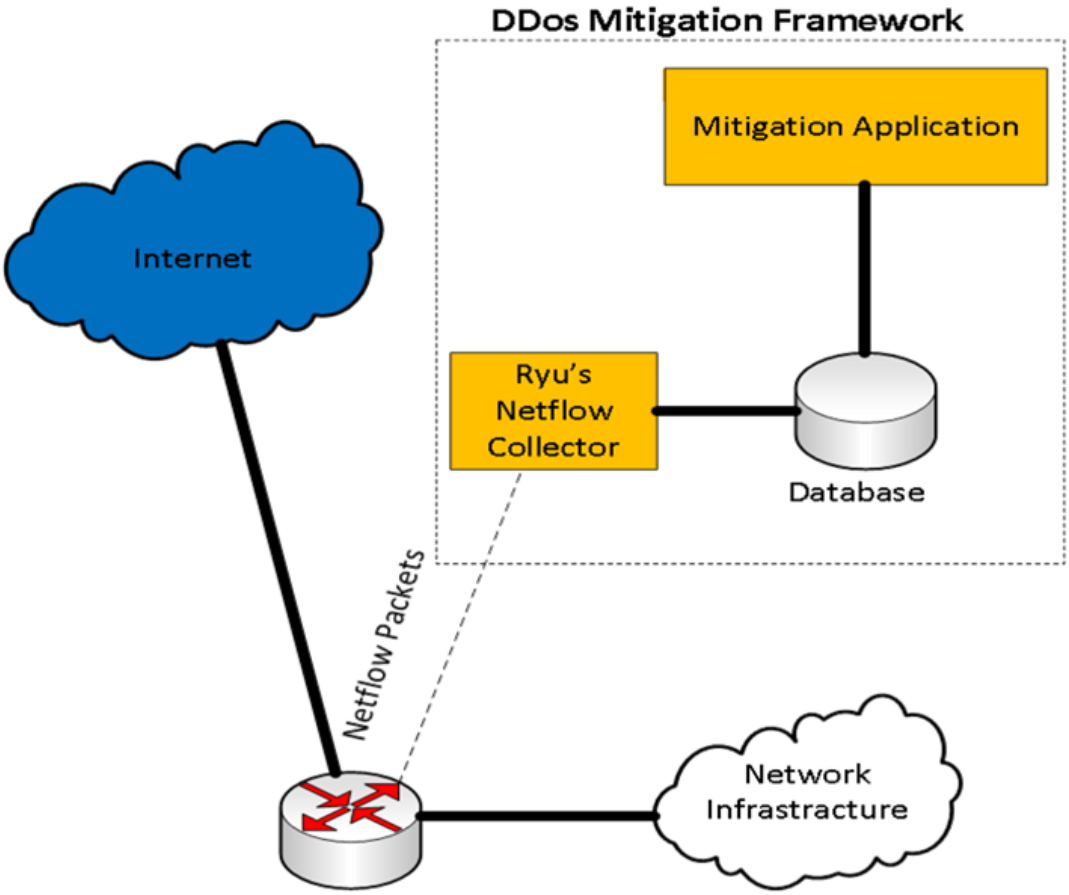
Παρακάτω παρατίθεται ο κώδικας για τη δημιουργία ενός custom Event, το οποίο δημιουργείται όταν σταλεί μήνυμα BGP Update με τα αντίστοιχα πεδία μεγέθους μηδέν και στο πεδίο NLRI, να υπάρχει uri που δείχνει σε IODEF αναφορά. Τόσο για να χειρισθεί κάποια εφαρμογή αλλά και να ”ακούει” τέτοια Events, πρέπει αρχικά να κάνει register σε αυτά και να έχει και μία συνάρτηση χειρισμού τέτοιων Events.

```
import socket
from ryu.base import app_manager
from ryu.controller import event
from ryu.lib import hub

SDNI_EV_DISPATCHER='sdni'
BUFSIZE = 65535

class EventSdni(event.EventBase):
    def __init__(self, msg):
        super(EventSdni, self).__init__()
        self.msg = msg
class SdniCollector(app_manager.RyuApp):
    def __init__(self):
        super(SdniCollector, self).__init__()
        self.name = 'sdni_collector'
        self._start_recv()

    def start(self):
        return self.thread

    def _recv_loop(self):
        self.sock.listen(5)
        while True:
            self.sock.setblocking(True)
            (clientsocket, address) = self.sock.accept()
            data=clientsocket.recv(1024)
            #print data
            self.send_event_to_observers(EventSdni(data))

    def _start_recv(self):
        self.sock =socket.socket(socket.AF_INET, socket.SOCK_STREAM)

        self.sock.bind(('0.0.0.0',
                        9000))
        self.thread = hub.spawn_after(1, self._recv_loop)
```

Σχήμα 17: Custom Event στον Ryu για χειρισμό μηνυμάτων BGP Update που μεταφέρουν uri

3.7 Επέκταση του ελεγκτή Ryu για συλλογή δεδομένων κίνησης από το δίκτυο

Στα πλαίσια της παρούσας διπλωματικής εργασίας επεκτάθηκε ο ελεγκτής Ryu ώστε να δέχεται μηνύματα Netflow v5 και να τα αποθηκεύει. Συγκεκριμένα κατασκευάστηκε event-based προσέγγιση για συλλογή, επεξεργασία και αξιοποίηση των Netflow πακέτων. Παρακάτω παρουσιάζεται ο προτεινόμενος τρόπος αποστολής και λήψης των πακέτων:



Σχήμα 18: Συλλέκτης Netflow μηνυμάτων στον ελεγκτή Ryu

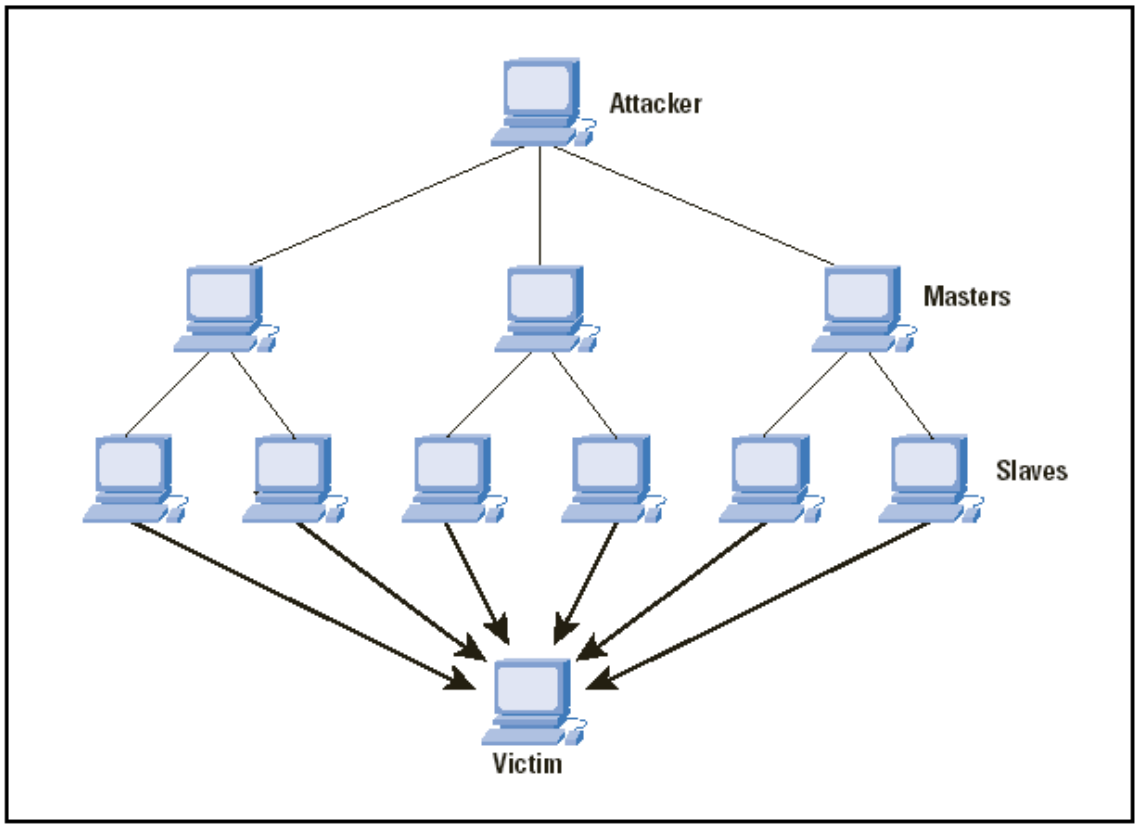
4 Τρόποι Αντιμετώπισης και Πρόληψης DDoS επιθέσεων

4.1 Γενικά Στοιχεία

Όπως έχει ήδη αναφερθεί, μπορεί να θεωρηθεί ότι μια επίθεση είναι τύπου DDoS, όταν ένας μεγάλος αριθμός μηχανημάτων έχει προσβληθεί από κακόβουλο κώδικα και προσπαθεί ταυτόχρονα και συντονισμένα, μετά των οδηγιών του επιτιθέμενου να εξουθενώσουν τους πόρους του θύματος και να το αναγκάσουν να αρνηθεί τις υπηρεσίες του στους νόμιμους χρήστες. Σύμφωνα με το [6] μπορούμε να χωρίσουμε τις επιθέσεις άρνησης υπηρεσίας σε δύο μεγάλες κατηγορίες στις τυπικές DDoS επιθέσεις και στις Distributed Reflector Denial of Service (DRDoS) επιθέσεις.

4.1.1 DDoS επιθέσεις

Σε μία απλή επίθεση άρνησης παροχής υπηρεσίας ο επιτιθέμενος χρησιμοποιεί ένα δίκτυο από προσβεβλημένους υπολογιστές οι οποίοι διακρίνονται σε αφέντες (master) και σκλάβους (slave). Ο επιτιθέμενος καθοδηγεί και προστάζει στους αφέντες και εκείνοι με τη σειρά τους καθοδηγούν τους σκλάβους να εκκινηθεί η επίθεση προς το επιθυμητό θύμα όπως φαίνεται στην παρακάτω εικόνα:



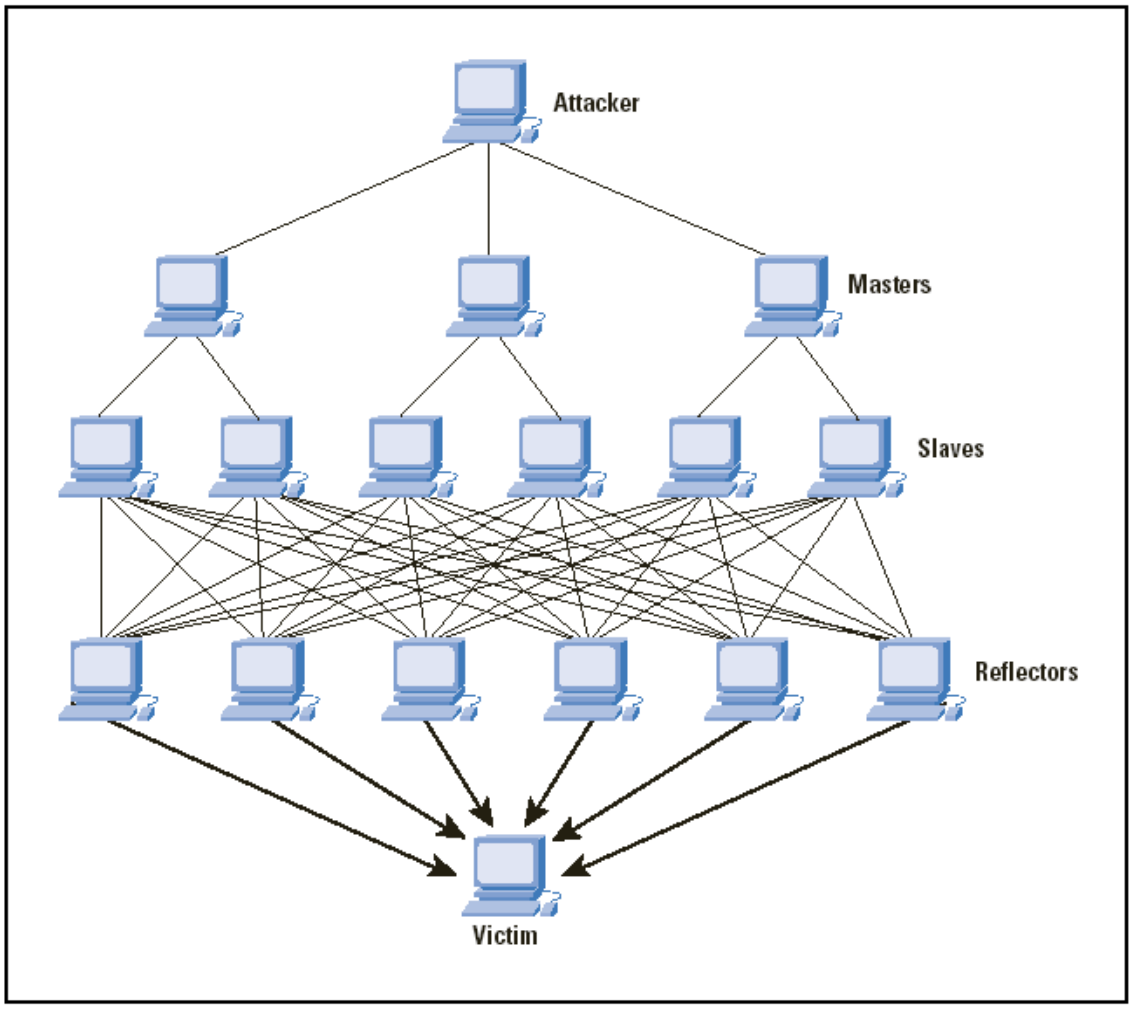
Σχήμα 19: DDoS επίθεση

Είναι αρκετά σύνηθες σε τέτοιου είδους επιθέσεις τα πακέτα που στέλνονται στο θύμα να έχουν διαφορετικές διευθύνσεις από τις πραγματικές διευθύνσεις (spoofed). Αυτή η τεχνική προτιμάται για δύο λόγους. Από τη μία ο επιτιθέμενος θέλει να αποκρύψει τις πραγματικές διευθύνσεις των σκλάβων για να μην μπορεί να ανιχνευθεί η πηγή της επίθεσης και από την άλλη να μπορεί να εναλλάσσει δυναμικά τις διευθύνσεις των επιτιθέμενων ώστε να μην μπορούν να εφαρμοσθούν στατικά φίλτρα για την αντιμετώπιση των επιθέσεων.

4.1.2 DRDoS επιθέσεις

Εν αντιθέσει με τις απλές DDoS επιθέσεις στις DRDoS το δίκτυο που χρησιμοποιείται για την επίθεση εκτός από αφέντες και σκλάβους υπολογιστές έχει και υπολογιστές που λειτουργούν ως κάτοπτρα (reflectors)[7]. Η διαδικασία επίθεσης που ακολουθείται μέχρι ενός σημείου είναι ίδια με την παραπάνω, ωστόσο τα πακέτα που στέλνουν οι σκλάβοι έχουν διεύθυνση πηγής τη διεύθυνση του θύματος και διεύθυνση προορισμού τους reflectors, προτρέποντας με αυτόν τον τρόπο να συνδεθούν με το θύμα. Εν συνεχεία οι reflectors απαντούν στα ειλημμένα πακέτα θεωρώντας πως το θύμα είναι αυτό που έστειλε όλα αυτά τα αιτήματα, με αποτέλεσμα να στέλνεται μεγάλος όγκος κίνησης ως απάντηση στο θύμα. Είναι προφανές ότι οι reflectors είναι ένα

μη προσβεβλημένο δίκτυο που άθελα του συμμετέχει στην διεξαγωγή μιας επίθεσης άρνησης παροχής υπηρεσίας όπως φαίνεται παρακάτω:



Σχήμα 20: DDoS επίθεση με ανάκλαση

Παρότι βάσει της φύσης του πρωτοκόλλου UDP στηρίζονται πολλές επιθέσεις σε αυτό και στον τρόπο λειτουργίας του θα παραθέσουμε και μια σειρά επιθέσεων που ανάγονται στο πρωτόκολλο TCP και εκμεταλλεύονται αδυναμίες του πρωτοκόλλου. Παρακάτω αρχικά ακολουθούν επιθέσεις ανάκλασης που γίνονται μέσω του πρωτοκόλλου UDP καθώς και ο παράγοντας ενίσχυσης των επιθέσεων δηλαδή ο λόγος του μεγέθους της απάντησης προς το μέγεθος της ερώτησης. Έπειτα ακολουθεί ένας αντίστοιχος πίνακας για το πρωτόκολλο TCP.

Πρωτόκολλο	Παράγοντας ενίσχυσης
DNS	28 εως 54
NTP	556,9
SNMPv2	6,3
NetBIOS	3,8
SSDP	30,8
QOTD	140,3
BitTorrent	3.8
Kad	16,3
Quake Network Protocol	63.9
Steam	5.5
Multicast DNS	2 εως 10
RIPv1	131,24
PortMap	7 εως 28
LLDAP	46 εως 55
CharGEN	358,8

Πίνακας 1: UDP Amplification Επιθέσεις

Πρωτόκολλο	Παράγοντας ενίσχυσης
FTP	51
HTTP	409
IMAP	75.951
IPP	36.978
IRC	92.669
MYSQL	84.042
NetBIOS	11.760
NNTP	83.449
POP3	98.530
SIP	8.174
SMTP	56.234
SSH	1.713
Telnet	61

Πίνακας 2: TCP Amplification Επιθέσεις [8]

Παρότι παραπάνω φαίνεται ότι το πρωτόκολλο TCP δίνει υψηλούς πολλαπλασιαστικούς παράγοντες, παίζει σημαντικό ρόλο ο αριθμός των συσκευών που θα λειτουργήσουν ως reflectors καθώς το συνολικό μέγεθος της επίθεσης προκύπτει πολλαπλασιάζοντας τον παράγοντα ενίσχυσης με τον αριθμό των reflectors. Στο συνολικό χώρο των IPv4 διευθύνσεων είναι πολύ μεγαλύτερος ο αριθμός των ενισχυτών-reflectors που είναι ευάλωτοι σε εκμετάλλευση του UDP πρωτοκόλλου έναντι του TCP.

Συγκρίνοντας τα δύο είδη μπορεί να επισημανθεί ότι οι DRDoS επιθέσεις είναι αρκετά πιο επιζήμιες αφού η επίθεση γίνεται από περισσότερα μηχανήματα και προφανώς είναι πιο κατανεμημένη. Συνακόλουθο της κατανεμημένης φύσης της επίθεσης είναι και ο μεγαλύτερος όγκος της κίνησης που παράγεται και μεγαλύτερη η ζημιά που προκαλείται.

4.1.3 Γνωστές Δικτυακές επιθέσεις

Πριν συνεχίσουμε στους τρόπους πρόληψης και αντιμετώπισης καθίσταται αναγκαίο να αναφερθούμε στις πιο γνωστές επιθέσεις [9]:

- **Apache2:** Η συγκεκριμένη επίθεση γίνεται σε έναν Apache εξυπηρετητή δικτύου όπου ο πελάτης στέλνει ένα HTTP αίτημα με πολλές HTTP επικεφαλίδες, με αποτέλεσμα ο εξυπηρετητής να δέχεται πολλά αιτήματα και μην μπορώντας να διαχειριστεί το φόρτο τελικά να πέφτει (crash).

- **ARP Poison:** Προϋπόθεση σε αυτή την επίθεση είναι ο επιτιθέμενος να έχει φυσική πρόσβαση στο τοπικό δίκτυο του θύματος. Ο επιτιθέμενος εξαπατά τους υπολογιστές ενός τοπικού δικτύου δίνοντας σε αυτούς λάθος MAC διεύθυνση για υπολογιστές που είναι γνωστή η διεύθυνση IP τους. Ο τρόπος που επιτυγχάνεται αυτό είναι η ταχύτατη απάντηση, από τον επιτιθέμενο, σε πακέτα "arp who-has" με αποτέλεσμα να δοθεί ως απάντηση διαφορετική MAC διεύθυνση από αυτήν που θα έπρεπε να δοθεί στην πραγματικότητα.

- **Back:** Η επίθεση αυτή στοχεύει τους Apache εξυπηρετητές δικτύου και συγκεκριμένα τους πλημμυρίζει με αιτήματα που το URL τους απαρτίζεται από μεγάλο αριθμό του συμβόλου / (front-slash) και αυτοί με τη σειρά τους αφιερώνουν μεγάλο χρονικό διάστημα στην επεξεργασία τέτοιων αιτημάτων και δεν εξυπηρετούν "νόμιμους" χρήστες.

- **Land:** Στις Land επιθέσεις, ο επιτιθέμενος στέλνει στο θύμα ένα παραποιημένο πακέτο TCP SYN με διεύθυνση προορισμού αλλά και διεύθυνση πηγής τη διεύθυνση IP του θύματος με αποτέλεσμα το θύμα να απαντάει στον εαυτό του και εν τέλει το σύστημα του θύματος να κλειδώνεται.

- **SYN flood:** Οι επιθέσεις πλημμύρας πακέτων TCP SYN τον τρόπο που γίνεται η τριπλή χειραψία στο πρωτόκολλο TCP. Η σύνδεση μεταξύ δύο υπολογιστών με το πρωτόκολλο TCP αποτελείται από τα ακόλουθα γεγονότα, ο πελάτης στέλνει ένα μήνυμα TCP SYN, ο εξυπηρετητής απαντάει με ένα μήνυμα TCP SYN/ACK και τοποθετεί το αίτημα σύνδεσης σε μία ουρά και τέλος ο πελάτης στέλνει ένα μήνυμα ACK όπου σηματοδοτείται η εκκίνηση της συνεδρίας. Αν ο επιτιθέμενος ανοίξει πολλές χειραψίες με τον εξυπηρετητή αλλά δεν απαντάει με ACK πακέτα στο τέλος, τότε η ουρά του εξυπηρετητή-θύματος γεμίζει με αιτήματα σύνδεσης και δεν δέχεται άλλες TCP συνδέσεις, με αποτέλεσμα να μην μπορούν να εξυπηρετηθούν καλόβουλοι χρήστες.

- **Ping of Death:** Ο επιτιθέμενος σε αυτή την επίθεση δημιουργεί πακέτα με μέγεθος μεγαλύτερο από το επιτρεπτό μέγεθος στο πρωτόκολλο IP με αποτέλεσμα τα συστήματα των δεκτών αυτών των πακέτων να μην μπορούν να ανταποκριθούν και να επανεκκινούνται.

- **Smurf Attack:** Ένα από τα πιο διαδεδομένα είδη επιθέσεων λόγω της ευκολίας υλοποίησης της είναι η επίθεση Smurf. Ο επιτιθέμενος στέλνει πακέτα ICMP τύπου "echo-request" με διεύθυνση πηγής την διεύθυνση του επιθυμητού θύματος και διεύθυνση προορισμού τη διεύθυνση broadcast πολλών υποδικτύων. Αυτό οδηγεί τα μηχανήματα των υποδικτύων να απαντήσουν στο θύμα και να το πλημμυρίσουν με "echo-reply" μηνύματα.

- **TCP Reset:** Άλλη μια επίθεση που στοχεύει το πρωτόκολλο TCP είναι η επίθεση TCP Reset, μέσω της οποίας, όταν βρεθεί ένα αίτημα για σύνδεση με το θύμα ο κακόβουλος χρήστης στέλνει ένα παραποιημένο πακέτο στο θύμα με σκοπό να τερματισθεί η σύνδεση που μόλις δημιουργήθηκε.

- **Teardrop:** Είναι σύνηθες ένα πακέτο όταν ταξιδεύει από τον αποστολέα προς το δέκτη να χωρισθεί σε μικρότερα θραύσματα (fragments) κατά τη διαδικασία

του κατακερματισμού στο πρωτόκολλο IP. Κατά την teardrop επίθεση ο επιτιθέμενος κατασκευάζει ένα ρεύμα (stream) από θραύσματα IP με πεδίο offset διαφορετικό από το αναμενόμενο με αποτέλεσμα όταν το θύμα προσπαθήσει να επανασυναρμολογήσει τα θραύσματα να προκληθεί ασυμφωνία και το σύστημα να σταματήσει να λειτουργεί και να απαιτείται επανεκκίνηση.

- **UDP Storm:** Σε μία σύνδεση πάνω από το πρωτόκολλο UDP, λειτουργεί μια υπηρεσία που ονομάζεται "chargen" (character generation) όπου κάθε φορά που λαμβάνεται ένα πακέτο UDP δημιουργεί μια σειρά απο χαρακτήρες, ενόσω μια υπηρεσία ηχούς αναπαράγει τους χαρακτήρες αυτούς. Ο επιτιθέμενος μπορεί να εκμεταλλευτεί αυτές τις υπηρεσίες στέλνοντας ένα παραποιημένο πακέτο με διεύθυνση πηγής αυτή του θύματος σε κάποιο μηχάνημα που υποστηρίζει chargen. Με αυτόν τον τρόπο ανταλλάσσεται συνεχώς stream άσκοπης κίνησης μεταξύ των μηχανημάτων αυτών επιβαρύνοντας το ίδιο το δίκτυο, αφού το μηχάνημα προορισμού θα απαντήσει και θα στείλει στο θύμα και αυτό με τη σειρά του θα ακολουθήσει την ίδια διαδικασία κ. ο. κ.

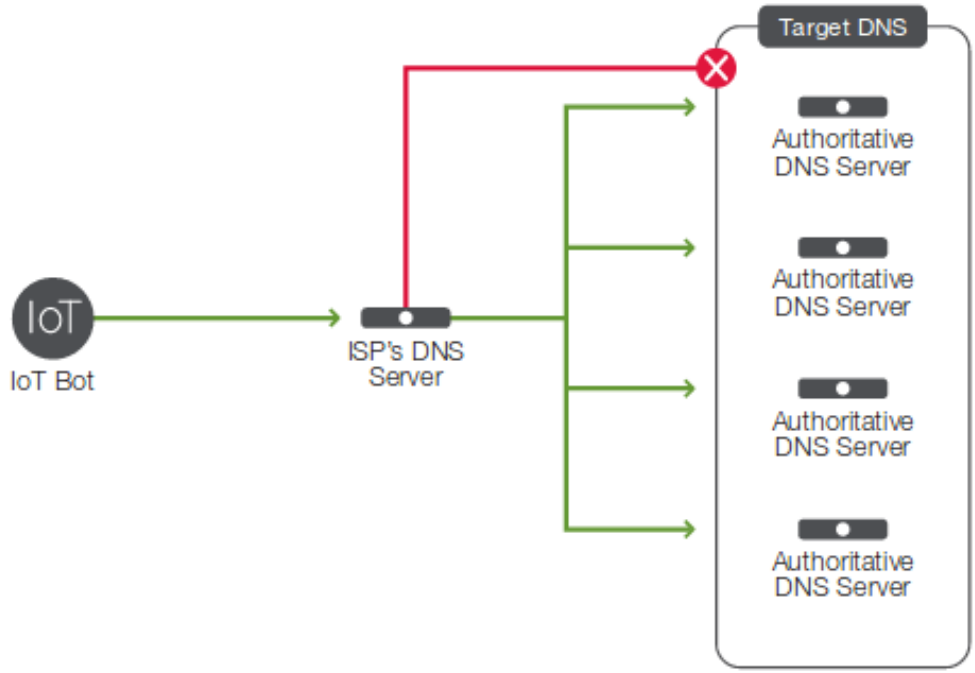
Αξίζει να σημειωθεί σε αυτό το σημείο ότι πλέον οι επιθέσεις έχουν γίνει αρκετά πιο περίπλοκες στον τρόπο με τον οποίο λειτουργούν αφού συνδυάζουν τόσο ευθείς επιθέσεις και επιθέσεις ανάκλασης δυσκολεύοντας τόσο την αντιμετώπιση αλλά και την αναγνώριση τέτοιου είδους επιθέσεων. Επίσης, οι multi vector attacks (συνδυαστικές επιθέσεις) δημιουργούν πολύ μεγάλα προβλήματα αφού έχουν ως στόχο τόσο την εξουθένωση των μηχανημάτων θυμάτων αλλά και την πλημμύρα του δικτύου από κακόβουλη κίνηση.

Το 2016 έλαβαν χώρα τρεις επιθέσεις άξιες αναφοράς λόγω του μεγέθους τους [10]:

- Επίθεση 630Gbps ενάντια στην Krebs
- Επίθεση 990 Gbps ενάντια στην γαλλική εταιρεία OVH
- Επίθεση 1, 2 Tbps ενάντια στην εταιρία παροχής DNS DYN

Οι επιθέσεις αυτές πέραν της πολυπλοκότητας τους και των πολλαπλών διαφορετικών τεχνικών που περιείχαν δημιούργησαν μεγάλα προβλήματα στο διαδίκτυο. Συγκεκριμένα χρησιμοποιήθηκαν ως botnet πολλές internet of things συσκευές λόγω της ελλιπούς ασφάλειας τους. Οι συσκευές αυτές είχαν προσβληθεί από το malware Mirai, όπου αναζητούνται συσκευές με default ονόματα χρηστών και κωδικούς δημόσια διαθέσιμες στο internet. Οι συσκευές αυτές χρησιμοποιούνται ως slaves για να στέλνουν αιτήματα που αφορούν στο πρωτόκολλο DNS, ωστόσο τα καταλλήλως διαμορφωμένα αιτήματα που στέλνουν οδηγούν τον DNS εξυπηρετητή στον οποίο απευθύνονται να μην μπορεί να τα εξυπηρετήσει και με τη σειρά του να ζητά από authoritative εξυπηρετητές τις απαντήσεις των ερωτημάτω ν που έλαβε. Με αυτό τον τρόπο ο ενδιάμεσος εξυπηρετητής γίνεται μέρος της επίθεσης πλημμυρίζοντας με ερωτήματα τους άλλους authoritative εξυπηρετητές. Αυτή η τεχνική ονομάζεται DNS water torture και είναι αρκετά διαφορετική από τις προαναφερθείσες τεχνικές. Η τεχνική αυτή

επεκτάθηκε ακόμα παραπάνω αφού παρατηρήθηκε και GRE κίνηση (τα GRE tunnel χρησιμοποιούνται κανονικά για VPN συνδέσεις) και με αυτόν τον τρόπο η κίνηση διερχόταν από τους δρομολογητές χωρίς να γίνει αισθητή, επιτρέποντας μάλιστα στα πακέτα αυτά μεγάλο payload προκαλώντας στον στόχο και overhead επεξεργασίας για την ανασυγκρότηση των πακέτων. Η επίθεση αυτή δημιούργησε τόσο πρόβλημα στους εξυπηρετητές ως προς την επεξεργαστική τους ισχύ αλλά και στο ίδιο το δίκτυο λόγω της κίνησης που δημιουργήθηκε. Ακολουθεί σχηματικά η υλοποίηση της DNS Water Torture επίθεσης:



Σχήμα 21: DNS Water Torture επίθεση

4.2 Reverse Path Forwarding

Το Unicast Reverse Path Forwarding όπως αυτό αρχικά προτάθηκε [11] είναι μια δυνατότητα (feature) που υποστηρίζεται από δρομολογητές (edge router) και μπορεί να χρησιμοποιηθεί για φιλτράρισμα της κίνησης που προέρχεται από το εσωτερικό του δίκτυο (ingress filtering). Οι στόχοι κατά την ανάπτυξή του, ήταν η δημιουργία μιας δυνατότητας που θα μπορούσε να αυτοματοποιήσει τη διαδικασία ελέγχου εισερχόμενης κίνησης σε κάποια διεπαφή του δρομολογητή, η κλιμακωσιμότητα καθώς ο αριθμός των διευθύνσεων IP αυξάνεται και η άμεση απόκριση . Μπορούμε να συνοψίσουμε την λειτουργία του RPF στο εξής:Για να προωθήσουμε ένα πακέτο που φθάνει στο interface του δρομολογητή, που είναι το RPF ενεργοποιημένο, πρέπει να γνωρίζουμε έναν τρόπο να στείλουμε στην διεύθυνση πηγής από την οποία μας ήρθε

το πακέτο αυτό αλλιώς απορρίπτεται. Δηλαδή, αν θέλουμε να προωθήσουμε ένα πακέτο στην διεύθυνση αυτή πρέπει να υπάρχει γνωστή διαδρομή στον δρομολογητή προς αυτή την διεύθυνση, μηχανισμός που επιτρέπει την αποφυγή προώθησης πακέτων με παραποιημένες διευθύνσεις IP. Το RPF μπορεί να ενεργοποιηθεί ανά διεπαφή του δρομολογητή και υπάρχουν τρεις διαφορετικές λειτουργίες μέσω των οποίων διακρίνεται ο προαναφερθείς τρόπος :

- **Strict mode**
- **Loose mode**
- **Virtual routing and forwarding mode**

Πριν όμως προχωρήσουμε στις διαφορές των παραπάνω λειτουργιών θεωρείται αναγκαίο να περιγραφεί επιγραμματικά ο τρόπος με τον οποίο γίνεται η δρομολόγηση στο διαδίκτυο. Η δρομολόγηση χωρίζεται σε δύο βασικές κατηγορίες στη συμμετρική δρομολόγηση και την ασύμμετρη. Το σύνηθες μοντέλο είναι να υπάρχει συμμετρική δρομολόγηση μεταξύ του δικτύου πελάτη-παρόχου και ασύμμετρη δρομολόγηση μεταξύ των δικτύων των παρόχων . Αυτό δημιουργεί την εξής κατάσταση στους δρομολογητές, αν πρόκειται για δίκτυο πελάτη-παρόχου (customer-ISP) υπάρχουν κατά κύριο λόγο συμμετρικές ροές και αναμένεται η επικοινωνία με κάποια διεύθυνση IP να γίνει μέσω του ίδιου interface. Από την άλλη αν πρόκειται για δρομολογητή μεταξύ παρόχων τότε δεν είναι απαραίτητο ότι η επικοινωνία με μία διεύθυνση IP θα γίνει καθολικά από το ίδιο interface, αφού υπάρχει περίπτωση να στείλουμε ένα πακέτο από κάποιο interface του δρομολογητή και να λάβουμε την αντίστοιχη απάντηση από άλλο. Ο λόγος που συμβαίνει αυτό βασίζεται στην επιλογή καλύτερης διαδρομής του πρωτοκόλλου BGP, μέσω του οποίου παρότι υπάρχουν όλες οι εφικτές διαδρομές στον πίνακα δρομολόγησης, μόνο η καλύτερη διαδρομή επιλέγεται και εισάγεται στον πίνακα προώθησης του δρομολογητή. Στη συνέχεια θα αναλύσουμε τις τρεις διαφορετικές λειτουργίες του RPF.

4.2.1 Strict mode

Η ενεργοποίηση του strict mode στο interface ενός δρομολογητή είναι κατάλληλη στην περίπτωση που χρησιμοποιείται το ίδιο interface για να απαντηθούν πακέτα που έφθασαν μέσω αυτού. Αυτό όμως όπως αναφέραμε παραπάνω αποτελεί την αρχή της συμμετρικής δρομολόγησης και για αυτόν τον λόγο σε σημεία του δικτύου που την αναμένουμε μπορεί να ενεργοποιηθεί η συγκεκριμένη λειτουργία. Με την ενεργοποίηση της εξασφαλίζεται ότι αν φθάσει πακέτο από το εσωτερικό του δικτύου με διεύθυνση διαφορετική (spoofed συνήθως) από τις διευθύνσεις που σχετίζονται με το συγκεκριμένο interface και κατ' επέκταση με το ίδιο το δίκτυο, το πακέτο απορρίπτεται. Η τεχνική αυτή επιτρέπει στους διαχειριστές δικτύων να εξασφαλίσουν ότι συσκευές από το δίκτυο τους δεν θα στέλνουν πακέτα στο υπόλοιπο διαδίκτυο με

κρυμμένη την διεύθυνση πηγής και κατά συνέπεια να συμμετέχουν στην εξαπόλυση μιας DDoS επίθεσης και δη DRDoS.

4.2.2 Loose mode

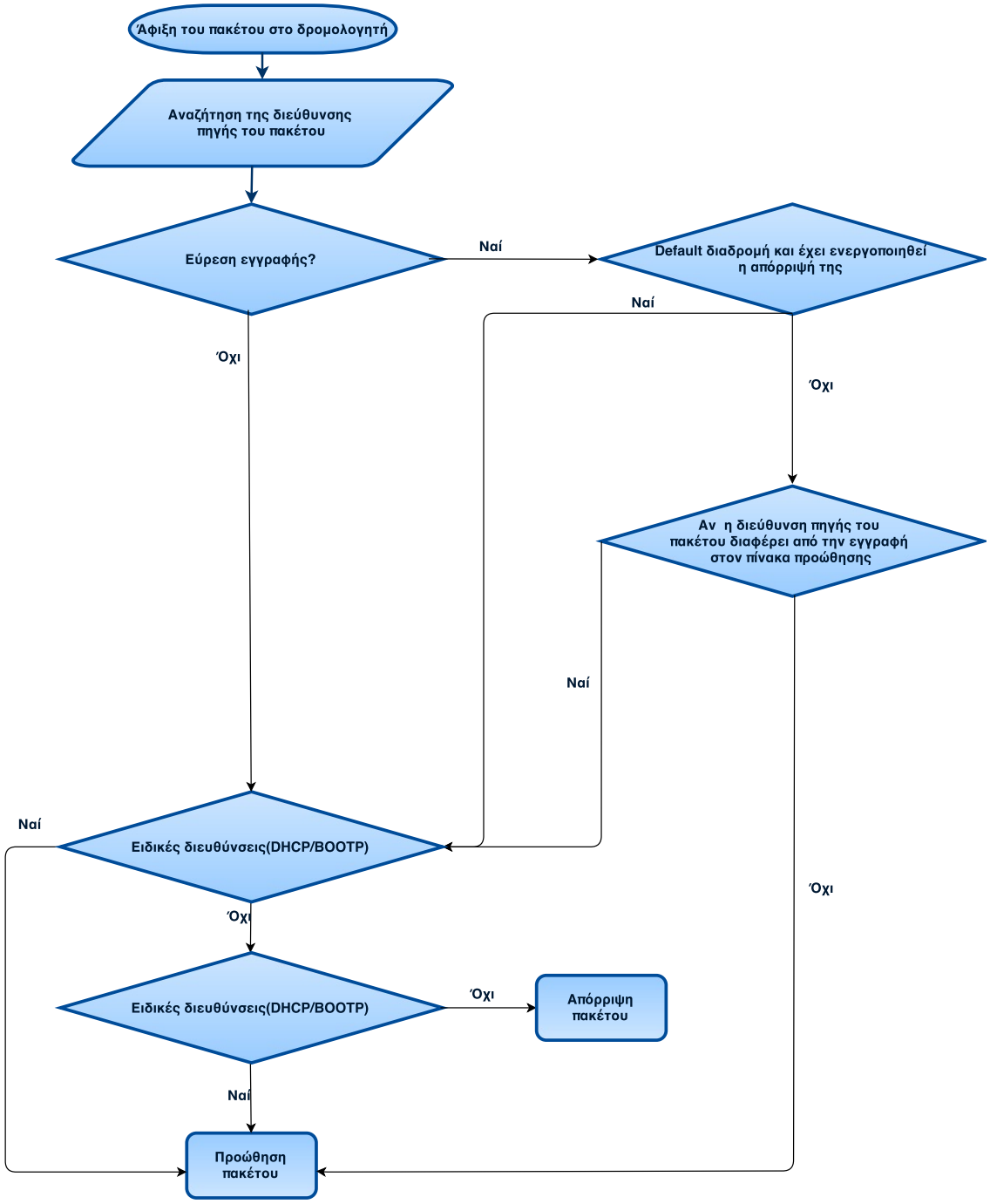
Ωστόσο, όπως αναφέρθηκε και παραπάνω το διαδίκτυο δεν απαρτίζεται μόνο από συμμετρικές ροές πακέτων και κατά συνέπεια η ύπαρξη μόνο του Strict mode δεν επαρκούσε αφού όπου υπήρχε ασύμμετρη δρομολόγηση δεν μπορούσε να εφαρμοσθεί, καθώς θα απορρίπτονταν κανονική κίνηση. Ως εκ τούτου χρειάσθηκε η ανάπτυξη μιας λειτουργίας που θα ανταποκρίνεται στο μοντέλο της ασύμμετρης δρομολόγησης. Κινούμενοι προς αυτή την κατεύθυνση αναπτύχθηκε ο loose έλεγχος (loose mode) [12] μέσω του οποίου αρκεί να υπάρχει διαδρομή από οποιοδήποτε interface προς τη διεύθυνση του πακέτου που φθάνει στο interface του δρομολογητή. Αυτό επιτρέπει στο RPF έλεγχο να εντοπίζει και να απορρίπτει αυτόματα πακέτα που φθάνουν από συγκεκριμένες διευθύνσεις:

- Διευθύνσεις που σχετίζονται με ιδιωτικά δίκτυα [13].
- Διευθύνσεις IPv4 ειδικής χρήσης [14].
- Διευθύνσεις που δεν έχουν αποδοθεί από τον RIR (Regional Internet Registry).
- Διευθύνσεις που οδηγούν σε μηδενικό inteface (null interface) στον δρομολογητή.

Οποιοδήποτε πακέτο από τις προαναφερθείσες διευθύνσεις δεν θα έπρεπε να διασχίζει το Internet και μέσω του RPF γίνεται αυτοματοποιημένα η αναγνώριση και η απόρριψη αυτών των πακέτων στα όρια μεταξύ των παρόχων υπηρεσιών Διαδικτύου. Με αυτόν τον τρόπο δίνεται η δυνατότητα πρόληψης από διάφορες επιθέσεις που σχετίζονται με την παραποίηση της διεύθυνσης πηγής. Σε αυτό το σημείο αξίζει να σημειωθεί ότι το RPF μπορεί να παρομοιασθεί με έναν δυναμικό τρόπο κατασκευής λιστών πρόσβασης (access-list) και ότι συνδυάζεται με λίστες πρόσβασης βάσει των αναγκών του εκάστοτε δικτύου. Παρακάτω ακολουθούν οι περιπτώσεις που χρησιμοποιείται αναλόγως ο loose ή ο strict έλεγχος και εν συνεχεία παρατίθεται το διάγραμμα ροής του loose ελέγχου:

Deployment Situation	Type of uRPF to Use
Leased Line Customer	Strict check
Multihomed Leased Line Customer (Same ISP)	Strict check or loose check
Multihomed Leased Line Customer (Different ISPs)	Strict check or loose check
Dialup Customers	Strict check
DSL Customers	Strict check
Cable Modem Customers	Strict check
IXP Connection—No Private Peering	Strict check
IXP Connection with Private Peering	Loose check
Private Peering—Dedicated Router	Strict check

Σχήμα 22: Περιπτώσεις χρήσης RPF



Σχήμα 23: Έλεγχος ροής RPF

4.2.3 VRF mode

Η VRF (Virtual routing and forwarding) είναι μια τεχνολογία που επιτρέπει την ταυτόχρονη συνύπαρξη πολλών στιγμιοτύπων του πίνακα δρομολόγησης στον ίδιο δρομολογητή. Είναι ακόμη σε στάδιο ανάπτυξης, ωστόσο υπάρχουν δρομολογητές που υποστηρίζουν τη λειτουργία RPF-vrf. Εν προκειμένω κάθε eBGP συνεδρία θα ανα-

κοινώνει τις διαδρομές της σε έναν συγκεκριμένο πίνακα VRF. Αυτό θα επιτρέψει στο RPF να αναζητά τις διαδρομές σε αυτόν τον πίνακα, ο οποίος θα περιέχει όλες τις διαδρομές οι οποίες σχετίζονται με αυτήν την eBGP συνεδρία. Ως εκ τούτου καθίσταται δυνατή η επαλήθευση της διεύθυνσης πηγής πακέτων η οποία αντιστοιχίζεται με τις ανακοινωθείσες διαδρομές μέσω του πρωτοκόλλου BGP που βρίσκονται αποθηκευμένες στον πίνακα VRF. Παρόλο που η παρούσα λειτουργία είναι ιδιαιτέρως χρήσιμη δεν είναι ευρέως διαδεδομένη και βρίσκεται ακόμη σε πρώιμο στάδιο.

4.2.4 Προσδοκίες επίδοσης

Η ανάπτυξη του RPF έγινε βασισμένη στον τρόπο λήψης αποφάσεων του ίδιου του δρομολογητή για αυτό και αναμένεται πολύ μικρή επίδραση στην επίδοση του, με την ενεργοποίηση του συγκεκριμένου χαρακτηριστικού. Ο λόγος που θεωρείται αυτό, είναι ότι το RPF χρησιμοποιεί ίδιου τύπου αναζήτηση της διεύθυνσης πηγής του πακέτου με την αναζήτηση που κάνει ο δρομολογητής για την διαδικασία προώθησης. Παρόλα αυτά επειδή η αρχιτεκτονική των ειδικού σκοπού ολοκληρωμένων κυκλωμάτων (ASIC) ποικίλει ανά μοντέλο δρομολογητή σίγουρα υπάρχουν διακυμάνσεις στην επίδραση του RPF στον εκάστοτε δρομολογητή. Σύμφωνα με το [12] η επίδραση στην μονάδα επεξεργασίας του δρομολογητή όταν αυτός βρίσκεται υπό επίθεση είναι η εξής:

Attack Stream	Interface	Packet Size	CPU of LC	Packets per Second
0 Mbps	Gigabit 4/0	64 bytes	0%	0
30 Mbps	Gigabit 4/0	64 bytes	52%	61,000
50 Mbps	Gigabit 4/0	64 bytes	57%	100,000
100 Mbps	Gigabit 4/0	64 bytes	88%	210,000
160 Mbps	Gigabit 4/0	64 bytes	98%	330,000

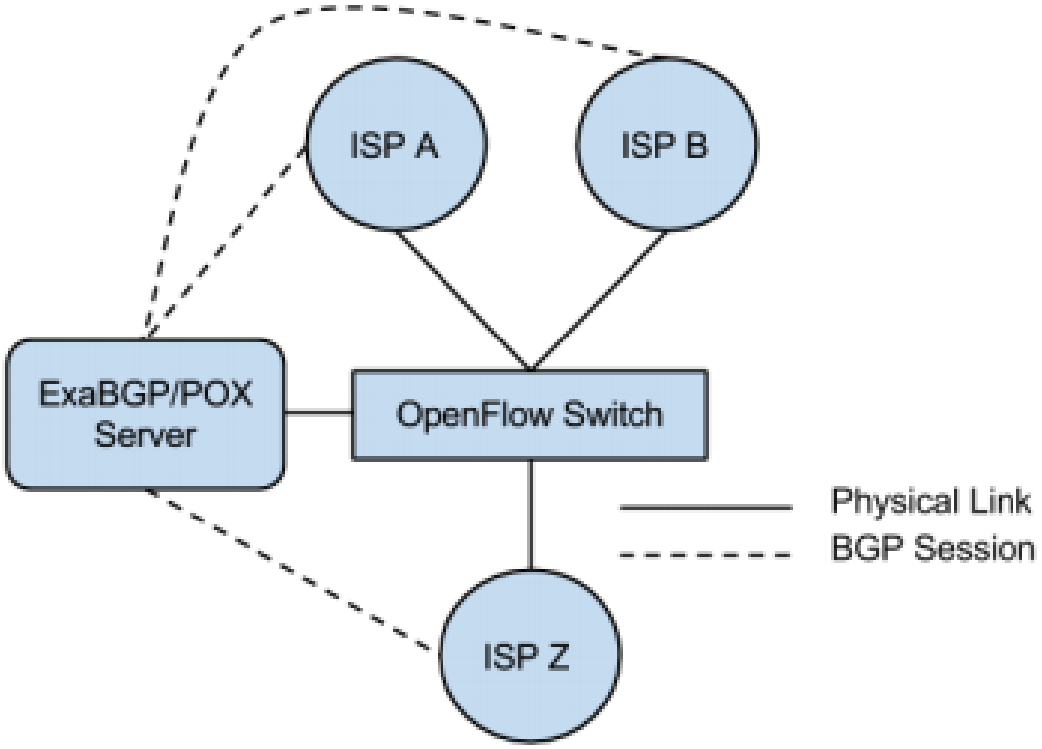
Σχήμα 24: Επίδοση χωρίς RPF

Attack Stream	Interface	Packet Size	CPU of LC	Packets per Second
0 Mbps	Gigabit 4/0	64 bytes	3%	0
30 Mbps	Gigabit 4/0	64 bytes	55%	61,000
50 Mbps	Gigabit 4/0	64 bytes	60%	100,000
100 Mbps	Gigabit 4/0	64 bytes	90%	210,000
160 Mbps	Gigabit 4/0	64 bytes	100%	330,000

Σχήμα 25: Επίδοση με RPF

4.2.5 SDN προσέγγιση

Στο [15] έγινε προσπάθεια επέκτασης της λειτουργίας RPF σε SDN δίκτυα. Συγκεκριμένα προτείνεται μια υλοποίηση για Tier 2 παρόχους χρησιμοποιώντας έναν Openflow μεταγωγέα και έναν BGP Speaker. Βάσει αυτής της υλοποίησης οι διαδρομές που ανακοινώνονται στον BGP Speaker "μεταφράζονται" σε κανόνες στον μεταγωγέα με ευφυή τρόπο, προσομοιώνοντας τη λογική του rpf που περιγράφηκε παραπάνω, εκμεταλλευόμενη ουσιαστικά την RIB και όχι την FIB, αφού λαμβάνει υπόψιν όλες τις ανακοινωθείσες διαδρομές. Με αυτόν τον τρόπο ο έλεγχος RPF μεταβαίνει στους μεταγωγείς και δεν επηρεάζεται η απόδοση του δρομολογητή.



Σχήμα 26: RPF σε SDN

4.3 Τρόποι αντιμετώπισης

Παρακάτω θα αναλυθούν οι τρόποι μέσω των οποίων μπορούν να αντιμετωπισθεί η κακόβουλη κίνηση σε ένα δίκτυο, αρχικά σε επίπεδο δρομολογητή και στη συνέχεια σε επίπεδο OpenFlow μεταγωγέα.

4.3.1 Λίστες Ελέγχου Πρόσβασης (ACL)

Οι δικτυακές λίστες πρόσβασης αποτελούν ένα σύνολο κανόνων που υπαγορεύουν ποίοι υπολογιστές του δικτύου θα έχουν πρόσβαση σε μια συγκεκριμένη υπηρεσία. Λίστες πρόσβασης υποστηρίζονται από δρομολογητές του δικτύου καθώς και από διάφορους εξυπηρετητές. Οι λίστες αυτές μπορούν να χρησιμοποιηθούν για να ρυθμισθεί το φιλτράρισμα τόσο των εισερχόμενων όσο και και των εξερχόμενων πακέτων παίζοντας κατά κάποιο τρόπο παρόμοιο ρόλο με αυτόν των firewalls. Οι λίστες πρόσβασης παρέχουν μια σειρά από ευέλικτες επιλογές όσον αφορά στο φιλτράρισμα, δηλαδή κανόνες που σχετίζονται με:

- πρωτόκολλα επιπέδου 4
- TCP και UDP πόρτες
- είδη και κωδικούς για ICMP πακέτα
- τύπους IGMP
- Εγκαθιδρυμένες TCP συνδέσεις
- TCP πακέτα με ενεργά τα bits για σημαίες ACK, FIN, PSH, RST, SYN ή URG

Γνωρίζοντας τις διευθύνσεις των συστημάτων που επιτελούν την επίθεση μπορούμε με τον ορισμό τέτοιων λιστών να απορρίψουμε κίνηση με συγκεκριμένα χαρακτηριστικά από αυτές τις πηγές. Ωστόσο η λύση αυτή απαιτεί πολύ χρόνο για την αντιμετώπιση μιας DDoS επίθεσης καθώς απαιτείται ο στατικός ορισμός των λιστών αυτών και επίσης δεν αποτελεί μια κλιμακώσιμη λύση.

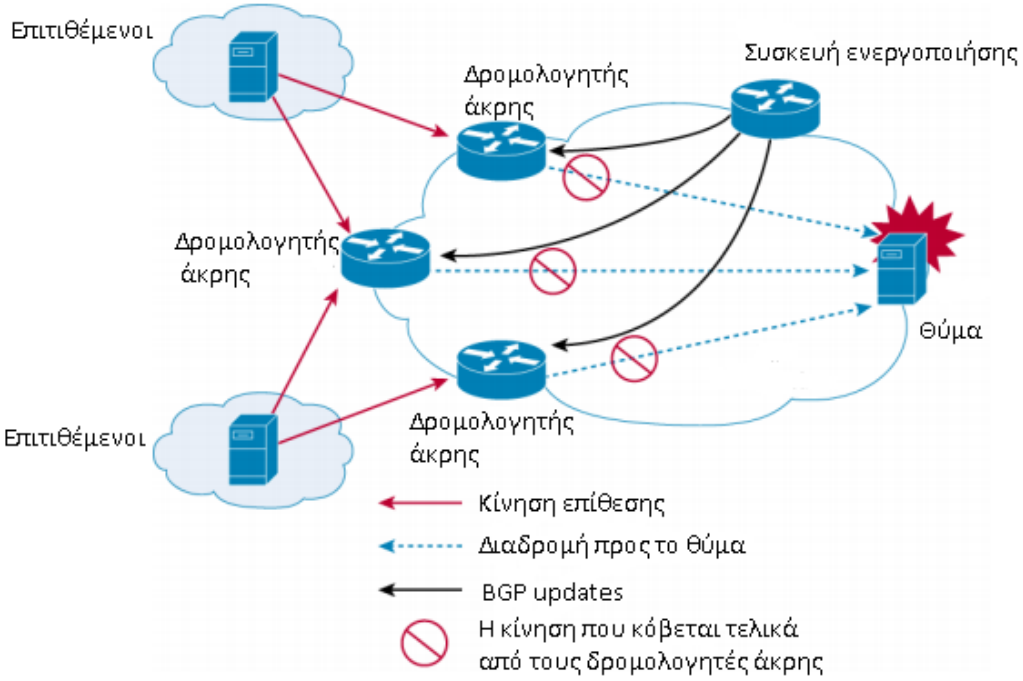
4.3.2 Απομακρυσμένη ενεργοποίηση μαύρης τρύπας (RTBH)

Στον τομέα της ασφάλειας δικτύου, οι μαύρες τρύπες αποτελούν ένα μέρος του δικτύου στο οποίο προωθείται η ανεπιθύμητη κίνηση όπου και απορρίπτεται. Η τεχνική Remotely Triggered Black Hole (RTBH) είναι ένας τρόπος να απορρίψουμε την κακόβουλη κίνηση πριν αυτή εισέλθει στο δίκτυο μας. Αφότου εντοπισθεί η επίθεση τότε με τη χρήση της συγκεκριμένης τεχνικής μπορεί να απορριφθεί η κίνηση στην άκρη του δικτύου και ειδικότερα στον δρομολογητή άκρης. Για την επίτευξη αυτού χρησιμοποιείται και χειρίζεται καταλλήλως το πρωτόκολλο δρομολόγησης BGP με σκοπό, πακέτα είτε από συγκεκριμένες διευθύνσεις πηγής είτε με συγκεκριμένες διευθύνσεις προορισμού να καταλήγουν στην επιθυμητή μαύρη τρύπα. Σύμφωνα με το [16] το φιλτράρισμα της κίνησης με την παρούσα τεχνική κυριαρχεί της τεχνικής των ACL's, αφού βασίζεται στην επίδοση των δρομολογητών και ειδικά στην επίδοση τους όσον αφορά στη δρομολόγηση των πακέτων.

4.3.3 RTBH βάσει διεύθυνσης προορισμού

Παρότι δεν είναι τόσο προφανές, κατά τη διάρκεια μιας επίθεσης DDoS δεν επηρεάζεται μόνο το θύμα αλλά υπάρχει και πιθανότητα παράπλευρης ζημιάς στο υπόλοιπο δίκτυο αφού καταναλώνεται μεγάλο ποσοστό του bandwidth των γραμμών του δικτύου. Επίσης όπως έχει ήδη προαναφερθεί υπάρχουν επιθέσεις που αυξάνουν τη χρήση των υπολογιστικών πόρων του θύματος δημιουργώντας πρόβλημα στην ίδια συσκευή. Ακόμη υπάρχει πιθανότητα απώλειας διάφορων υπηρεσιών στο δίκτυο μας οι οποίες δεν σχετίζονται άμεσα με την επίθεση, δηλαδή δεν αποτελούν στόχο της. Μια τεχνική που προκύπτει για την αποφυγή των παραπάνω προβλημάτων που δημιουργούνται κατά τη διάρκεια μιας επίθεσης είναι η RTBH βάσει διεύθυνσης προορισμού. Σύμφωνα με την τεχνική αυτή αντί να αποστέλλεται η κίνηση στο θύμα, την ανακατευθύνουμε σε έναν προορισμό όπου και απορρίπτεται, με σκοπό να αποφευχθεί να εισέλθει κακόβουλη κίνηση στο εσωτερικό του δικτύου μας. Παρακάτω θα συνεχίσουμε στο πως μπορεί να υλοποιηθεί η τεχνική αυτή και στις απαιτήσεις που υπάρχουν σε επίπεδο συσκευών.

Αρχικά απαιτείται η ύπαρξη μιας συσκευής η οποία θα μπορεί να υποστηρίξει το πρωτόκολλο BGP, όπως λόγου χάρη ένας δρομολογητής, για την ενεργοποίηση της τεχνικής αυτής. Στη συνέχεια είναι απαραίτητο να δημιουργηθεί σύνδεση iBGP μεταξύ αυτής της συσκευής και των δρομολογητών άκρης του δικτύου. Οι δρομολογητές άκρης απαιτείται να έχουν δεσμεύσει μια συγκεκριμένη διεύθυνση IP όπως λόγου χάρη η διεύθυνση 192. 0. 2. 1/32, η οποία δεν χρησιμοποιείται στο internet αλλά είναι για ιδιωτική χρήση και να έχει οριστεί διαδρομή με τη διεύθυνση αυτή προορισμό και επόμενο βήμα ένα interface απόρριψης πακέτων (Null0 interface σε cisco δρομολογητές). Έπειτα στη συσκευή ενεργοποίησης θέτουμε μια διαδρομή με προορισμό τη διεύθυνση του θύματος και επόμενο βήμα την προκαθορισμένη διεύθυνση (εδώ 192. 0. 2. 1/32). Αυτό έχει ως αποτέλεσμα την αποστολή ενός μηνύματος BGP Update από την συσκευή ενεργοποίησης προς τον δρομολογητή άκρης όπου ανακοινώνεται η συγκεκριμένη διαδρομή. Μετά από την αποδοχή αυτής της διαδρομής στον πίνακα προώθησης του δρομολογητή άκρης, η κίνηση προς τη διεύθυνση που ανακοινώσαμε θα απορρίπτεται στην άκρη του δικτύου, μην επιτρέποντας να πλημμυρίσει το εσωτερικό του δικτύου και να προξενήσει τα προαναφερθέντα προβλήματα. Παρακάτω ακολουθεί σχηματικά η τεχνική που μόλις περιγράφηκε:



Σχήμα 27: RTBH βάσει διεύθυνσης προορισμού

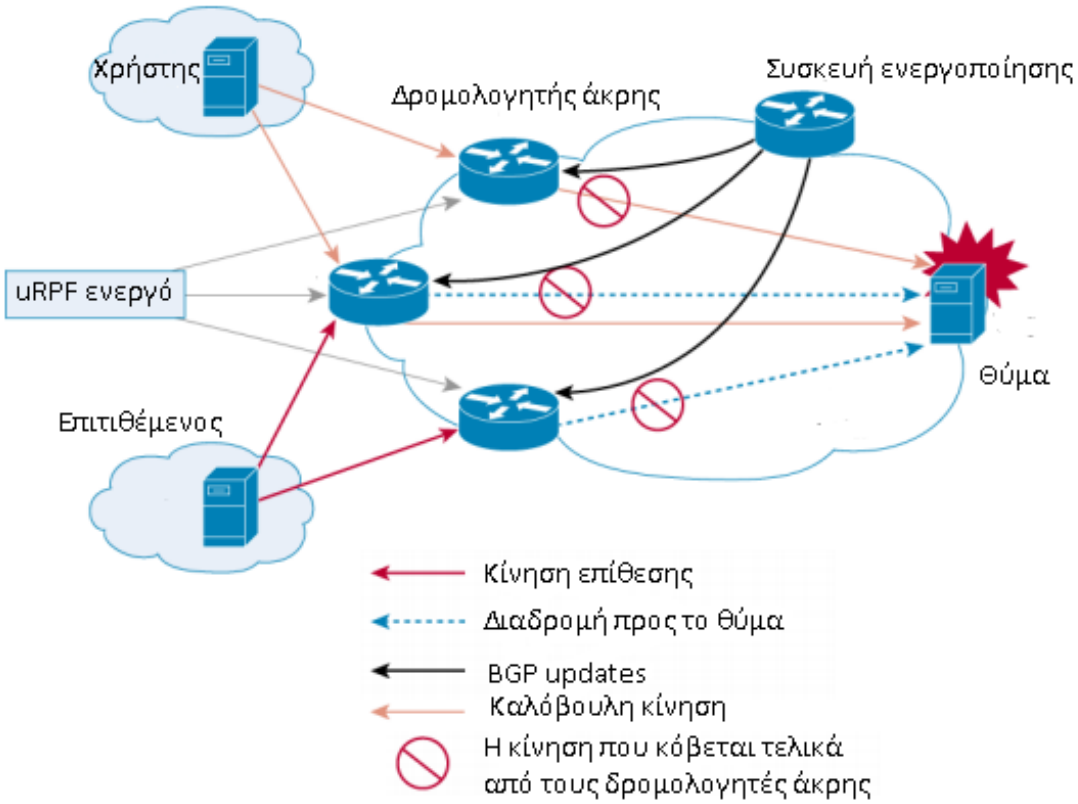
Παρόλο που βάσει αυτής της τακτικής δεν προκαλείται ζημιά στο υπόλοιπο δίκτυο μας, ουσιαστικά ολοκληρώνεται επιτυχώς η DDoS επίθεση αφού οποιαδήποτε κίνηση προς το θύμα πλέον απορρίπτεται συμπεριλαμβανομένης και της καλόβουλης. Μία λύση που προσφέρεται είναι η αλλαγή της διεύθυνσης IP του θύματος και η ενημέρωση των DNS εγγραφών με αποτέλεσμα το σύστημα-θύμα να γίνει και πάλι προσβάσιμο στο διαδίκτυο. Ωστόσο αν η επίθεση γίνεται βάσει ονόματος και όχι βάσει διεύθυνσης θα συνεχισθεί κ. ο. κ. Για αυτό τον λόγο μπορούμε να προχωρήσουμε στην επόμενη τεχνική.

4.3.4 RTBH βάσει διεύθυνσης πηγής

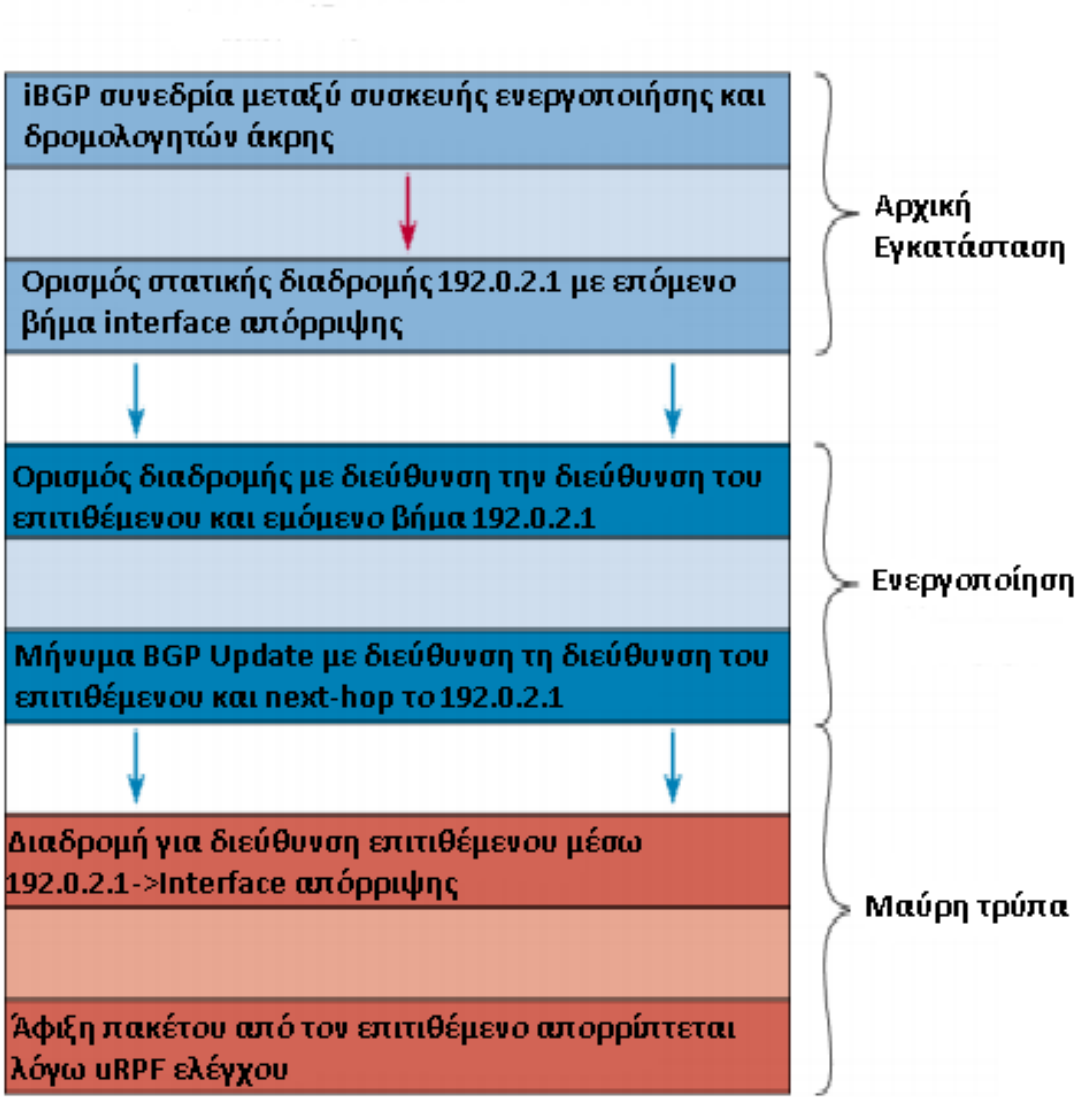
Οι μαύρες τρύπες αυτής της τεχνικής μας παρέχουν τη δυνατότητα αποκοπής κακόβουλης κίνησης στο άκρο ενός δικτύου και συγκεκριμένα στους δρομολογητές άκρης του βάσει της διεύθυνσης πηγής της. Όπως προαναφέρθηκε στην προηγούμενη προσέγγιση χάνεται και όλη η καλόβουλη κίνηση προς τη συσκευή θύμα κάτι το οποίο δεν είναι επιθυμητό . Για αυτό το λόγο αναζητήθηκε μία λύση που να σχετίζεται με την πηγή της κακόβουλης κίνησης ώστε να γίνεται δυνατή η επικοινωνία μεταξύ καλόβουλων χρηστών και στόχου της DDoS επίθεσης. Για να είναι δυνατή η εφαρμογή αυτής της τεχνικής ωστόσο, απαιτούνται δυο πολύ βασικές προϋποθέσεις:

- Η ενεργοποίηση του RPF στο interface εισόδου της κίνησης.
- Η γνώση των διευθύνσεων από τις οποίες προέρχεται η επίθεση.

Τα βήματα που ακολουθούνται για την υλοποίηση αυτής της τεχνικής είναι ίδια με την προηγούμενη, όμως σε αυτή την περίπτωση η συσκευή ενεργοποίησης καλείται να πληροί την εξής προϋπόθεση, δεν πρέπει να ανακοινώνονται οι διαδρομές που ορίζονται από αυτήν στους δρομολογητές άκρης εκτός του αυτόνομου συστήματος. Αν κάτι τέτοιο συνέβαινε τότε οι δρομολογητές που επικοινωνούσαν με τους δρομολογητές άκρης θα ενημέρωναν και αυτοί τις διαδρομές τους με αποτέλεσμα να δημιουργηθεί σύγχυση στο διαδίκτυο. Θεωρώντας τώρα ότι πληρούνται οι παραπάνω προϋποθέσεις η συσκευή ενεργοποίησης ορίζει διαδρομή με προορισμό τη διεύθυνση πηγής και και επόμενο βήμα την διεύθυνση που οδηγεί στην απόρριψη των πακέτων. Αφού γίνει αυτό ανακοινώνεται η διαδρομή μέσω BGP Update όπως και στην προηγούμενη μέθοδο στους δρομολογητές άκρης. Αυτό που συμβαίνει στην προκείμενη μέθοδο είναι το εξής, όταν φτάσει πακέτο με διεύθυνση πηγής αυτή που ορίσαμε στην διαδρομή ελέγχεται μέσω του RPF ο τρόπος που θα το προωθούσε ο δρομολογητής, όμως ο δρομολογητής βλέπει ότι αυτό το πακέτο θα το προωθούσε μέσω ενός interface απόρριψης και το απορρίπτει ακολουθώντας τις οδηγίες του αλγορίθμου RPF όπως αυτός περιγράφηκε παραπάνω. Κατά αυτόν τον τρόπο δίνεται η δυνατότητα απόρριψης κίνησης στα άκρα του δικτύου βάσει της διεύθυνσης πηγής η οποία μπορεί να είναι ακόμη και ολόκληρο δίκτυο και όχι μόνο μια διεύθυνση IPv4. Βέβαια το πρόβλημα που ανακύπτει με αυτή την μέθοδο είναι ότι με αυτόν τον τρόπο δεν υπάρχει πρόσβαση προς τα μηχανήματα των επιτιθέμενων που σε πολλές περιπτώσεις μπορεί να είναι reflectors επίθεσης (π. χ. DNS εξυπηρετητές) και να είναι χρήσιμα για το δίκτυο μας. Παρακάτω ακολουθεί σχηματικά η περιγραφείσα μέθοδος και οι απαραίτητες ενέργειες για την εφαρμογή της:



Σχήμα 28: RTBH βάσει διεύθυνσης πηγής



Σχήμα 29: Σειρά γεγονότων από την εγκατάσταση εως την απόρριψη

4.3.5 BGP Flow Specification

Είναι αρκετά εμφανές πως η προηγούμενη τεχνική απορρίπτει την κίνηση βάσει ενός και μόνο παράγοντα, την διεύθυνση πηγής. Παρότι αυτό επαρκεί σε κάποιες περιπτώσεις, υπάρχουν και σενάρια στα οποία δεν θέλουμε να υλοποιούμε μια τόσο αδρομερή τεχνική (coarse-grained). Για αυτόν τον λόγο προτάθηκε [17] ένας τρόπος

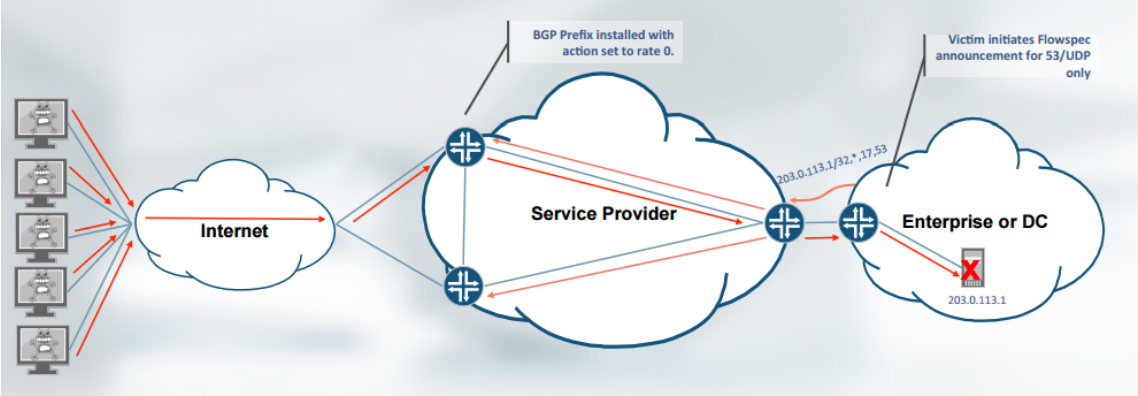
μέσω του οποίου καθίσταται δυνατή η αξιολόγηση και η απόρριψη της κίνησης βάσει περισσοτέρων χαρακτηριστικών των πακέτων στους δρομολογητές. Ειδικότερα αξιοποιείται το πεδίο NLRI του πρωτοκόλλου BGP και δεν παρέχει πλέον προσβάσιμες διαδρομές αλλά κανόνες οι οποίοι μεταφέρονται μέσω αυτού του πεδίου προς ομότιμους BGP (που υποστηρίζουν το συγκεκριμένο πρωτόκολλο) και μετά από έγκριση εγκαθιστώνται στον δρομολογητή σε μορφή φίλτρου firewall. Τα χαρακτηριστικά των κανόνων του BGP Flowspec είναι τα ακόλουθα:

- Διεύθυνση προορισμού
- Διεύθυνση πηγής
- Πρωτόκολλο IP
- Θύρα προορισμού
- Θύρα πηγής
- Τύπος ICMP
- Κωδικός ICMP
- Σημαίες του πρωτοκόλλου TCP
- Μέγεθος πακέτου
- DSCP
- Κωδικοποίηση θραύσματος (Fragmentation encoding)

Οι δράσεις που μπορούν να περιγραφούν για τα πακέτα που αντιστοιχίζονται στον εκάστοτε κανόνα είναι:

- Ορισμός ρυθμού κίνησης (Απόρριψη πακέτων)
- Δειγματοληψία και καταγραφή της συγκεκριμένης κίνησης
- Ανακατεύθυνση της κίνησης σε έναν VRF πίνακα
- Μαρκάρισμα της κίνησης (Τροποποίηση των DSCP bits του αντιστοιχιζόμενου πακέτου)

Παρακάτω ακολουθεί σχηματικά ο η συγκεκριμένη τεχνική αντιμετώπισης:



Σχήμα 30: Διαμοιρασμός BGP Flowspec κανόνα και εφαρμογή του

Το βασικό πλεονέκτημα αυτού του πρωτοκόλλου είναι ότι έχει τη βάση του στο πρωτόκολλο BGP γεγονός που κυριαρχεί στο διαδίκτυο για την αξιοπιστία, την ευελιξία και την εμπιστοσύνη που παρέχει. Πέραν τούτου η διάδοση των κανόνων μεταξύ των δρομολογητών γίνονται πλέον αυτοματοποιημένα και δεν απαιτείται η συμβολή κάποιου φυσικού προσώπου, ωστόσο είναι σύνηθες να γίνεται πάντα έλεγχος πριν την εφαρμογή κάποιου κανόνα από τους διαχειριστές του αντίστοιχου αυτόνομου συστήματος, όταν πρόκειται για eBGP συνεδρία. Επιπρόσθετα δίνεται η δυνατότητα μιας fine-grained τακτικής στο διαχωρισμό της καλόβουλης από την κακόβουλη κίνηση. Ωστόσο παρόλο που ακούγεται ως πανάκεια το συγκεκριμένο πρωτόκολλο, υποστηρίζεται από λίγους κατασκευαστές υλικών και μάλιστα λίγοι δρομολογητές που χρησιμοποιούνται σήμερα στο διαδίκτυο υποστηρίζουν το συγκεκριμένο πρωτόκολλο.

4.3.6 Κανόνες Openflow

Το πρωτόκολλο Openflow δίνει τη δυνατότητα απόρριψης κίνησης με πολύ συγκεκριμένα χαρακτηριστικά. Από την έκδοση 1. 0 εώς την 1. 5 δίνονται όλο και περισσότερες δυνατότητες ταιριάσματος πακέτων. Κάποια από τα πιο σημαντικά πεδία που μπορούν να ταιριασθούν είναι:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	TCP src port	TCP dst port	Action

Σχήμα 31: Πίνακας ταιριάσματος πρωτοκόλλου Openflow

Με τη χρήση Openflow μεταγωγέων μπορούν να εφαρμοσθούν πολύ ειδικοί κανόνες για απόρριψη κίνησης επιτρέποντας εξαιρετικά λεπτομερή προσδιορισμό της κακό-

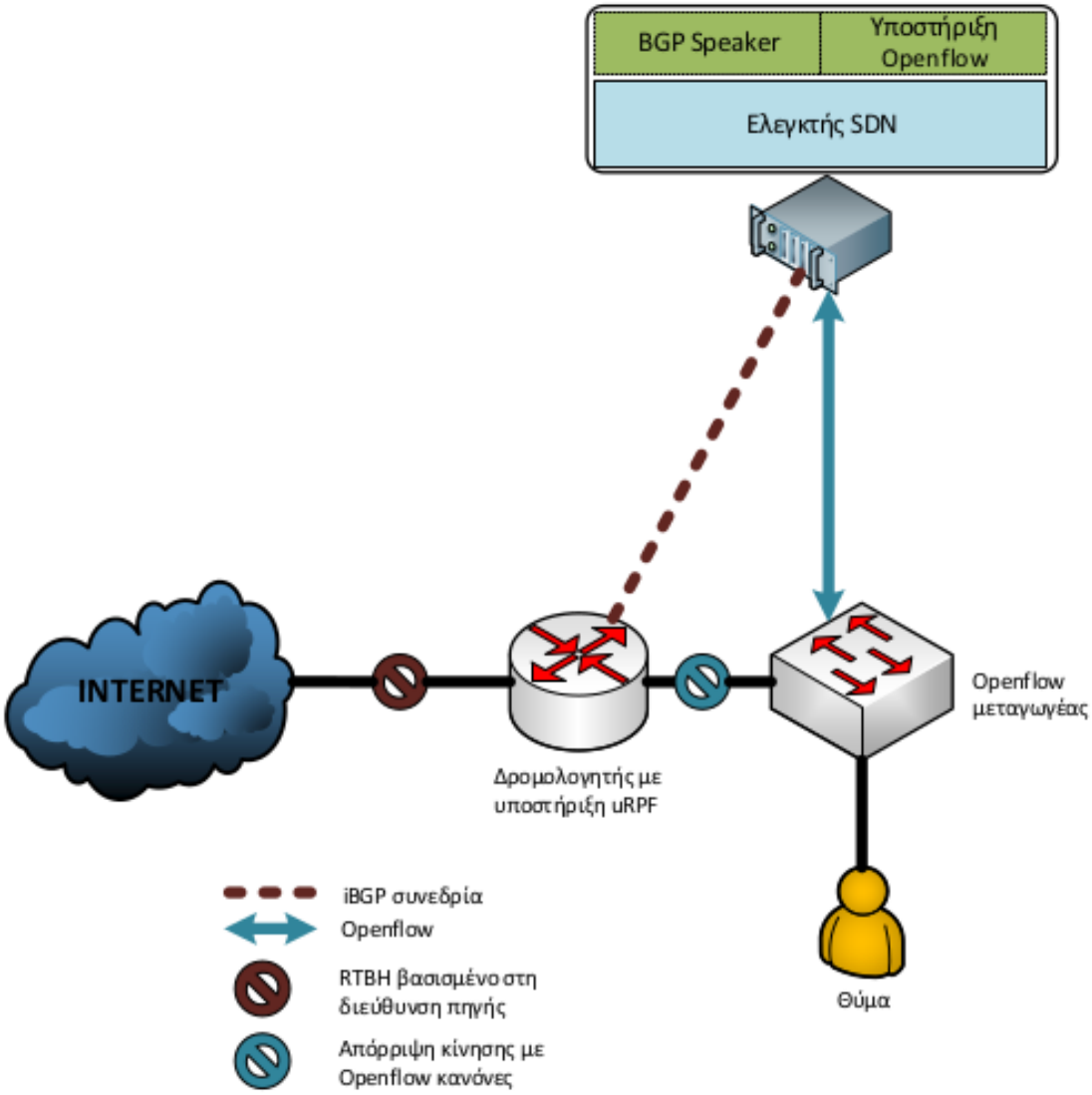
βουλης κίνησης και αντίστοιχα και της καλόβουλης έτσι ώστε να αποφευχθεί η απόρριψη της τελευταίας λόγω γενικών κανόνων όπως συμβαίνει στις τεχνικές RTBH.

4.4 Προτεινόμενη τεχνική αντιμετώπισης DDoS επιθέσεων

Η προτεινόμενη αρχιτεκτονική αντιμετώπισης DDoS επιθέσεων προϋποθέτει:

- Έναν SDN ελεγκτή που υποστηρίζει το πρωτόκολλο Openflow και λειτουργεί και ως BGP Speaker
- Έναν Openflow μεταγωγέα
- Έναν δρομολογητή άκρης που υποστηρίζει RPF

Προτείνεται ένας μηχανισμός που απορρίπτει κίνηση σε δύο επίπεδα. Αρχικά απορρίπτει κίνηση πριν αυτή εισέλθει στο εσωτερικό του δικτύου μέσω του δρομολογητή βασισμένη στη διεύθυνση πηγής της κίνησης. Σε δεύτερο επίπεδο όποια, κίνηση χρίζει απόρριψης αλλά με πιο συγκεκριμένα χαρακτηριστικά επιτρέπουμε να εισέλθει στο εσωτερικό του δικτύου και την κόβουμε στον Openflow μεταγωέα. Ο τρόπος με τον οποίο απορρίπτεται η κίνηση στον δρομολογητή είναι βάσει της τεχνικής Source-based RTBH με συσκευή ενεργοποίησης τον SDN Ελεγκτή. Ο ελεγκτής διατηρεί iBGP συνεδρία με τον δρομολογητή άκρης και όποτε αυτό είναι επιθυμητό στέλνει την διεύθυνση από την οποία δεν θέλουμε να φθάνει κίνηση στο δίκτυο μας. Αν όμως υπάρχουν κακόβουλες ροές που έχουν πιο συγκεκριμένα χαρακτηριστικά επιτρέπουμε τη διέλευση τους εντός του δικτύου και ο ελεγκτής μέσω του πρωτοκόλλου Openflow εισάγει κανόνες απόρριψης στον Openflow μεταγωγέα. Παρατηρούμε ότι με αυτόν τον τρόπο προκύπτει ευελιξία στον τρόπο απόρριψης της κακόβουλης κίνησης. Συν τις άλλοις, η αρχιτεκτονική που προτείνεται σέβεται την δομή των σημερινών δικτύων επεκτείνοντας σε αυτά SDN τομείς, χωρίς να αλλάζει τη δομή τους αλλά ούτε και τον τρόπο με τον οποίο λειτουργούν. Ακόμη η συγκεκριμένη δομή επιτρέπει και την χρήση της τεχνικής Destination-based RTBH με δυναμικό τρόπο σε περιπτώσεις όπου η κακόβουλη κίνηση στο δίκτυο υπερβεί κάποιο ποσοστό. Παρακάτω ακολουθεί σχηματικά η περιγραφείσα αρχιτεκτονική:



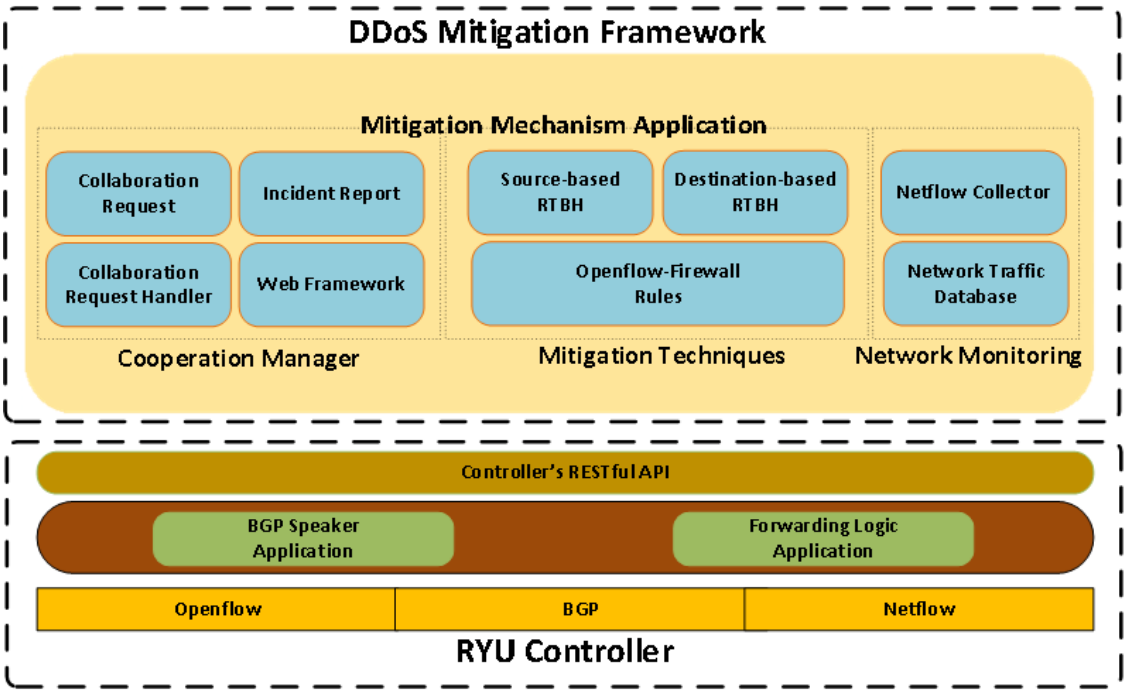
Σχήμα 32: Ο προτεινόμενος αμυντικός μηχανισμός σε δύο επίπεδα

5 Προτεινόμενος αμυντικός μηχανισμός

Ο αμυντικός μηχανισμός που προτείνεται στην παρούσα διπλωματική εργασία δίνει τις εξής δυνατότητες στο αυτόνομα συστήματα :

- Συλλογή δεδομένων κίνησης από το δίκτυο, εξάγοντας τα από τον δρομολογητή με τον τρόπο που περιγράφεται στα προηγούμενα κεφάλαια
- Ένα σύστημα αίτησης βοήθειας από συνεργαζόμενα αυτόνομα συστήματα στο μονοπάτι της διαδρομής από την οποία προέρχεται η κακόβουλη κίνηση
- Μια σειρά αμυντικών τεχνικών για απόρριψη της κακόβουλης κίνησης σε διαφορετικά επίπεδα του δικτύου

Συνοπτικά, ο προτεινόμενος μηχανισμός μπορεί να αποτυπωθεί στην παρακάτω εικόνα:



Σχήμα 33: Συνολικός μηχανισμός αντιμετώπισης DDoS επιθέσεων

Το παραπάνω framework χρησιμοποιεί τα πρωτόκολλα Openflow, BGP, Netflow κάποιες εφαρμογές του ελεγκτή Ryu καθώς και το API του για την κατασκευή τριών εφαρμογών όπως φαίνεται παραπάνω που αντικατοπτρίζουν τον προτεινόμενο αμυντικό μηχανισμό τις οποίες θα αναλύσουμε εκτενέστερα παρακάτω.

5.1 Συλλογή δεδομένων κίνησης του δικτύου

Η εφαρμογή υποστηρίζει ένα σύστημα παρακολούθησης των δεδομένων κίνησης του δικτύου. Συγκεκριμένα δίνεται η δυνατότητα λήψης και αποθήκευσης δεδομένων από δικτυακές συσκευές που υποστηρίζουν το πρωτόκολλο Netflow v5 [18]. Με αυτόν τον τρόπο καθίσταται δυνατή η παρακολούθηση της κίνησης του δικτύου. Ωστόσο ο λόγος ύπαρξης δυνατότητας παρακολούθησης του δικτύου σε αυτό τον μηχανισμό ανάγεται στη συνεργασία των αυτόνομων συστημάτων. Συγκεκριμένα καθίσταται δυνατή η αντιστοίχιση της κακόβουλης κίνησης (των IP διευθύνσεων από τις οποίες προέρχεται) με τα γειτονικά αυτόνομα συστήματα από τα οποία διέρχεται. Κατά αυτόν τον τρόπο μπορούν να σταλούν στα συνεργαζόμενα αυτόνομα συστήματα μόνο οι διευθύνσεις οι οποίες τα αφορούν. Με αυτόν τον τρόπο στέλνεται σε κάθε AS μόνο η πληροφορία που σχετίζεται με αυτό και όχι όλες οι διευθύνσεις από τις οποίες εκκινείται η επίθεση.

Πέραν τούτου όμως η συλλογή δεδομένων από το δίκτυο μπορεί να λειτουργήσει ως δικλείδα ασφαλείας για την συμμετοχή σε έναν συνεργαζόμενο μηχανισμό αντιμετώπισης επιθέσεων. Για παράδειγμα στο μοντέλο παρόχου-πελάτη ας υποθέσουμε ότι υπάρχει εμπιστοσύνη μεταξύ πελάτη και παρόχου και αιτείται ο πελάτης την απόρριψη από συγκεκριμένες IP διευθύνσεις. Η κίνηση που εξέρχεται από τον δρομολογητή άκρης του ISP εισέρχεται στον δρομολογητή άκρης του πελάτη και αντίστροφα. Συνεπώς μπορεί να ταυτισθεί η ύπαρξη της κακόβουλης κίνησης που φθάνει για την οποία ζητείται απόρριψη από το αιτηθέν αυτόνομο σύστημα (στη συγκεκριμένη περίπτωση από τον πελάτη). Με αυτόν τον τρόπο μπορεί να εξασφαλισθεί ότι ο μηχανισμός δεν μπορεί να εκμεταλλευθεί εύκολα για να απορριφθούν καλόβουλα πακέτα από συνεργαζόμενα αυτόνομα συστήματα γεγονός που τον χρίζει πιο ασφαλή. Το επιχείρημα γίνεται εμφανές μέσα από το παρακάτω σχήμα:

5.2 Διαχειριστής συνεργασίας

Ο διαχειριστής συνεργασίας επιτελεί 5 βασικές λειτουργίες:

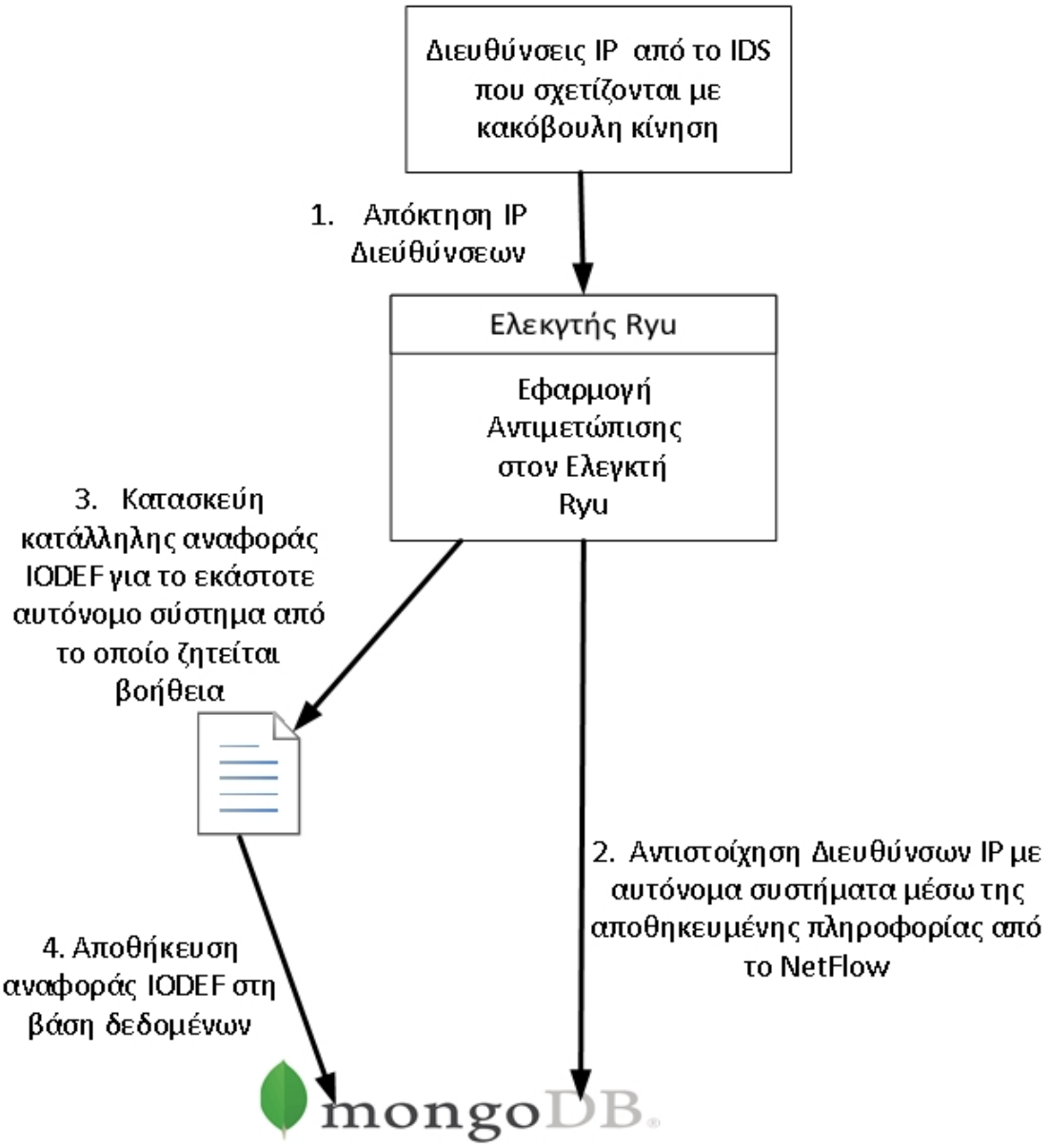
- Κατασκευή κατάλληλης αναφοράς με στοιχεία της επίθεσης για το εκάστοτε συνεργαζόμενο αυτόνομο σύστημα και αποθήκευση στη βάση δεδομένων.
- Κατασκευή κατάλληλου BGP μηνύματος που να δείχνει στην σωστή αναφορά
- Λήψη BGP μηνύματος που δείχνει σε αναφορά και ανάκτηση της
- Επεξεργασία και εξαγωγή δεδομένων από την αναφορά
- Διαδικτυακή εφαρμογή για αποστολή της IODEF αναφοράς

Η διαδικασία από την πλευρά του θύματος είναι η εξής: Αρχικά λαμβάνονται οι πηγές της επίθεσης. Στη συνέχεια όπως περιγράφηκε στην προηγούμενη ενότητα αντι-

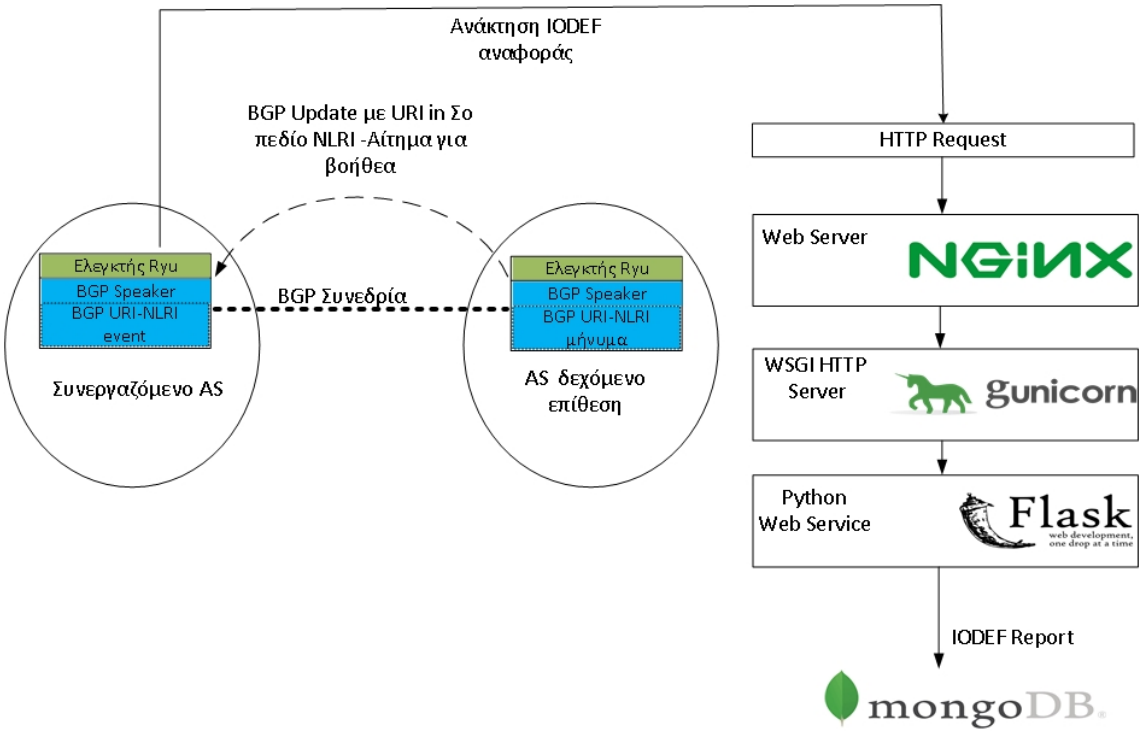
στοιχίζονται οι διευθύνσεις αυτές με τα γειτονικά συνεργαζόμενα αυτόνομα συστήματα. Έπειτα κατασκευάζεται μια αναφορά IODEF για το καθένα η οποία περιέχει πληροφορίες σχετικές με την επίθεση καθώς και τις διευθύνσεις που αιτούμαστε να απορριφθεί κίνηση από αυτές η οποία αποθηκεύεται στη βάση δεδομένων. Τέλος ανακτάται ένα αναγνωριστικό κλειδί αυτής της αναφοράς από τη βάση δεδομένων και κατασκευάζεται ένα uri μέσω του οποίου μπορεί να ανακτήσει το συνεργαζόμενο αυτόνομο σύστημα την αναφορά που το αφορά. Το uri αυτό οδηγεί σε μια διαδικτυακή εφαρμογή που δίνοντας το κατάλληλο κλειδί επιστρέφει μια αναφορά IODEF σε JSON μορφή. Το uri αυτό στέλνεται με την τεχνική που αναφέρεται στο τρίτο κεφάλαιο μέσω του πρωτοκόλλου BGP στα συνεργαζόμενα αυτόνομα συστήματα και συγκεκριμένα στους ελεγκτές SDN αυτών που εκτελούν την προτεινόμενη εφαρμογή.

Από την πλευρά του συνεργαζόμενου αυτόνομου συστήματος εκτελούνται οι παρακάτω ενέργειες: Κατά τη λήψη μηνύματος BGP που δείχνει σε uri μήνυμα ενεργοποιείται ο χειριστής τέτοιων μηνυμάτων μέσω του οποίου καθίσταται δυνατή η ανάκτηση του uri. Στη συνέχεια ζητείται το συγκεκριμένο uri και ανακτάται η αναφορά IODEF . Ο χειριστής εξάγει τις κατάλληλες πληροφορίες από αυτήν και ελέγχει αν υπάρχει η κίνηση που του ζητείται να απορριφθεί. Σε περίπτωση που ισχύει η ταύτιση της κίνησης και αποφασιστεί να συνεργαστεί, κατασκευάζονται κατάλληλες αναφορές με τον τρόπο που αναφέρεται παραπάνω για τα προσκείμενα σε αυτό συνεργαζόμενα αυτόνομα συστήματα με τις διευθύνσεις που του αιτήθηκαν να απορρίψει. Στις IODEF αναφορές που θα κατασκευάσει προστίθεται το αρχικό uri του μηνύματος για να λειτουργήσει ως δικλείδα ασφαλείας όσον αφορά στην αλλοίωση της αρχικής πληροφορίας.

Παρακάτω ακολουθεί σχηματικά όλη η περιγραφείσα διαδικασία:



Σχήμα 34: Κατασκευή κατάλληλης αναφοράς IODEF για τα συνεργαζόμενα αυτόνομα συστήματα



Σχήμα 35: Αίτημα για βοήθεια και απόκτηση της IODEF αναφοράς

Είτε στην περίπτωση του θύματος είτε στην περίπτωση του συνεργαζόμενου αυτόνομου συστήματος είναι απαραίτητη με κάποιον τρόπο η απόρριψη της κακόβουλης κίνησης κάτι το οποίο γίνεται δυνατό με το τρίτο μέρος του προτεινόμενου μηχανισμού που είναι οι τεχνικές αντιμετώπισης.

5.3 Τεχνικές αντιμετώπισης

. Μέσω της εφαρμογής που υλοποιήθηκε δίνονται τρεις διαφορετικές τεχνικές αντιμετώπισης DDoS επιθέσεων αξιοποιήσιμες ανάλογα με τις δυνατότητες του ίδιου του δικτύου. Συνεπώς είτε το δίκτυο έχει Openflow μεταγωγείς είτε όχι και πάλι καθίσταται δυνατή η εφαρμογή κανόνων για την απόρριψη κακόβουλης κίνησης βάσει της διεύθυνσης πηγής. Εδώ κρίνεται και πάλι αναγκαίο να γίνει διάκριση μεταξύ του ίδιου του θύματος και των συνεργαζόμενων ASeS που εκτελείται η εφαρμογή όσων αφορά στις τεχνικές αντιμετώπισης.

Στην πλευρά του θύματος μπορούν να εφαρμοσθούν και οι τρεις τεχνικές αντιμετώπισης:

- Κανόνες openflow L2-L4 με συγκεκριμένα χαρακτηριστικά
- Source-based RTBH για εισαγωγή κανόνων στον δρομολογητή βάσει διεύθυνσης πηγής

- Destination-based RTBH για την απόρριψη όλης της κίνησης και καλόβουλης και κακόβουλης προς το θύμα.

Από την άλλη στην πλευρά των συνεργαζόμενων αυτόνομων συστημάτων μπορούν να εφαρμοσθούν μόνο οι δύο πρώτες τεχνικές για την απόρριψη της κακόβουλης κίνησης όσο το δυνατόν πιο κοντά στην πηγή της.

Με αυτή την εφαρμογή δίνεται η δυνατότητα απόρριψης της κακόβουλης κίνησης εντός του αυτόνομου συστήματος σε πολλά επίπεδα και με διαφορετικούς τρόπους με το αντίστοιχο trade-off. Πιο συγκεκριμένα η εισαγωγή κανόνων στον Openflow μεταγωγέα δίνει τη δυνατότητα για απόρριψη κακόβουλων ροών με πιο συγκεκριμένα χαρακτηριστικά αφού βασίζεται στις δυνατότητες ταιριάσματος του πρωτοκόλλου Openflow που υποστηρίζει μέχρι και πεδία του επιπέδου μεταφοράς. Από την άλλη η τεχνική Sb-RTBH απορρίπτει την κακόβουλη κίνηση πριν εισέλθει εντός του αυτόνομου συστήματος δημιουργώντας με αυτόν τον τρόπο τη δυνατότητα για αποσυμφόρηση των διαύλων του δικτύου εντός του αυτόνομου συστήματος, ωστόσο επιτρέπει μόνο την εισαγωγή διεύθυνσης πηγής της κακόβουλης κίνησης και επίσης δεν επιτρέπει και καθόλου κίνηση προς τις διευθύνσεις αυτές από οποιονδήποτε εντός του αυτόνομου συστήματος. Τέλος σε περιπτώσεις όπου υπάρχει μεγάλο ποσοστό κακόβουλης κίνησης που καταλαμβάνει τη γραμμή και δημιουργεί μεγάλο πρόβλημα στο δίκτυο μπορεί να χρησιμοποιηθεί η μέθοδος Db-RTBH που βγάζει το θύμα εκτός δικτύου και εξασφαλίζει την επιτυχία της επίθεσης, αλλά εξασφαλίζει και την ορθή επικοινωνία των υπολοίπων του αυτόνομου συστήματος με το υπόλοιπο διαδίκτυο.

Συνολικά αυτή η εφαρμογή δίνει τη δυνατότητα εφαρμογής κανόνων απόρριψης κίνησης σε διάφορα επίπεδα εντός του δικτύου και επιτρέπει με αυτόν τον τρόπο την εκμετάλλευση διάφορων δικτυακών συσκευών για τη συμμετοχή τους στην αντιμετώπιση της επίθεσης με ποικίλλους τρόπους και πολλές διαφορετικές δυνατότητες.

6 Θέματα υλοποίησης

6.1 Αρχές προσομοίωσης

Για την αξιολόγηση του προτεινόμενου μηχανισμού αντιμετώπισης DDoS επιθέσεων κρίθηκε απαραίτητο να κατασκευασθούν εικονικά οι κατάλληλες τοπολογίες, ώστε να επαληθευθούν τα επιμέρους του τμήματα. Συγκεκριμένα για την προσομοίωση του Openflow μεταγωγέα αλλά και τη σύνδεση του με τον ελεγκτή χρησιμοποιήθηκε το Mininet [29] ενώ για την προσομοίωση των δρομολογητών χρησιμοποιήθηκε το GNS3, στα οποία αναφερόμαστε παρακάτω.

6.1.1 Mininet

Το mininet αποτελεί έναν εξομοιωτή δικτύου που επιτρέπει την δημιουργία εικονικών δικτύων που μπορούν να περιέχουν εικονικούς χρήστες, μεταγωγείς, ελεγκτές και συνδέσεις μεταξύ τους. Το κύριο πλεονέκτημα του mininet είναι η υποστήριξη Openflow μεταγωγέων καθώς και τη ευκολία στην σύνδεση ελεγκτών σε αυτούς. Πέραν τούτου το mininet έχει τα παρακάτω πλεονεκτήματα:

- Παρέχει μια απλή και εύχρηστη πλατφόρμα δοκιμών για δικτυακές τοπολογίες.
- Επιτρέπει την ανάπτυξη εφαρμογής ταυτόχρονα από διαφορετικούς προγραμματιστές την ίδια τοπολογία.
- Επιτρέπει την κατασκευή πολύπλοκων τοπολογιών με μεγάλη ευκολία χωρίς την ύπαρξη πραγματικού δικτύου.
- Περιέχει γραμμή εντολών μέσω της οποίας μπορούμε να συνδεθούμε στις συσκευές της εκάστοτε τοπολογίας.
- Έχει έναν αριθμό έτοιμων τοπολογιών αλλά επιτρέπει και την κατασκευή τους είτε γραφικά είτε μέσω μιας διεπαφής προγραμματισμού εφαρμογών γραμμένη σε Python για τον ορισμό περισσότερων λεπτομερειών που αφορούν την τοπολογία.

Το mininet όπως και τα λειτουργικά συστήματα χρησιμοποιεί εικονικοποιήση των στοιχείων του βασισμένη σε διεργασίες (process) του πυρήνα (kernel) του λειτουργικού συστήματος στο οποίο εκτελείται. Συγκεκριμένα μπορούν να φιλοξενηθούν στο εικονικό δίκτυο ένας μεγάλος αριθμός χρηστών και Openflow μεταγωγέων. Από την έκδοση 2. 2. 26 κάθε host υλοποιείται με τη χρήση network namespaces, μία αρκετά ελαφριά εικονικοποιήση που δημιουργεί ξεχωριστές διεργασίες για κάθε host οι οποίες περιέχουν στοιχεία που σχετίζονται με τους πίνακες δρομολόγησης τους, τους πίνακες arp κ. α. Το mininet μπορεί να δημιουργήσει στον χώρο του πυρήνα ή στον χώρο

του χρήστη τις δικτυακές συσκευές που υποστηρίζει τις οποίες τις συνδέει μεταξύ τους μέσω εικονικού Ethernet (veth ζευγάρια). Παρότι προς το παρόν το Mininet εξαρτάται από τον πυρήνα των Linux, γίνεται προσπάθεια να υποστηρίζεται και σε άλλα λειτουργικά συστήματα όπως το FreeBSD ή το Solaris. Τέλος είναι άξιο αναφοράς ότι το mininet είναι αμιγώς γραμμένο σε Python και έχει και ελάχιστα σημεία σε C.

Συνολικά το mininet είναι ένα περιβάλλον που μπορεί να εγκατασταθεί εύκολα, παρέχει κλιμακωσιμότητα στην ανάπτυξη δικτύων, ξεκινά πολύ γρήγορα αφού είναι εφαρμογή και όχι υλικό και αποτελεί μια λύση μέσω της οποίας ρυθμίζονται και επαναρυθμίζονται ταχύτατα οι εικονικές συσκευές. Το mininet εκτελεί πραγματικό κώδικα αναλόγως με τον ελεγκτή που θα επιλεχθεί και μας δίνει τη δυνατότητα σύνδεσης του εικονικού δικτύου με πραγματικά δίκτυα. Από την άλλη λόγω της εικονικοποιήσης δεν μπορούμε να πάρουμε πραγματικά αποτελέσματα για τις συσκευές και για τις επιδόσεις τους (όσον αφορά στους μεταγωγείς).

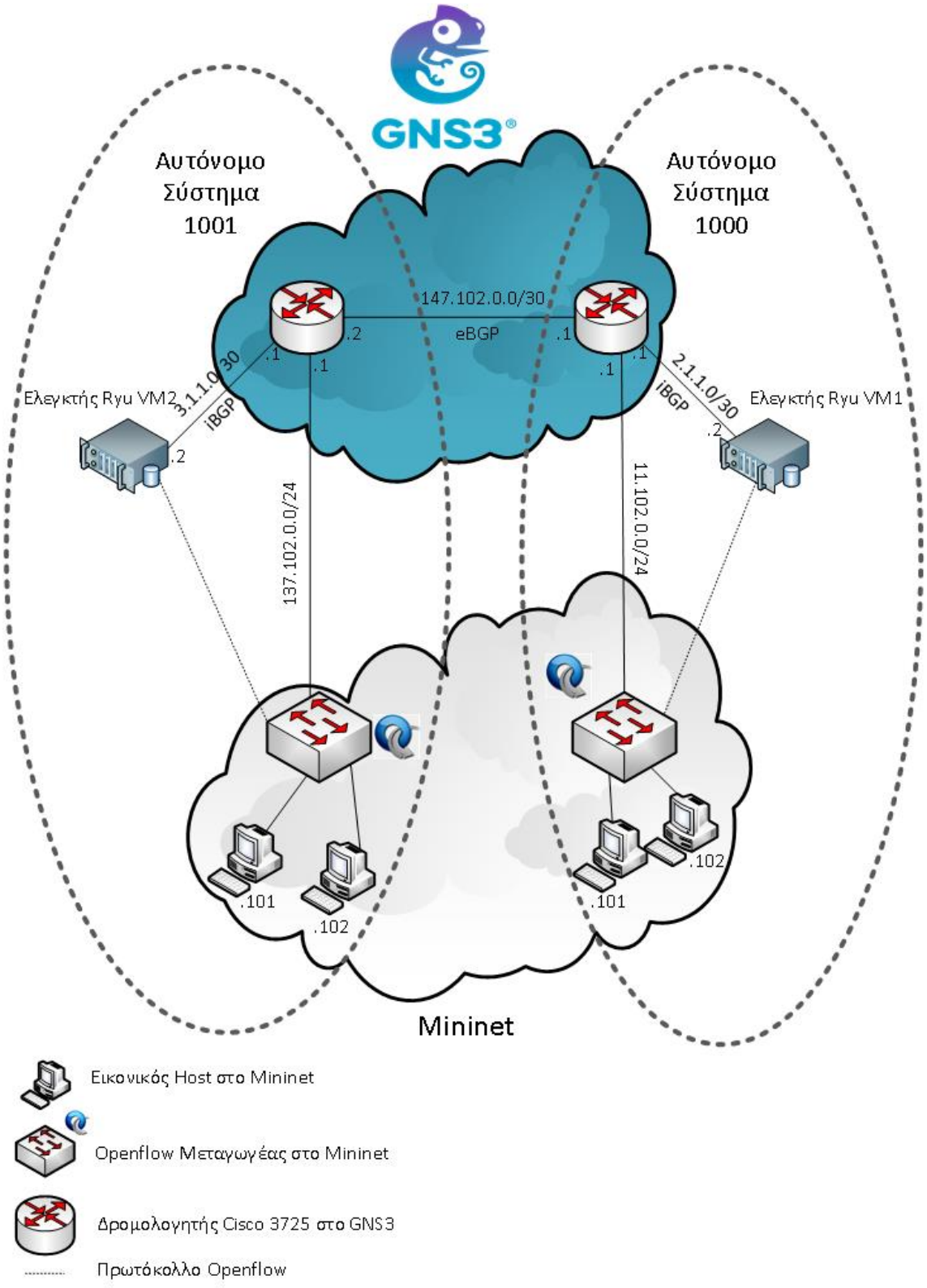
6.1.2 GNS3

Ο GNS3 [30] είναι ένας εξομοιωτής δικτύου που επιτρέπει την κατασκευή πολύπλοκων δικτύων που απαρτίζονται είτε από εικονικές συσκευές, είτε από πραγματικές συσκευές είτε απο συνδυασμό τους. Ο GNS3 δεν υποστηρίζει Openflow μεταγωγείς για αυτό στην κατασκευή της πλατφόρμας δοκιμών χρησιμοποιήθηκε συνδυαστικά ο GNS3 με το mininet. Ο GNS3 επιτρέπει την εισαγωγή εικονικών δρομολογητών μέσω των images τους, δίνοντας μας τη δυνατότητα με χαρακτηριστική ευκολία να δοκιμάζουμε και να ρυθμίζουμε διαφορετικές συσκευές. Ένα από τα κυριότερα πλεονεκτήματα του GNS3 είναι η εισαγωγή τόσο εικονικών όσο και πραγματικών μηχανημάτων στην δικτυακή τοπολογία επεκτείνοντας με αυτόν τον τρόπο τις δυνατότητες για testing. Η σύνδεση των πραγματικών συσκευών ή των εικονικών μηχανημάτων με τις εικονικές συσκευές του GNS3 γίνεται μέσω UDP Tunnels μέσω του Loopback interface του host που εκτελεί την εφαρμογή, σε μια διάφανη διαδικασία κατά την οποία ο χρήστης δεν το αντιλαμβάνεται, αλλά ούτε και οι ίδιες οι συσκευές.

Στην παρούσα διπλωματική ο δρομολογητής που χρησιμοποιήθηκε είναι ο Cisco 3725 ο οποίος υποστηρίζει τόσο το Unicast Reverse Path Forwarding αλλά και το πρωτόκολλο Netflow. Η εξομοίωση των δρομολογητών γίνεται με τη βοήθεια του Dynamips, μιας εφαρμογής που εξομοιώνει τους Cisco δρομολογητές και πάλι σε μια διάφανη διαδικασία μέσω του GNS3.

6.1.3 Πλατφόρμα δοκιμών

Το testbed που δημιουργήσαμε, υλοποιήθηκε σε δύο εικονικά μηχανήματα του εργαστηρίου. Συγκεκριμένα στο πρώτο VM εκτελούμε τόσο το GNS3 και το mininet συνδέοντας τα στην παρακάτω διάταξη.



Σχήμα 36: Η πλατφόρμα δοκιμών

Επίσης σε κάθε VM εκτελείται ο ελεγκτής Ryu και συγκεκριμένα εκτελείται το αρ-

χείο που δημιουργήθηκε για τις ανάγκες του προτεινόμενου μηχανισμού αλλά και το έτοιμο πρόγραμμα simple_switch_13. py μέσω του οποίου το switch ενημερώνεται με κανόνες για την κανονική μεταγωγή των πακέτων στο εκάστοτε δίκτυο. Πέραν τούτου, ρυθμίσθηκαν καταλλήλως οι δρομολογητές ώστε να βρίσκονται σε διαφορετικά αυτόνομα συστήματα, αλλά και για την ενεργοποίηση του uRPF στα αντίστοιχα interfaces. Επιπρόσθετα δημιουργήσαμε τεχνητές BGP συνεδρίες μεταξύ των ελεγκτών και των δρομολογητών για την χρήση της τεχνικής Source-Based RTBH και ορίσαμε στους δρομολογητές να μην διαφημίζονται τα prefixes που προέρχονται από τον ελεγκτή ούτε εντός του δικτύου αλλά ούτε και σε BGP ομότιμους άλλου αυτόνομου συστήματος.

6.2 Αποθήκευση Δεδομένων

6.2.1 NoSQL

Η αποθήκευση των δεδομένων στην παρούσα διπλωματική εργασία έγινε αμιγώς στην βάση δεδομένων MongoDB [31]. Πριν προχωρήσουμε σε λεπτομέρειες που σχετίζονται με την MongoDB θα κάνουμε μια μικρή εισαγωγή στις NoSQL βάσεις δεδομένων.

Οι NoSQL (non SQL, non relational) βάσεις δεδομένων παρέχουν ένα μηχανισμό για αποθήκευση και ανάκτηση δεδομένων, ο οποίος έχει αρκετά διαφορετική δομή από τις σχεσιακές βάσεις δεδομένων. Η ύπαρξη τους χρονολογείται από το 1970 ωστόσο η πλήρης υιοθέτηση τους άρχισε να γίνεται από τις αρχές του 21ου αιώνα εξαιτίας των σημερινών αναγκών. Η κύρια χρήση τους αφορά εφαρμογές πραγματικού χρόνου και πολλών δεδομένων (big data). Παρότι το μοντέλο τους διαφέρει αρκετά από τις κλασσικές βάσεις δεδομένων, υποστηρίζονται και ερωτήματα που θυμίζουν αρκετά τα ερωτήματα των σχεσιακών βάσεων δεδομένων. Τέλος ένα αρκετά σύνηθες σενάριο είναι ο συνδυασμός τους με τις σχεσιακές βάσεις δεδομένων χάριν ταχύτητας ανάκτησης δεδομένων.

Το κύριο κίνητρο για την υιοθέτηση τους είναι η απλότητα σχεδιασμού, η οριζόντια κλιμακωσιμότητα σε συστοιχίες μηχανημάτων το οποίο αποτελεί πολύ μεγάλο πρόβλημα για τις σχεσιακές βάσεις δεδομένων. Η ταχύτητα ανάκτησης δεδομένων ανάγεται στον τρόπο με τον οποίο αποθηκεύεται η πληροφορία. Μια NoSQL βάση μπορεί να αποθηκεύσει τα δεδομένα της μέσω ενός από τους παρακάτω τρόπους:

- ζεύγος κλειδιού-τιμής
- σε ευρεία στήλη
- σε γράφο
- σε έγγραφα

Η ευχέρεια που δίνεται στον τρόπο αποθήκευσης των δεδομένων δίνει μεγάλη ευελιξία στον σχεδιασμό μιας NoSQL βάσης αλλά εξαρτάται και κατά κύριο λόγο στα δεδομένα που χρίζουν αποθήκευσης.

6.2.2 MongoDB

Η MongoDB είναι μια βάση δεδομένων στην οποία τα δεδομένα αποθηκεύονται σε μορφή εγγράφου δηλαδή σε μορφή παρόμοια με το πρότυπο JSON και μπορεί να χαρακτηρισθεί ως μία NoSQL βάση δεδομένων. Υποστηρίζει μια σειρά από ερωτήματα στην βάση, όπως λόγου χάρη ερωτήματα εύρους, αναζητήσεις κανονικών εκφράσεων όπως και αναζητήσεις βάσει πεδίου όπως οι κλασικές βάσεις δεδομένων. Επίσης δίνεται η δυνατότητα να επιστραφούν μόνο κάποια πεδία της εγγραφής που ζητείται όπως και ένα τυχαίο δείγμα των αποτελεσμάτων του ερωτήματος.

Η MongoDB παρέχει υψηλή απόδοση, υψηλή διαθεσιμότητα και αυτόματη κλιμάκωση. Μερικά από τα χαρακτηριστικά της βάσης που συντελούν στα παραπάνω είναι:

- Υποστήριξη ενσωμάτωσης αντικειμένων στο μοντέλο δεδομένων της (nested documents), δυνατότητα η οποία μειώνει την ανάγκη για πολλαπλές αναγώσεις/εγγραφές στους χώρους αποθήκευσης.
- Παρέχει δείκτες (indexes) οι οποίοι μπορούν να δεικτοδοτήσουν κλειδιά και σε ενσωματωμένα έγγραφα (documents).
- Η υπηρεσία λειτουργίας αντιγράφων της βάσης (replication) παρέχει αυτόματη ανάκαμψη από βλάβες (automatic failover) και πλεονασμό δεδομένων (data redundancy).
- Παρέχει οριζόντια κλιμάκωση ως βασική της υπηρεσία και παρέχει την δυνατότητα κατακερματισμού των δεδομένων (sharding) σε ένα σύνολο (cluster) υπολογιστών.

6.3 Κατασκευή Web Service για ανάκτηση των IODEF αναφορών

Για την κατασκευή του web site χρησιμοποιήθηκαν ο nginx ως reverse proxy, το Flask microframework και το gunicorn ως application server για τη σύνδεση της εφαρμογής του Flask με τον nginx. Προκύπτει συνεπώς μια αρχιτεκτονική τεσσάρων επιπέδων όπου ο nginx χειρίζεται τα αιτήματα που σχετίζονται με στατικά αρχεία, το Guni corn χειρίζεται αιτήματα που αφορούν δυναμικά αιτήματα και τα προωθεί στην εφαρμογή του Flask, το οποίο με τη σειρά του επικοινωνεί με τη βάση δεδομένων για την ανάκτηση της ζητηθείσας πληροφορίας.

6.3.1 Nginx

Ο nginx [25] είναι ένας δωρεάν, ανοικτού κώδικα, υψηλής απόδοσης HTTP server και reverse proxy, καθώς και IMAP/POP3 proxy server. Το nginx project ξεκίνησε από τον Igor Sysoev το 2002 εστιάζοντας έντονα στον υψηλό συγχρονισμό, στον υψηλής απόδοσης προσανατολισμό και ελαχιστοποίηση της χρήσης μνήμης. Το 2004 έγινε διαθέσιμη η πρώτη έκδοση για το κοινό.

Ο nginx είναι ένας από τους λίγους servers προγραμματισμένους ώστε να αντιμετωπίζουν το C10K πρόβλημα. Αυτό σημαίνει να μπορούν να διαχειριστούν πάνω από 10. 000 ταυτόχρονες συνδέσεις με μικρή κατανάλωση μνήμης (2. 5 MB ανά 10k αδρανών HTTP keep-alive συνδέσεις). Αντίθετα με τους παραδοσιακούς servers, ο nginx δεν στηρίζεται σε threads για να χειρίζεται τα requests. Αντίθετα χρησιμοποιεί πολύ πιο επεκτάστιμη event-driven (ασύγχρονη) αρχιτεκτονική. Αυτή η αρχιτεκτονική χρησιμοποιεί λίγη, αλλά το πιο σημαντικό, προβλεπόμενη ποσότητα μνήμης κάτω από φόρτο.

Ακόμα κι αν κάποιος ή κάποια δεν περιμένει να χειριστεί χιλιάδες ταυτόχρονων αιτημάτων, μπορεί ακόμα να επωφεληθεί από την υψηλή απόδοση και μικρή κατανάλωση μνήμης του nginx. Ο nginx επεκτείνεται σε όλες τις κατευθύνσεις: από το πιο μικρό VPS μέχρι και σε επίπεδο clusters από servers.

Ο nginx εξυπηρετεί μια σειρά από μεγάλης αναγνωρισιμότητας sites, όπως: YouTube, Netflix, Hulu, Pinterest, CloudFlare, Airbnb, WordPress. com, GitHub, SoundCloud, Zynga, Eventbrite, Zappos, Media Temple, Heroku, RightScale, Engine Yard και NetDNA.

6.3.2 Gunicorn

Το Gunicorn [26] είναι ένα Web Server Gateway Interface για Python. Μέσω του Gunicorn δίνεται η δυνατότητα να εκτελεσθούν οι εφαρμογές του Flask, με σκοπό την υποστήριξη χειρισμού δυναμικών αιτημάτων. Συγκεκριμένα το Gunicorn είναι ο ενδιάμεσος κρίκος μεταξύ του web server (Nginx στην προκείμενη περίπτωση) και του Framework που χρησιμοποιούνται (Flask). Το Gunicorn είναι βασισμένο στο μοντέλο master-worker, δηλαδή μία διεργασία master χειρίζεται μια λίστα από workers που αναμένουν τον χειρισμό των αιτημάτων. Υποστηρίζει τόσο σύγχρονους αλλά και ασύγχρονους workers. Το gunicorn μπορεί να χρησιμοποιηθεί και ως middleware:

- Για την δρομολόγηση των αιτημάτων στις διάφορες εφαρμογές του Web Framework
- Για τον χειρισμό πολλαπλών αιτημάτων ή frameworks κάτω από την ίδια διεργασία
- Για εξισορρόπηση φόρτου και απομακρυσμένη επεξεργασία, προωθώντας αιτήματα και απαντήσεις σε διαφορετικό δίκτυο

6.3.3 Flask

Το Flask [27] αποτελεί ένα micro web framework για ανάπτυξη εφαρμογών σε Python, βασισμένο στο Werkezeug toolkit και στη μηχανή προτύπων Jinja2 και πλέον είναι ένα από τα πιο διαδεδομένα web framework στον κόσμο αφού είναι μια γρήγορη και απλή λύση για την ανάπτυξη ιστοσελίδας. Παρότι δεν περιέχει στην πυρήνα του πολλές βιβλιοθήκες, παρέχονται κάποιες αρκετά χρήσιμες:

- Για την κατασκευή συναρτήσεων που σχετίζονται με κάποιο συγκεκριμένο URL
- Τη μηχανή προτύπων Jinja2 για την διευκόλυνση κατασκευής της δομής των ιστοσελίδων σε HTML.
- Για χειρισμό συνεδριών
- Για την ανάλυση των HTTP αιτημάτων και την ευέλικτη κατασκευή απαντήσεων.
- Διαδραστικό web-based αναλυτή σφαλμάτων

Το βασικό πλεονέκτημα του Flask είναι ότι αποτελεί ένα ελαφρύ framework, ευκόλως επεκτάσιμο και ευέλικτο στην ανάπτυξη εφαρμογών.

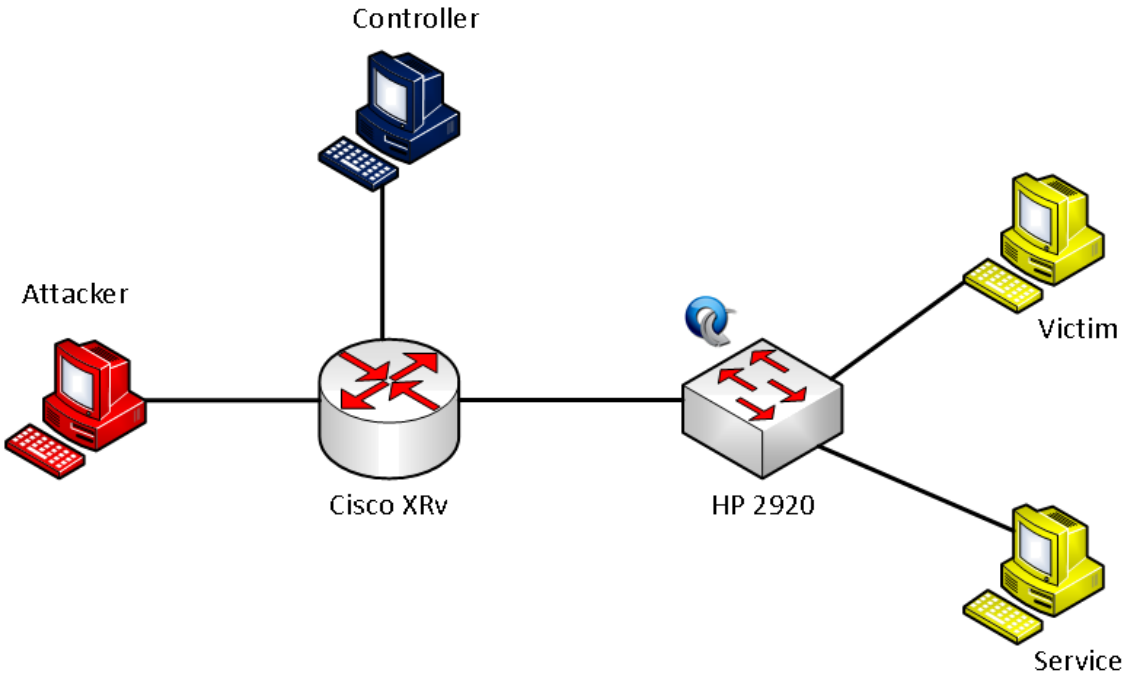
7 Πειραματική διαδικασία και αποτελέσματα

Για την εξαγωγή ρεαλιστικών αποτελεσμάτων κρίθηκε αναγκαία η υλοποίηση του αμυντικού μηχανισμού σε πραγματικό εξοπλισμό. Αξίζει να αναφερθεί ότι η επικοινωνία μεταξύ των αυτόνομων συστημάτων αλλά και η λήψη δεδομένων που σχετίζονται με την κίνηση στο δίκτυο υλοποιήθηκαν μόνο σε επίπεδο προσομοίωσης λόγω του ότι δεν χρειαζόντουσαν ρεαλιστικές μετρικές για την αξιολόγηση τους. Για αυτό και η συγκεκριμένη πειραματική διαδικασία απαρτίζεται από τα τρία παρακάτω βασικά σημεία:

- Κατασκευή της παρούσας δικτυακής αρχιτεκτονικής σε πραγματικά μηχανήματα και ρύθμιση ενός ευάλωτου service για γίνει επίθεση.
- Κατασκευή και αποστολή μια multi-vector επίθεσης.
- Υλοποίηση τακτικής για την αντιμετώπιση της βάσει του προτεινόμενου μηχανισμού και αποκόμιση επιθυμητών μετρικών.

7.1 Ρύθμιση και υλοποίηση του αμυντικού μηχανισμού σε πραγματικό εξοπλισμό

Για την υλοποίηση των πειραμάτων, χρησιμοποιήθηκαν 4 VMs ένας δρομολογητής Cisco XRv καθώς και ένα HP Openflow switch 2900 όπως φαίνεται παρακάτω:



Σχήμα 37: Εξοπλισμός που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων.

Όπως γίνεται αντιληπτό και από την παραπάνω εικόνα η κατανομή των πόρων έγινε ως εξής:

- 1 VM για την εγκατάσταση του ελεγκτή Ryu με σκοπό την αποστολή εντολών για την αντιμετώπιση της επίθεσης όπως περιγράφεται παραπάνω.
- 1 VM μέσω του οποίου γίνεται replay μία επίθεση που κατασκευάστηκε
- 2 VMs με ρόλο θύματος στην επίθεση, το ένα για να φιλοξενήσει το ευάλωτο service και το άλλο για την αποστολή επιθέσεων πλημμύρας.

Το ευάλωτο service που φιλοξενήθηκε στο ένα VM είναι ουσιαστικά ένας Flask Web Server πίσω από έναν Reverse Proxy (Nginx) ο οποίος επέστρεφε μία εικόνα όταν ζητούσαμε ένα συγκεκριμένο url. Ο λόγος που επιλέχθηκε ένα τέτοιο service είναι ότι σε περίπτωση ανάκτησης πολλές φορές της εικόνας υπήρχε μεγάλη επίδραση στον επεξεργαστή του VM. Ακόμη μπορεί με αυτό τον τρόπο να διατηρήσουμε καλόβουλες συνεδρίες για να αξιολογηθεί μια σειρά μετρικών που επηρεάζονται κατά τη διάρκεια της επίθεσης. Παρακάτω γίνεται πιο εκτενής αναφορά στα είδη αυτών των μετρικών και ο λόγος χρησιμοποίησης τους.

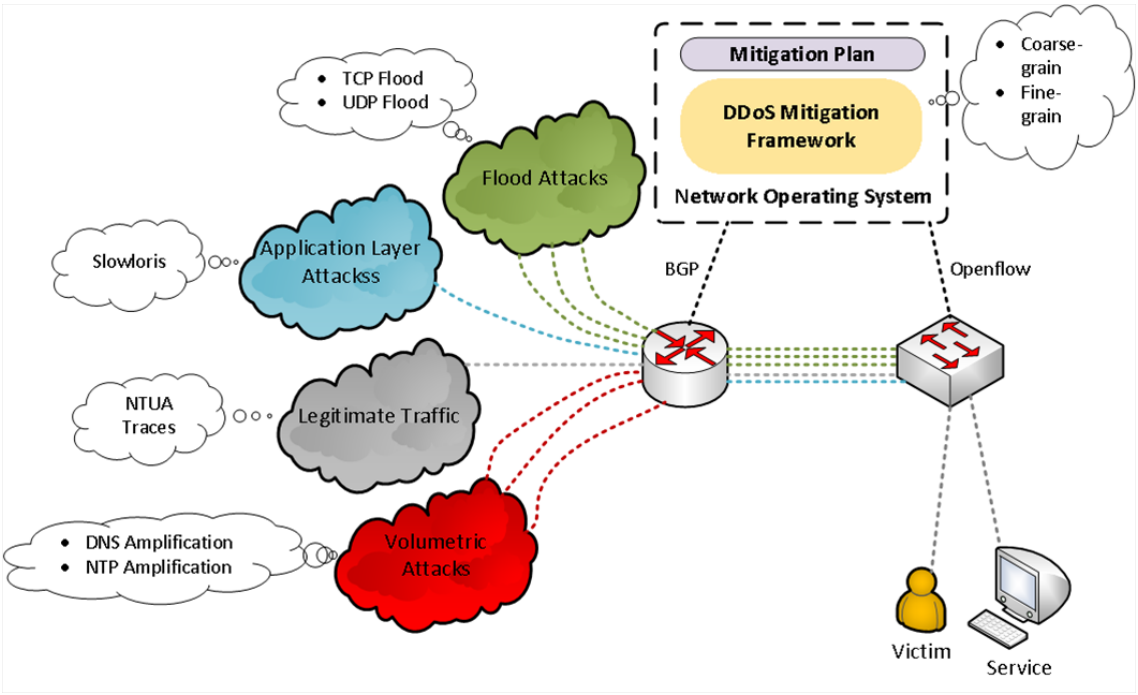
7.2 Κατασκευή DDoS multi-vector επίθεσης

Λόγω του ότι είναι επιθυμητό να δείξουμε τις δυνατότητες του μηχανισμού που προτείνεται επιλέχθηκε η κατασκευή μιας επίθεσης με συγκεκριμένα χαρακτηριστικά και όχι η επιλογή ενός υπάρχοντος data set επίθεσης. Συγκεκριμένα για την δημιουργία των πακέτων της επίθεσης χρησιμοποιήθηκαν μια σειρά από εργαλεία που σχετίζονται με penetration testing και δίνουν τη δυνατότητα παραγωγής κακόβουλης κίνησης διαφόρων ειδών. Αφού κατασκευάστηκε η κακόβουλη κίνηση κατά τον τροόπο τον οποίο ήταν επιθυμητός, στη συνέχεια αποθηκεύτηκε σε ένα αρχείο . pcap . Με τη χρήση αυτού του αρχείου πολλαπλασιάστηκε η συγκεκριμένη κίνηση αλλάζοντας ταυτόχρονα τις IP πηγής των πακέτων θεωρώντας ότι η επίθεση προέρχεται από ένα σύνολο 10400 μοναδικών διευθύνσεων IPv4. Ειδικότερα τα εργαλεία που χρησιμοποιήθηκαν είναι:

- το hping3 [32] εργαλείο της πλατφόρμας Kali για την κατασκευή UDP, TCP, ICMP κίνησης
- ένα εργαλείο σε python για την υλοποίηση slowloris επίθεσης [33]
- το Mausezahn (Mz) [34] για την κατασκευή DNS και NTP απαντήσεων με σκοπό την προσομοίωση DNS και NTP amplification επιθέσεων.
- τα tcpreplay, tcprewrite, editcap, mergecap για τον πολλαπλασιασμό της κίνησης και την αλλαγή διευθύνσεων πηγής της καθώς επίσης και για την αποστολή της κίνησης από τον επιτιθέμενος στα θύματα.
- το ab (apache benchmarking tool) καταχρηστικά για δημιουργία HTTP Flood επίθεσης.

Ακόμη πέραν της κακόβουλης κίνησης για την αξιολόγηση ενός τέτοιου μηχανισμού είναι αναγκαίο παράλληλα με την αποστολή της να σταλεί και καλόβουλη για την πλήρη προσομοίωση πραγματικών συνθηκών. Για αυτό το λόγο χρησιμοποιήσαμε traces του εργαστηρίου Netmode θεωρώντας την ως καλόβουλη κίνηση ύστερα από κατάλληλη επεξεργασία, καθώς και το εργαλείο wrk ένα εργαλείο http benchmarking για κατασκευή καλόβουλης HTTP κίνησης με στόχο την διατήρηση συνεδριών http με τον http server.

Παρατίθενται παρακάτω οι τύποι των επιθέσεων που κατασκευάστηκαν από τους οποίους όμως δεν χρησιμοποιήθηκαν όλοι στην πειραματική διαδικασία που ακολουθήθηκε ωστόσο αξιολογήθηκαν και χρησιμοποιήθηκαν μόνο αυτές οι οποίες μπορούσαν να δείξουν με πιο εμφανή τρόπο τις δυνατότητες του αμυντικού μηχανισμού καθώς:



Σχήμα 38: Καλόβουλη και κακόβουλη κίνηση που κατασκευάστηκε

Η επίθεση που τελικά αποφασίσαμε να γίνει με σκοπό την αποκόμιση αποτελεσμάτων απαρτίζεται από τρεις περιόδους με κλιμακούμενη ένταση και πολυποίκιλες επιπτώσεις στα θύματα και προφανώς και στο αντίστοιχο αυτόνομο σύστημα. Συγκεκριμένα συνολικά στην επίθεση χρησιμοποιήθηκαν 4 είδη επιθέσεων τα οποία φαίνονται παρακάτω και τα οποία συνδυάστηκαν στις φάσεις της επίθεσης με σκοπό να βλάψουν με οποιονδήποτε τρόπο το θύμα καθώς και ίδιο το αυτόνομο σύστημα στο οποίο αυτό βρίσκεται. Για να γίνει σαφέστερο διακρίναμε την επίθεση σε 3 φάσεις όπου από την πρώτη προς την τρίτη αυξανόταν τόσο η ένταση της επίθεσης αλλά προσθέταμε κι άλλο ένα είδος επίθεσης με σκοπό να βλάψουμε σε πολλά διαφορετικά σημεία τόσο το ίδιο το δίκτυο όσο και την επίδοση της υπηρεσίας του http server αλλά και το ίδιο το θύμα. Επίσης κρίθηκε απαραίτητη μια μηδενική φάση κατά την οποία υπάρχει ομαλή λειτουργία τόσο της υπηρεσίας που γίνεται η επίθεση αλλά και στο ίδιο το δίκτυο στο οποίο λαμβάνει χώρα. Ακολουθούν σχηματικά οι τρεις διαφορετικές περίοδοι της επίθεσης:

	No attack	First Phase	Second Phase	Third Phase
HTTP Slowloris		✓	✓	✓
HTTP GET Flood		✓	✓	✓
TCP Flood			✓	✓
UDP Flood				✓

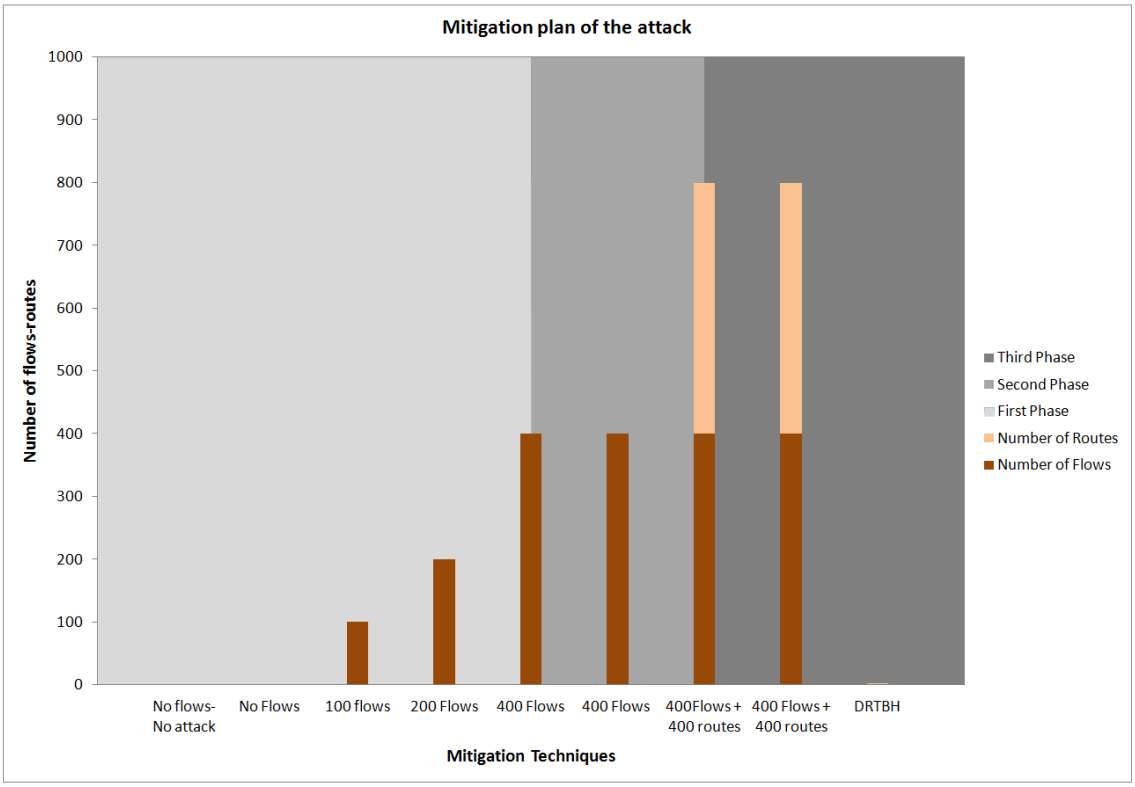
Σχήμα 39: Φάσεις της επίθεσης

Τέλος θεωρήθηκε ότι η κακόβουλη κίνηση προερχόταν από τα υποδίκτυα 99. 0. 0-102. 0/24 και συγκεκριμένα από 100 διευθύνσεις σε καθένα από τα 102 παραπάνω υποδίκτυα. Δεν υπήρχε κάποιος συγκεκριμένος λόγος επιλογής αυτών των διευθύνσεων ωστόσο απαραίτητο για τον αμυντικό μηχανισμό που σχεδιάστηκε είναι η γνώση των διευθύνσεων πηγής της επίθεσης κάτι που γίνεται εφικτό όταν η επίθεση είναι αμιγώς κατασκευασμένη από εμάς. Ακόμη θεωρήσαμε ότι από αυτές τις διευθύνσεις κάποιες από εξ αυτών έχουν αξία να είναι προσβάσιμες από το αυτόνομο σύστημα το οποίο δέχεται επίθεση. Ο λόγος που έγινε αυτό είναι για να δείξουμε πιθανές παράπλευρες απώλειες που προκύπτουν από τις διαφορετικές τεχνικές του αμυντικού μηχανισμού όσον αφορά στην αντιμετώπιση της επίθεσης.

7.3 Αντιμετώπιση της επίθεσης και αποκόμιση επιθυμητών μετρικών

Η αντιμετώπιση της επίθεσης βασίστηκε σε όλες τις δυνατές τεχνικές αντιμετώπισης που παρέχεται από τον αμυντικό μηχανισμό. Συγκεκριμένα στην πρώτη περίοδο της επίθεσης επιχειρήθηκε να χρησιμοποιηθεί μόνο ο Openflow μεταγωγέας και οι δυνατότητες που παρέχονται από το πρωτόκολλο. Προσπαθήσαμε αυξάνοντας τον αριθμό των κανόνων στον μεταγωγέα να αντιμετωπισθεί η επίθεση όπως γίνεται εμφανές στα διαγράμματα που θα ακολουθήσουν. Αφού εξομαλύνθηκε η κατάσταση τόσο του δικτύου όσο και της υπηρεσίας, αυξήθηκε η ένταση της επίθεσης αλλά και μεταβλήθηκε ο τύπος της. Αυτό δημιούργησε πρόβλημα στο αυτόνομο σύστημα με ποικίλους τρόπους όπως γίνεται εμφανές και για αυτό χρειάσθηκε η συμβολή του δρομολογητή. Ειδικότερα αξιοποιήσαμε την τεχνική RTBH βασισμένη στη διεύθυνση πηγής με σκοπό να εξομαλυνθεί η κατάσταση στο αυτόνομο σύστημα. Παρότι αυτή η τεχνική έχει παράπλευρες απώλειες παρατηρούμε ότι επανέφερε το δίκτυο σε φυσιο-

λογικά πλαίσια και μείωσε τους χρόνους απάντησης των HTTP αιτημάτων αλλά και τον rtt χρόνο από το αυτόνομο σύστημα προς κάποιον υπολογιστή εκτός αυτόνομου συστήματος. Ωστόσο, η επίθεση τελικώς αυξάνεται κι άλλο σε ένταση κι επεκτείνεται κι από άλλη επίθεση πλημμύρας με αποτέλεσμα να δημιουργήσει μεγάλες απώλειες στο δίκτυο και υψηλή χρησιμοποίηση της γραμμής. Λόγω αυτού, κρίθηκε απαραίτητη η χρησιμοποίηση της τεχνικής RTBH βασισμένη στη διεύθυνση προορισμού εισάγοντας διαδρομές στον δρομολογητή με προορισμό τα θύματα μέσω null interfaces. Κατά αυτόν τον τρόπο τόσο η υπηρεσία όσο και το θύμα βγαίνουν εκτός δικτύου και απορρίπτεται όλη η κακόβουλη αλλά και όλη η καλόβουλη κίνηση προς αυτά, όμως εξομαλύνεται πλήρως η λειτουργία του δικτύου και επανέρχεται το αυτόνομο σύστημα σε φυσιολογικές μετρικές, στις οποίες δηλαδή βρίσκεται υπό φυσιολογικές συνθήκες. Ακολουθεί διαγραμματικά η τακτική αντιμετώπισης της επίθεσης:



Σχήμα 40: Πλάνο αντιμετώπισης της επίθεσης

7.3.1 Μετρικές αξιολόγησης αμυντικού μηχανισμού αντιμετώπισης DDoS επιθέσεων

Πριν ακολουθήσουν διαγραμματικά αποτελέσματα με κάποιες ενδιαφέρουσες μετρικές που προέκυψαν κατά τη διάρκεια της επίθεσης είναι επιτακτική ανάγκη να αναφερθούμε στον λόγο επιλογής τους αλλά και να αναφερθεί ότι η παρούσα αντιμετώπιση λειτούργησε σαν proof of concept του μηχανισμού και δεν αποτελεί μέθοδος

αναγνώρισης της επίθεσης. Ο λόγος δηλαδή που απεικονίζονται τα διαγράμματα είναι για την επίδειξη των τεχνικών και την επεξήγηση των επιπτώσεων που έχουν στο αυτόνομο σύστημα με τη χρήση κάποιων μετιρκών ικανών την αξιολόγηση αμυντικών μηχανισμών [35].

Η διάκριση των μετρικών που σχετίζονται με την αξιολόγηση ενός αμυντικού μηχανισμού μπορούν να κατηγοριοποιηθούν στις εξής κατηγορίες:

- Μετρικές που σχετίζονται με την επίδραση της επίθεσης σε επίπεδο πακέτων.
- Μετρικές που σχετίζονται με το στρώμα εφαρμογής και συγκεκριμένα με την HTTP υπηρεσία.
- Μετρικές για την ίδια την αντιμετώπιση της επίθεσης.

Οι μετρικές που σχετίζονται με την επίδραση της επίθεσης σε επίπεδο πακέτων είναι:

Ποσοστό απώλειας πακέτων: Είναι ο αριθμός των πακέτων ή bytes που χάνονται λόγω της αλληλεπίδρασης της καλόβουλης με την κακόβουλη κίνηση. Μπορεί να μετρηθεί εύκολα ως τον αριθμό των πακέτων που δεν έφθασαν στον επιθυμητό προορισμό και έχει αξία σε επιθέσεις που δημιουργούν συμφόρηση στο δίκτυο. Ιδανικά ένας αμυντικός μηχανισμός οφείλει να ελαχιστοποιεί αυτό το ποσοστό.

$$Loss = \frac{\sum Legitimate\,Traffic}{\sum Total\,Traffic} \times 100 \tag{1}$$

Ποσοστό κακόβουλης κίνησης που απορρίφθηκε: Ορίζεται ως το ποσοστό της κακόβουλης κίνησης που απορρίφθηκε από τον αμυντικό μηχανισμό και ιδανικά ένας μηχανισμός οφείλει να μεγιστοποιεί αυτό το ποσοστό.

$$Percentage\,of\ Attack\,Traffic\,Dropped = \frac{Attack\,Traffic}{Total\,Traffic} \times 100 \tag{2}$$

Ποσοστό καλόβουλης κίνησης που απορρίφθηκε: Ορίζεται ως το ποσοστό της καλόβουλης κίνησης που απορρίφθηκε από τον αμυντικό μηχανισμό και ιδανικά ένας μηχανισμός οφείλει να ελαχιστοποιεί αυτό το ποσοστό.

$$Benign\,Traffic\,Dropped\,Percentage = \frac{BenignPackesDropped}{TotalBenignTraffic} \times 100 \tag{3}$$

$$Normal\,Packet\,Survival\,Ratio = \frac{Legitimate\,Packets}{Attack\,Packets + Legitimate\,Packets} \tag{4}$$

Χρονική καθυστέρηση πακέτων (Delay):Ορίζεται ως ο χρόνος που απαιτείται να φθάσει το πακέτο από την πηγή στον προορισμό και απαιτείται από τον αμυντικό

σύστημα να μειώσει όσο το δυνατόν τον χρόνο αυτό.

$$Average\,End\,to\,End\,Delay = \sum Packet\,Arrival\,Time_i - Packet\,Start\,Time_i \quad (5)$$

Η δεύτερη κατηγορία αφορά στην HTTP υπηρεσία και συγκεκριμένα σε μετρικές που σχετίζονται με το στρώμα εφαρμογής: Ποσοστό αποτυχίας HTTP συναλλαγών : Για να θεωρηθεί μια HTTP συναλλαγή αποτυχημένη πρέπει να μην έχει ληφθεί απάντηση για το HTTP ερώτημα σε διάστημα 10 δευτερολέπτων [36]. Συνεπώς το ζητούμενο ποσοστό προκύπτει από την παρακάτω σχέση.

$$Packet\,Failure\,Rate = \frac{failed\,transactions}{total\,number\,of\,transactions} \times 100 \quad (6)$$

Χρόνος καθυστέρησης μεταξύ HTTP ερώτησης-απάντησης : Είναι ο χρόνος μεταξύ αποστολής της ερώτησης και λήψης ολοκληρωμένης απάντησης από την HTTP υπηρεσία.

$$Request\,Response\,Delay = T_{req} - T_{resp} \quad (7)$$

Μέση χρονική καθυστέρηση : Ορίζεται ως η μέση τιμή της χρονική καθυστέρησης λήψης της πρώτης HTTP απάντησης από τη στιγμή που έγινε το HTTP ερώτημα:

$$Average\,Latency = \frac{\sum_{i=1}^{N}(T_{req} - T_{resp})}{N} \quad (8)$$

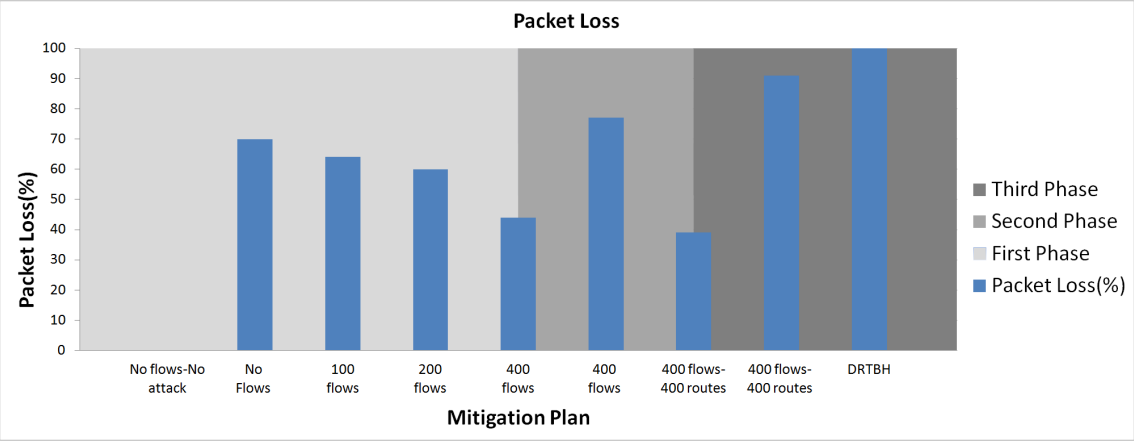
Τέλος στην τρίτη κατηγορία εξετάζουμε προβλήματα που δημιουργεί ο αμυντικός μηχανισμός στην καλόβουλη κίνηση.

Παράπλευρες απώλειες: Θεωρούμε ως παράπλευρη απώλεια τη ζημιά που κάνει ο αμυντικός μηχανισμός στην καλόβουλη κίνηση. Αρχικά σε επίπεδο προσβάσιμων IPv4 διευθύνσεων και ακόμη σε επίπεδο απόρριψης καλόβουλης κίνησης από τη χρήση κανόνων για απόρριψη της κακόβουλης κίνησης.

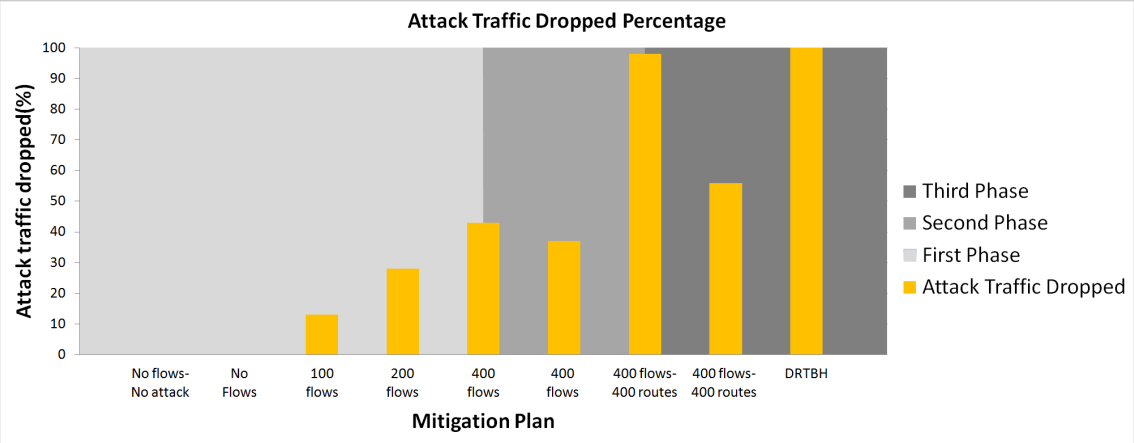
$$Collateral\,Damage = Number\,of\,IPv4\,addresses\,not\,accessible \quad (9)$$

7.3.2 Αποτελέσματα πειράματος

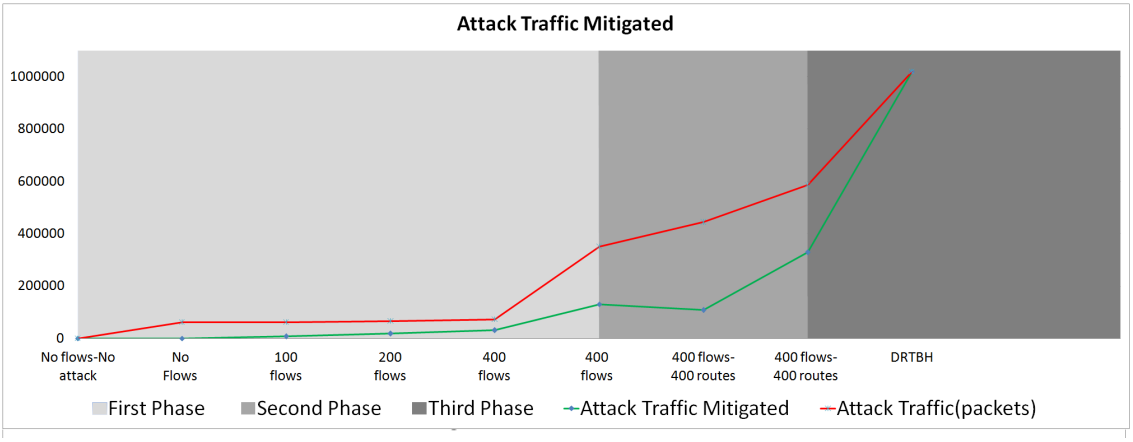
Βάσει τις παραπάνω ανάλυσης ακολουθεί και η αντίστοιχη κατηγοριοποίηση των διαγραμμάτων που κατά κύριο λόγο απεικονίζουν τις επιπτώσεις που είχε το πλάνο που εφαρμόσαμε στο δίκτυο και στις υπηρεσίες του αυτόνομου συστήματος. Κύριος στόχος του πειράματος είναι η επίδειξη των δυνατοτήτων του μηχανισμού και h ανάδειξη των επιπτώσεων που κάθε τεχνική έχει. Τα διαγράμματα που ακολουθούν απεικονίζουν τις φάσεις της επίθεσης, τον τρόπο που αντιμετωπιζόταν η επίθεση σε κάθε φάση της (άξονας x) και τις παραπάνω μετρικές στον άξονα y, εκτός της μέσης χρονικής καθυστέρησης όπου απεικονίζει την κατάσταση του δικτύου καθ' όλη τη διάρκεια της επίθεσης. Αρχικά θα παρουσιάσουμε τα αποτελέσματα των μετρικών σε επίπεδο πακέτων:



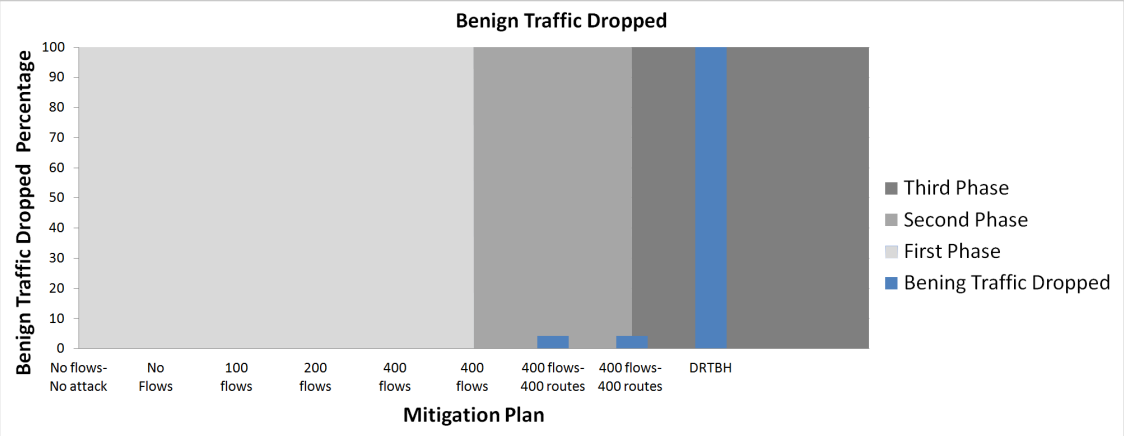
Σχήμα 41: Ποσοστό απώλειας πακέτων



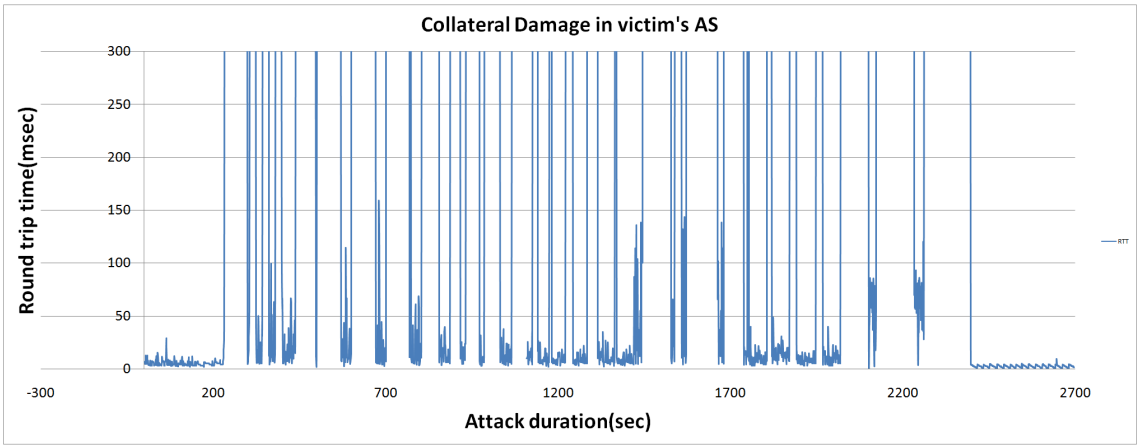
Σχήμα 42: Ποσοστό κακόβουλης κίνησης που απορρίφθηκε



Σχήμα 43: Κακόβουλη κίνηση που απορρίφθηκε-Συνολική κακόβουλη κίνηση



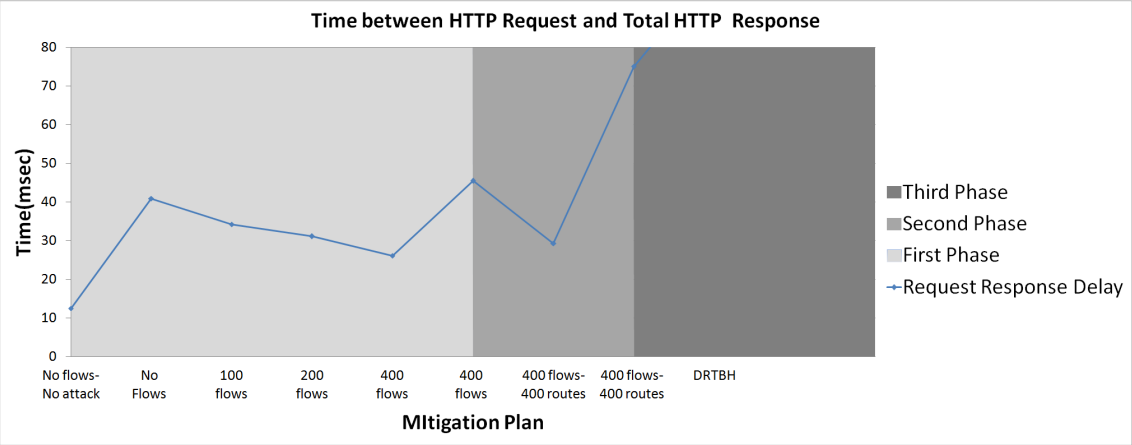
Σχήμα 44: Ποσοστό καλόβουλης κίνησης που απορρίφθηκε



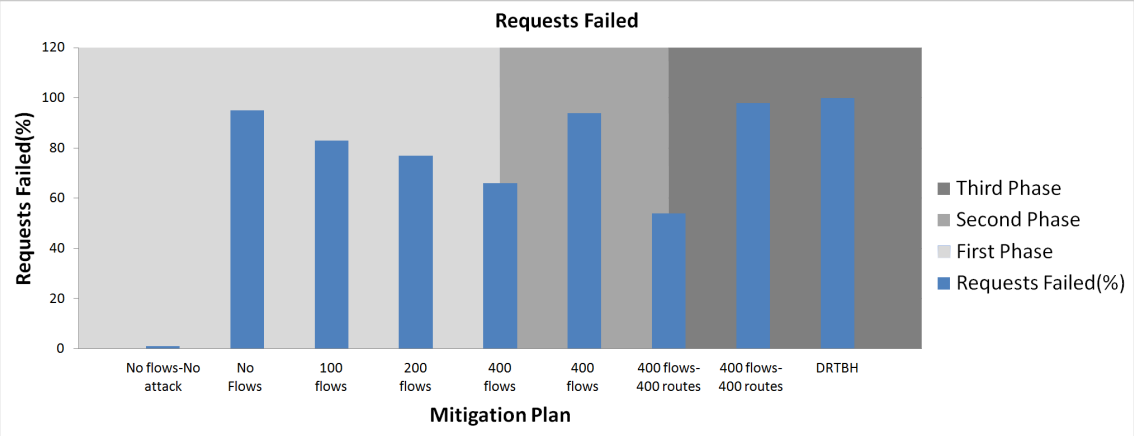
Σχήμα 45: Μέση χρονική καθυστέρηση κατά τη διάρκεια της επίθεσης

Παρατηρείται ότι ο αμυντικός μηχανισμός προσαρμόζεται στις αλλαγές της επίθεσης και σε κάθε φάση της προσπαθεί να ελαχιστοποιήσει τις απώλειες και να απορρίψει όσο το δυνατόν περισσότερη κακόβουλη κίνηση εφαρμόζοντας κάθε δυνατό είδος τεχνικής. Ωστόσο στο τέλος που επίθεση αποκτά μεγάλη ένταση και δημιουργεί μεγάλα προβλήματα στο δίκτυο με την τεχνική Db-RTBH απορρίπτεται όλη η καλόβουλη και η κακόβουλη κίνηση ωστόσο γίνεται εμφανές ότι εξομαλύνεται η κατάσταση στο υπόλοιπο δίκτυο όπως φαίνεται στο διάγραμμα με τη χρονική καθυστέρηση των πακέτων.

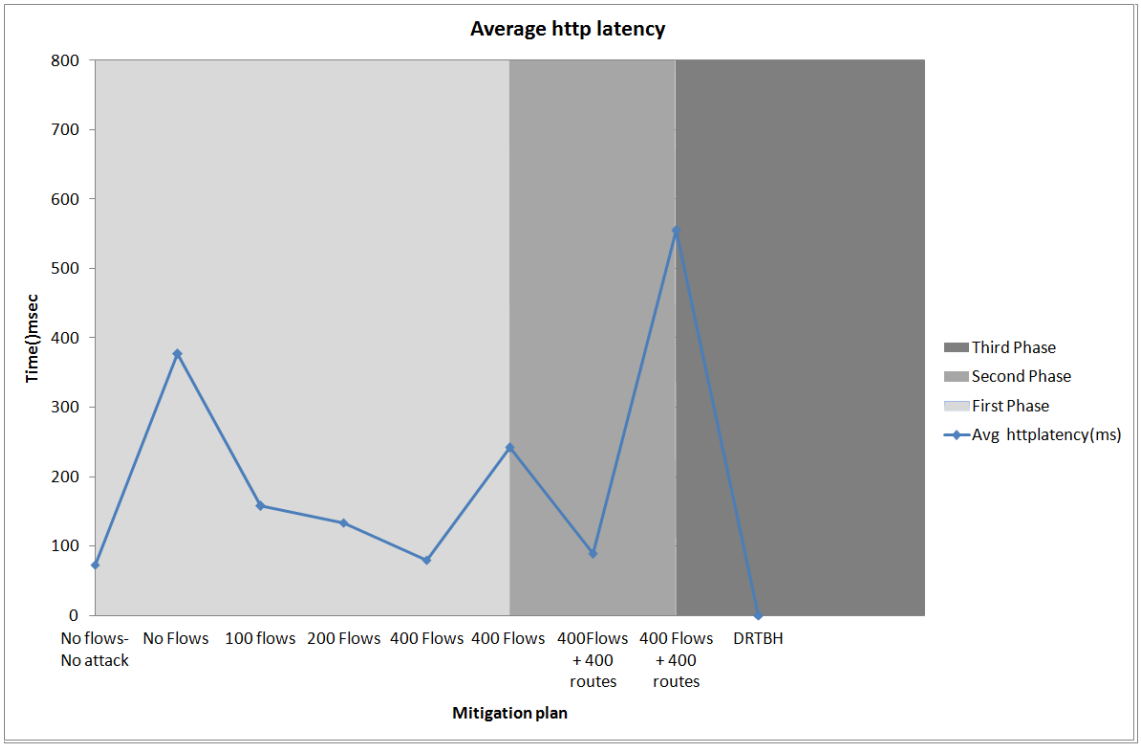
Στη συνέχεια ακολουθούν τα διαγράμματα των μετρικών που αφορούν την HTTP υπηρεσία, την επίδραση που είχε σε αυτή η επίθεση καθώς και τα αποτελέσματα που προέκυψαν μετά την τακτική που εφαρμόσθηκε για την αντιμετώπιση της.



Σχήμα 46: Αριθμός HTTP ερωτημάτων και απαντήσεων



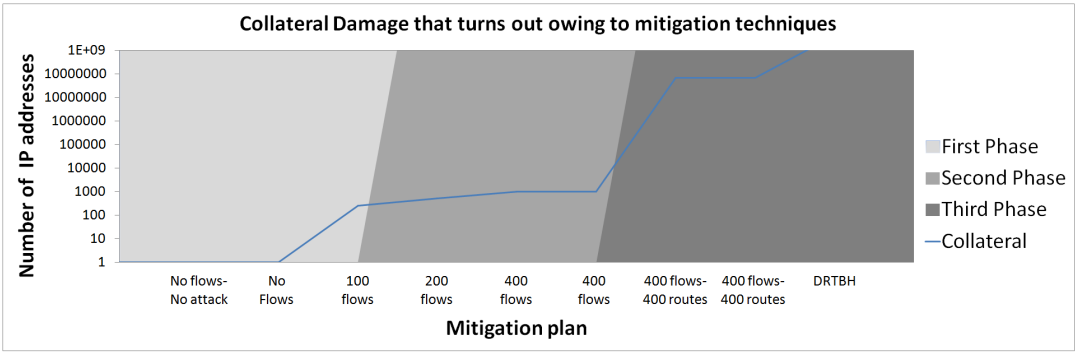
Σχήμα 47: Ποσοστό HTTP ερωτημάτων που απέτυχε



Σχήμα 48: Μέση χρονική καθυστέρηση HTTP ερωτημάτων

Όπως φάνηκε και από τα παραπάνω διαγράμματα και εδώ παρατηρείται η προσαρμογή του αμυντικού μηχανισμού στην ίδια την επίθεση βελτιώνοντας όσο αυτό καθίσταται δυνατό την ποιότητα της HTTP υπηρεσίας αυξάνοντας των αριθμό των HTTP απαντήσεων και μειώνοντας τους χρόνους άφιξης των HTTP απαντήσεων στους καλόβουλους χρήστες. Η κατάλληλη χρήση των αμυντικών τεχνικών όπως γίνεται εμφανές βοηθάει στην απόκριση της HTTP υπηρεσίας και στην εξυπηρέτηση των καλόβουλων χρηστών καθώς παράλληλα γίνεται και προσπάθεια για απόρριψη της κακόβουλης κίνησης. Τέλος καθότι δημιουργείται μεγάλο πρόβλημα στο δίκτυο αναγκαστικά βγαίνει η υπηρεσία εκτός λειτουργίας χάνοντας με αυτόν τον τρόπο όλη την καλόβουλη και κακόβουλη κίνηση και προς και από αυτήν.

Τέλος κρίθηκε αναγκαίο να παρουσιασθεί η παράπλευρη απώλεια σε ποιοτικό επίπεδο και συγκεκριμένα το σύνολο των διευθύνσεων οι οποίες με τις τεχνικές που εφαρμόσθηκαν είναι μη προσβάσιμες από το αυτόνομο σύστημα. Σε αυτό το σημείο μπορεί να γίνει αντιληπτό το trade-off που γίνεται για την αντιμετώπιση της επίθεσης σε αντιδιαστολή με τον αριθμό των διευθύνσεων που δεν μπορούν να επικοινωνήσουν πλέον με τους χρήστες και της υπηρεσίες εντός του αυτόνομου συστήματος.



Σχήμα 49: Παράπλευρες απώλειες εξαιτίας των αμυντικών τεχνικών που εφαρμόσθηκαν

8 Συμπεράσματα-Μελλοντικές Επεκτάσεις

Συνοπτικά στην παρούσα εργασία παρουσιάστηκε ένας συνεργατικός μηχανισμός αντιμετώπισης δικτυακών επιθέσεων ο οποίος μπορεί να εφαρμοσθεί σε αυτόνομα συστήματα που διατηρούν συνεργατικές σχέσεις με άλλα αυτόνομα συστήματα και δίνει τη δυνατότητα αίτησης βοήθειας από αυτά . Πέραν τούτου επιχειρήθηκε η αξιοποίηση μια σειράς τεχνικών αντιμετώπισης της επίθεσης σε διάφορα επίπεδα του δικτύου και πιο συγκεκριμένα αρχικά στον δρομολογητή και σε δεύτερο επίπεδο σε Openflow μεταγωγείς αν αυτό είναι εφικτό. Υλοποιήθηκαν οι τεχνικές που αναφέρθηκαν και δοκιμάστηκαν σε πραγματικό εξοπλισμό καθώς και εξάχθηκαν πραγματικά δεδομένα από τις επιπτώσεις που είχαν στην λειτουργία του δικτύου αλλά και στην ίδια την αντιμετώπιση της επίθεσης αυτή καθ' αυτή. Για να γίνει εφικτή η παραπάνω διαδικασία κατασκευάστηκε επίθεση με κατάλληλα χαρακτηριστικά για την επίδειξη των παραπάνω τεχνικών.

θα μπορούσαμε να κατηγοριοποιήσουμε τις επεκτάσεις της παραπάνω εργασίας σε τρία διαφορετικά σκέλη κατά αντιστοιχία με τον τρόπο κατηγοριοποίησης του μηχανισμού που προτάθηκε.

- Αξίζουν μελέτης συστήματα εμπιστοσύνης που μπορούν να εφαρμοσθούν στην προκειμένη περίπτωση αλλά και γενικά η διερεύνηση συστημάτων εμπιστοσύνης ιδανικών για την εξασφάλιση της εγκυρότητας της πληροφορίας. Στην συγκεκριμένη περίπτωση αναφερόμαστε σε συστήματα που θα μπορούσε να αποφευχθεί η εκμετάλλευση συνεργατικών μηχανισμών με σκοπό την απόρριψη καλόβουλης κίνησης.
- Έχει μεγάλο ερευνητικό ενδιαφέρον η αναζήτηση αλγορίθμων για την κατανομή των πόρων και την κατάλληλη προσαρμογή του αμυντικού μηχανισμού στις εκάστοτε επιθέσεις, καθώς και η ίδια η επέκταση του μηχανισμού στην υποστήριξη κι άλλων αμυντικών μεθόδων όπως η επέκταση του ελεγκτή Ryu για αποστολή BGP Flowspec μηνυμάτων στους δρομολογητές για πιο αδρομερή αντιμετώπιση. Ακόμη θα ήταν επιθυμητή η κατασκευή εφαρμογής για την αυτοματοποιημένη εισαγωγή κανόνων από τον ελεγκτή τόσο στους ίδιους τους hosts του αυτόνομου συστήματος όσο και στις υπόλοιπες δικτυακές συσκευές όπου αυτό είναι εφικτό. Τέλος αξίζει να αναφερθούμε στην αναζήτηση αλγορίθμων για την ελαχιστοποίηση των κανόνων απόρριψης κακόβουλης κίνησης ανά τις δικτυακές συσκευές καθώς και την βέλτιστη τοποθέτηση των κανόνων αυτών στις διάφορες δικτυακές συσκευές εντός ενός αυτόνομου συστήματος.
- Πέρα των παραπάνω χρήζει υλοποίησης ένα σύστημα πολιτικών αντιμετώπισης εκμεταλλευόμενο τον υπάρχοντα αμυντικό μηχανισμό. Δηλαδή, αναφερόμαστε σε μια σειρά από αμυντικές πολιτικές οι οποίες θα εφαρμόζονται εκμεταλλευόμενες τις τεχνικές απόρριψης που παρουσιάστηκαν ή τις τεχνικές που υποστηρίζονται από το εκάστοτε αυτόνομο σύστημα.
- Τέλος θα μπορούσε η εφαρμογή του προτεινόμενου μηχανισμού να ενσωματωθεί εντός ενός συνολικότερου αμυντικού μηχανισμού ο οποίος θα ανταποκρινό-

ταν σε όλες τις μορφές αντιμετώπισης μιας δικτυακής επίθεσης και πιο συγκεκριμένα από την πρόληψη και την αναγνώριση δικτυακών επιθέσεων μέχρι τον κατάλληλο συνδυασμό των αμυντικών τεχνικών στις διαφορετικές δικτυακές επιθέσεις που εμφανίζονται.

Α΄ Παράρτημα κώδικα-ρυθμίσεων

Α΄.1 Mininet

Ακολουθεί το script για την τοπολογία στο mininet:

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def myNet():
  #c1
    c1='192.168.1.1'
#c2
    c2='192.168.1.2'

    net = Mininet( topo=None, build=False)
    # Create nodes
    h1 = net.addHost( 'h1', mac='01:00:00:00:01:00', ip='137.102.0.101/24' )
    h2 = net.addHost( 'h2', mac='01:00:00:00:02:00', ip='137.102.0.102/24' )
    h3 = net.addHost( 'h3', mac='01:00:00:00:03:00', ip='11.102.0.101/24'  )
    h4 = net.addHost( 'h4', mac='01:00:00:00:04:00', ip='11.102.0.102/24'  )
    # Create switches
    s1 = net.addSwitch( 's1', listenPort=6634, mac='00:00:00:00:00:01' )
    s2 = net.addSwitch( 's2', listenPort=6634, mac='00:00:00:00:00:02' )

    print "*** Creating links"
    net.addLink(h1, s1, )
    net.addLink(h2, s1, )
    net.addLink(h3, s2, )
    net.addLink(h4, s2, )
    # Add Controllers
    c_1 = net.addController( 'c1', controller=RemoteController, ip=c1, port=6633)
    c_2 = net.addController( 'c2', controller=RemoteController, ip=c2, port=6633)
    net.build()
    # Connect each switch to a different controller
    s1.start( [c_1] )
    s2.start( [c_2] )
    h1.cmd('ip route add 0.0.0.0/0 via 137.102.0.1')
    h2.cmd('ip route add 0.0.0.0/0 via 137.102.0.1')
    h3.cmd('ip route add 0.0.0.0/0 via 11.102.0.1')
    h4.cmd('ip route add 0.0.0.0/0 via 11.102.0.1')

    CLI( net )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNet()
```

Δημιουργία της τοπολογίας στο mininet

Α΄.2 Ryu

Στο αρχείο ryu/lib/packet/bgp. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
BGP_OPT_CAPABILITY = 2  # RFC 5492

BGP_CAP_MULTIPROTOCOL = 1  # RFC 4760
BGP_CAP_ROUTE_REFRESH = 2  # RFC 2918
BGP_CAP_CARRYING_LABEL_INFO = 4  # RFC 3107
BGP_CAP_GRACEFUL_RESTART = 64  # RFC 4724
BGP_CAP_FOUR_OCTET_AS_NUMBER = 65  # RFC 4893
BGP_CAP_ENHANCED_ROUTE_REFRESH = 70  # https://tools.ietf.org/html/\
# draft-ietf-idr-bgp-enhanced-route-refresh-05
BGP_ENABLE_URI=75 #CHANGES Chosen by us not assinged a specific number in rfc
BGP_CAP_ROUTE_REFRESH_CISCO = 128  # in cisco routers, there are two\
# route refresh code: one using the capability code of 128 (old),
# another using the capability code of 2 (new).

BGP_ATTR_FLAG_OPTIONAL = 1 << 7
BGP_ATTR_FLAG_TRANSITIVE = 1 << 6
BGP_ATTR_FLAG_PARTIAL = 1 << 5
BGP_ATTR_FLAG_EXTENDED_LENGTH = 1 << 4
```

Ορισμός τιμής για το BGP Capability NLRI-URI, γραμμή 65

```
@_OptParamCapability.register_type(BGP_CAP_ENHANCED_ROUTE_REFRESH)
class BGPOptParamCapabilityEnhancedRouteRefresh(_OptParamEmptyCapability):
    pass
#CHANGED
@_OptParamCapability.register_type(BGP_ENABLE_URI)
class BGPOptParamCapabilityEnableUri(_OptParamEmptyCapability):
    pass
#CHANGED

@_OptParamCapability.register_type(BGP_CAP_GRACEFUL_RESTART)
class BGPOptParamCapabilityGracefulRestart(_OptParamCapability):
    _CAP_PACK_STR = "!H"
```

Δημιουργία του BGP Capability NLRI-URI, γραμμές 1248-1252

```
def __init__(self, type_=BGP_MSG_UPDATE,
             withdrawn_routes_len=None,
             withdrawn_routes=[],
             total_path_attribute_len=None,
             path_attributes=[],
             nlri=[],uri_nlri=False,
             len_=None, marker=None):
    super(BGPUpdate, self).__init__(marker=marker, len_=len_, type_=type_)
    self.withdrawn_routes_len = withdrawn_routes_len
    self.withdrawn_routes = withdrawn_routes
    self.total_path_attribute_len = total_path_attribute_len
    self.path_attributes = path_attributes
    self._pathattr_map = {}
    for attr in path_attributes:
        self._pathattr_map[attr.type] = attr
    self.nlri = nlri
    self.uri_nlri=uri_nlri #CHANGED #Used in order to know if BGPUpdate contains uri
```

Ορισμός πεδίου στο BGPUpdate για μεταφορά URI στο NLRI πεδίο, γραμμή 2345

```
@property
def pathattr_map(self):
    return self._pathattr_map

def get_path_attr(self, attr_name):
    return self._pathattr_map.get(attr_name)
#CHANGED
def get_withdrawn_routes_len(self):
    return self.withdrawn_routes_len
def get_total_path_attribute_len(self):
    return self.total_path_attribute_len
def get_nlri(self):
    return self.nlri
#CHANGED
```

Δημιουργία μεθόδων για έλεγχο των πεδίων του πακέτου BGPUpdate, γραμμές 2352-2359

```
def serialize_tail(self):
    # fixup
    binroutes = bytearray()
    for r in self.withdrawn_routes:
        binroutes += r.serialize()
    self.withdrawn_routes_len = len(binroutes)
    binpathattrs = bytearray()
    for pa in self.path_attributes:
        binpathattrs += pa.serialize()
    self.total_path_attribute_len = len(binpathattrs)
    binnlri = bytearray()
    #CHANGES #Extend nlri to accept string
    if self.uri_nlri:
        binnlri.extend(self.nlri[0])
    else:
        for n in self.nlri:
            binnlri += n.serialize()

    msg = bytearray()
    offset = 0
    msg_pack_into('!H', msg, offset, self.withdrawn_routes_len)
    msg += binroutes
    offset += 2 + self.withdrawn_routes_len
    msg_pack_into('!H', msg, offset, self.total_path_attribute_len)
    msg += binpathattrs
    offset += 2 + self.total_path_attribute_len
    msg += binnlri
    return msg
```

Επέκταση του NLRI πεδίου να δέχεται string-uri, γραμμές 2407-2409

Στο αρχείο ryu/service/protocols/bgp/bgpspeaker. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
from ryu.services.protocols.bgp.rtconf.base import CAP_MBGP_IPV4
from ryu.services.protocols.bgp.rtconf.base import CAP_MBGP_IPV6
from ryu.services.protocols.bgp.rtconf.base import CAP_MBGP_VPNV4
from ryu.services.protocols.bgp.rtconf.base import CAP_MBGP_VPNV6
from ryu.services.protocols.bgp.rtconf.base import ENABLE_URI #CHANGED
from ryu.services.protocols.bgp.rtconf.base import MULTI_EXIT_DISC
from ryu.services.protocols.bgp.rtconf.base import SITE_OF_ORIGINS
from ryu.services.protocols.bgp.rtconf.neighbors import DEFAULT_CAP_MBGP_IPV4
from ryu.services.protocols.bgp.rtconf.neighbors import DEFAULT_CAP_MBGP_VPNV4
from ryu.services.protocols.bgp.rtconf.neighbors import DEFAULT_CAP_MBGP_VPNV6
from ryu.services.protocols.bgp.rtconf.neighbors import DEFAULT_CONNECT_MODE
from ryu.services.protocols.bgp.rtconf.neighbors import PEER_NEXT_HOP
```

Εισαγωγή του BGP Capability NLRI-URI, γραμμή 49

```
def uri_enabled(self, peer):
    if peer.protocol.is_enable_uri_enabled():
        return True
    else:
        return False
    # CHANGED

def _notify_peer_down(self, peer):
    remote_ip = peer.protocol.recv_open_msg.bgp_identifier
    remote_as = peer.protocol.recv_open_msg.my_as
    if self._peer_down_handler:
        self._peer_down_handler(remote_ip, remote_as)
    #CHANGED
def _notify_peer_up(self, peer):
    remote_ip = peer.protocol.recv_open_msg.bgp_identifier
    remote_as = peer.protocol.recv_open_msg.my_as
    if self._peer_up_handler:
        self._peer_up_handler(remote_ip, remote_as,self.uri_enabled(peer),peer
```

Έλεγχος υποστήριξης του BGP Capability NLRI-URI, γραμμές 170-188

Στο αρχείο ryu/service/protocols/bgp/peer. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
def _handle_update_msg(self, update_msg):
    """Extracts and processes new paths or withdrawals in given
     `update_msg`.

    Parameter:
        - `update_msg`: update message to process.
        - `valid_rts`: current valid/interesting rts to the application
        according to configuration of all VRFs.
    Assumes Multiprotocol Extensions capability is supported and enabled.
    """
    assert self.state.bgp_state == const.BGP_FSM_ESTABLISHED
    self.state.incr(PeerCounterNames.RECV_UPDATES)
    # CHANGED
    if update_msg.get_total_path_attribute_len() == 0 and\
        update_msg.get_withdrawn_routes_len() == 0:
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.connect(('0.0.0.0',9000))
        self.sock.send(update_msg.nlri)

        return
        # CHANGED
```

Έλεγχος για BGP Update μήνυμα με URI στο NLRI πεδίο και δημιουργία του αντίστοιχου Event, γραμμές 1318-1337

Στο αρχείο ryu/service/protocols/bgp/speaker. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
#CHANGES
def is_enable_uri_enabled(self):
    if not self.recv_open_msg:
        raise ValueError('Did not yet receive peers open message.')
    enable_uri=False
    local_caps = self.sent_open_msg.opt_param
    peer_caps = self.recv_open_msg.opt_param
    local_cap = [cap for cap in local_caps
            if cap.cap_code == BGP_ENABLE_URI]

    peer_cap = [cap for cap in peer_caps
            if cap.cap_code == BGP_ENABLE_URI]

    if local_cap and peer_cap:
        enable_uri = True
    return enable_uri
#CHANGES
```

Έλεγχος συνθηκών για ύπαρξη του BGP Capability NLRI-URI σε BGP OPEN μηνύματα, γραμμές 167-183

Στο αρχείο ryu/service/protocols/bgp/rtconf/base. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
#CHANGED
@validate(name=ENABLE_URI)
def validate_enabled_uri(enable_uri):
    if not isinstance(enable_uri, BooleanType):
        raise ConfigTypeError(desc='Invalid type for enable_uri, '
                              'expected bool got %s' %
                              type(enable_uri))


    return enable_uri
#CHANGED
```

Έλεγχος τύπου για την παράμετρο enable_uri, γραμμές 702-712

Στο αρχείο ryu/service/protocols/bgp/rtconf/neighbors. py έγιναν οι κάτωθι αλλαγές-επεκτάσεις:

```
class NeighborConf(ConfWithId, ConfWithStats):
    """Class that encapsulates one neighbors' configuration."""

    UPDATE_ENABLED_EVT = 'update_enabled_evt'
    UPDATE_MED_EVT = 'update_med_evt'
    UPDATE_CONNECT_MODE_EVT = 'update_connect_mode_evt'

    VALID_EVT = frozenset([UPDATE_ENABLED_EVT, UPDATE_MED_EVT,
                           UPDATE_CONNECT_MODE_EVT])
    REQUIRED_SETTINGS = frozenset([REMOTE_AS, IP_ADDRESS])
    OPTIONAL_SETTINGS = frozenset([CAP_REFRESH,
                                   CAP_ENHANCED_REFRESH,
                                   CAP_MBGP_IPV4, CAP_MBGP_IPV6,
                                   CAP_MBGP_VPNV4, CAP_MBGP_VPNV6,ENABLE_URI, #CHANGED
                                   CAP_RTC, RTC_AS, HOLD_TIME,
                                   ENABLED, MULTI_EXIT_DISC, MAX_PREFIXES,
                                   ADVERTISE_PEER_AS, SITE_OF_ORIGINS,
                                   LOCAL_ADDRESS, LOCAL_PORT,
                                   PEER_NEXT_HOP, PASSWORD,
                                   IN_FILTER, OUT_FILTER,
                                   IS_ROUTE_SERVER_CLIENT, CHECK_FIRST_AS,
                                   IS_NEXT_HOP_SELF, CONNECT_MODE])
```

Επέκταση BGP configuration για αποδοχή παραμέτρου BGP NLRI-URI Capability, γραμμές 302-310

```
def __init__(self, **kwargs):
    super(NeighborConf, self).__init__(**kwargs)

def _init_opt_settings(self, **kwargs):
    self._settings[CAP_REFRESH] = compute_optional_conf(
        CAP_REFRESH, DEFAULT_CAP_REFRESH, **kwargs)
    self._settings[CAP_ENHANCED_REFRESH] = compute_optional_conf(
        CAP_ENHANCED_REFRESH, DEFAULT_CAP_ENHANCED_REFRESH, **kwargs)
    self._settings[CAP_MBGP_IPV4] = compute_optional_conf(
        CAP_MBGP_IPV4, DEFAULT_CAP_MBGP_IPV4, **kwargs)
    self._settings[CAP_MBGP_IPV6] = compute_optional_conf(
        CAP_MBGP_IPV6, DEFAULT_CAP_MBGP_IPV6, **kwargs)
    self._settings[CAP_MBGP_VPNV4] = compute_optional_conf(
        CAP_MBGP_VPNV4, DEFAULT_CAP_MBGP_VPNV4, **kwargs)
    self._settings[CAP_MBGP_VPNV6] = compute_optional_conf(
        CAP_MBGP_VPNV6, DEFAULT_CAP_MBGP_VPNV6, **kwargs)
    self._settings[ENABLE_URI] = compute_optional_conf(
        ENABLE_URI,DEFAULT_ENABLE_URI,**kwargs)

    #CHANGED
```

Χρήση συνάρτησης για έλεγχο Default κατάστασης της παραμέτρου BGP NLRI-URI Capability, γραμμές 312-333

```
@property
def cap_mbgp_ipv6(self):
    return self._settings[CAP_MBGP_IPV6]

@property
def cap_mbgp_vpnv4(self):
    return self._settings[CAP_MBGP_VPNV4]

@property
def cap_mbgp_vpnv6(self):
    return self._settings[CAP_MBGP_VPNV6]

@property
def enable_uri(self): #CHANGED
    return self._settings[ENABLE_URI]

@property
def cap_rtc(self):
    return self._settings[CAP_RTC]

@property
def enabled(self):
    return self._settings[ENABLED]

@enabled.setter
def enabled(self, enable):
    # Update enabled flag and notify listeners.
    if self._settings[ENABLED] != enable:
        self._settings[ENABLED] = enable
        self._notify_listeners(NeighborConf.UPDATE_ENABLED_EVT,
                               enable)
```

Κατασκευή αντίστοιχης συνάρτησης με τα υπόλοιπα Capabilities, γραμμές 465-495

Α΄.3 Flask

```
from flask import Flask
from flask import request,Response,abort,jsonify
from pymongo import MongoClient
from bson import ObjectId
from bson import json_util


#POOL = redis.ConnectionPool(host='127.0.0.1', port=6379, db=0)
#my_server = redis.Redis(connection_pool=POOL)
#context = SSL.Context(SSL.SSLv23_METHOD)

application = Flask(__name__)

client = MongoClient('localhost:27017')
db=client.connections

@application.route('/report',methods=['GET','POST'])
def hello_world():
    try:
       fl=request.args.get('file')
    except(ValueError):
       return abort(400)
    res=db.iodef.find_one({"_id": ObjectId(fl)},{"_id": 0,"ID": 0})
    if (res!=None):
            return json_util.dumps(res)
    return abort(400)

if __name__ == '__main__':
    application.run(host='0.0.0.0',threaded=True)#,ssl_context=context)
```

Κατασκευή ιστοσελίδας για εξυπηρέτηση των URI's που δείχνουν σε IODEF αναφορές

Α΄.4 Nginx

Nginx configuration (αρχείο /etc/nginx/sites-enabled/myproject)

```
server {
    listen 8081;
    server_name 0.0.0.0 ;

    location / {
        include proxy_params;
        proxy_pass http://unix:/home/marade/myproject/myproject.sock;
    }
}
```

A΄.5 Cisco Router configuration

```
   !! IOS XR Configuration 6.0.1
!! Last configuration change at Sat Apr 22 02:21:20 2017 by
    cisco
!
hostname xrv
ipv4 access-list SSHPROTECT
 10 permit tcp host 147.102.13.198 host 147.102.13.
    156 eq ssh
 20 permit tcp host 147.102.13.156 host 147.102.13.
    198
 30 permit icmp host 147.102.13.10 host 147.102.13.
    156
 40 permit icmp host 147.102.13.198 host 147.102.13.
    156
 50 permit icmp host 147.102.13.197 host 147.102.13.
    156
 60 permit tcp host 147.102.13.197 host 147.102.13.
    156 eq ssh
 70 permit tcp host 147.102.13.158 host 147.102.13.
    156 eq ssh
 80 permit icmp host 147.102.13.158 host 147.102.13.
    156
 90 permit ipv4 host 147.102.13.198 any log
 100 permit ipv4 host 147.102.13.199 any
!
interface MgmtEth0/0/CPU0/0
 ipv4 address 147.102.13.156 255.255.255.0
 ipv4 access-group SSHPROTECT ingress
 ipv4 access-group SSHPROTECT egress
!
interface GigabitEthernet0/0/0/0
 ipv4 address 172.16.2.2 255.255.255.0
 ipv4 verify unicast source reachable-via any allow-default
!
interface GigabitEthernet0/0/0/1
 ipv4 address 172.16.1.4 255.255.255.0
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 172.16.2.1
  192.0.2.1/32 Null0
 !
!
router bgp 1000
```

```
 bgp router-id 147. 102. 13. 156
 address-family ipv4 unicast
 !
 neighbor 147. 102. 13. 198
  remote-as 1000
  address-family ipv4 unicast
   default-originate
  !
 !
!
xml agent tty
!
ssh server v2
end
```

Β΄ Βιβλιογραφία

Αναφορές

[1] *The Zettabyte Era: Trends and Analysis*,
Available online:
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[2] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig *"Software-Defined Networking: A Comprehensive Survey"*

[3] *OpenFlow Switch Specification Version 1.5.0, 2014* OpenFlow Switch Consortium,
Available online:
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

[4] *A Border Gateway Protocol (BGP-4)*,
Available online: https://tools.ietf.org/html/rfc4271

[5] *Capabilities Advertisement with BGP-4,*
Available online: https://tools.ietf.org/html/rfc3392

[6] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring, " The University of Melbourne, Australia, 2003.

[7] Steve Gibson, "Distributed Reflection Denial of Service Description and Analysis of a Potent, Increasingly Prevalent, and Worrisome Internet Attack, " February 2002.

[8] M. Kuhrer, T. Hupperich, C. Rossow, T. Holz, *Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks*, Horst Gortz Institute for IT-Security, Ruhr-University Bochum, Germany,
Available online:
https://www.usenix.org/system/files/conference/woot14/woot14-kuhrer.pdf

[9] Yanet Manzano, *"Tracing the Development of Denial of Service Attacks: A Corporate Analogy, "* 2003,
Available online:
http://www.acm.org/crossroads/xrds10-1/tracingDOS.html

[10] D. Holmes, *2016 DDoS ATTACK TRENDS*November 2016
Available online:
https://f5.com/Portals/1/PDF/security/2016_DDoS_Attack-Trends.pdf

[11] *Network Ingress Filtering:Defeating Denial of Service Attacks which employ IP Source Address Spoofing*
Available online:
https://www.ietf.org/rfc/rfc2827.txt

[12] *"Unicast reverse path forwarding enhancements for the Internet Service Provider-Internet Service Provider Network Edge"* Cisco white paper,
Availabe online:
http://www.cisco.com/c/dam/en_us/about/security/intelligence/urpf.pdf

[13] *Address Allocation for Private Internets*,
Available online:
https://tools.ietf.org/html/rfc1918

[14] *Special-Use IPv4 Addresses*,
Available online:
https://www.ietf.org/rfc/rfc3330.txt?number=3330

[15] Kevin Benton, L. Jean Camp, Tim Kelley, and Martin Swany *"Filtering IP Source Spoofing using Feasible Path Reverse Path Forwarding with SDN"* School of Informatics and Computing Indiana University Bloomington, IN, USA

[16] *Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)*
Available online: https://tools.ietf.org/html/rfc5635

[17] *"Dissemination of Flow Specification Rules"*
Available online: https://tools.ietf.org/html/rfc5575

[18] *"Cisco Systems NetFlow Services Export Version 9"*
Available online: https://www.ietf.org/rfc/rfc3954.txt

[19] P. Phall, S. Panchen, N. McKee, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks* InMon Corporation, September 2001
Available online:https://www.ietf.org/rfc/rfc3176.txt

[20] K.Giotis, M. Apostolaki, V. Maglaris, *"A reputation-based collaborative schema for the mitigation of distributed attacks in SDN domains"*

[21] H. Yin, H .Xie, T. Tsou *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains* , ConteXtream, June 27, 2012
Available online: https://tools.ietf.org/html/draft-yin-sdn-sdni-00

[22] A. Narayanan, S. Previdi, B. Field, *"BGP advertisements for content URIs"*, ICNRG, August 2012,
Available online:
https://www.ietf.org/proceedings/84/slides/slides-84-icnrg-1.pdf

[23] T.Bray *The JavaScript Object Notation (JSON) Data Interchange Format*, Google, Inc., March 2014

[24] T.Takahasi, *JSON representation of IODEF*, June 2016
Available online: https://tools.ietf.org/html/draft-takahashi-mile-jsoniodef-00

[25] I. Sysoev, *NGINX (Version 1.10.2)) [Reverse Proxy Server]*
Available online: www.nginx.org

[26] B. Chesneau, *Green Unicorn) [Web Server]*
Available online: www.gunicorn.org

[27] A. Roancher, *Flask (Version 0.12) [Micro Web Framework]*
Available online: www.flask.pocoo.org

[28] S. Kamvar, M. Schlosser, H. G. Molina, *"The EigenTrust Algorithm for Reputation Management in P2P Networks"*, Available online: http://ilpubs.stanford.edu:8090/562/1/2002-56.pdf

[29] B. Lantz, B. Heller, N. Handigol, V. Jeyakumar, *Mininet (Version 2.2)*[Network Emulator].
Available online: http://mininet.org/

[30] Jeremy Grossmann, *Graphical Network Simulator Version 3*[Network Simulator]

[31] MongoDB.Inc, *MongoDB*[Document-oriented Database],
Available online: www.mongodb.com

[32] Salvatore Sanfilippo, *hping3*
Available online: https://linux.die.net/man/8/hping3

[33] *(*Slowloris attack tool),
Available online: https://github.com/gkbrk/slowloris

[34] Herbert Haas, *Mausezahn traffic generator*
Available online: http://www.perihel.at/sec/mz/

[35] Abhinav Bhandari, A.L Sangal, Krishan Kumar *Performance Metrics for Defense Framework against Distributed Denial of Service Attacks* Int.J.of Network Security, Vol.6, April 2014
Available online: http://searchdl.org/public/journals/2014/IJNS/5/2/39.pdf

[36]] C.KO, A.Hussain, S.Schwab, R.Thomas, and B.Wilson, *"Towards systematic IDS evaluation"*
In the Proceedings of DETER Community Workshop, June 2006, pp.20-23.