Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

ΣΤΗΝ ΑΣΦΑΛΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Βελτίωση Ασφάλειας Δικτύου Με SDN : Software Defined Networks

Παντελής Μανάκας

Επιβλέπουσα Καθηγήτρια

Αδαμαντίνη Περατικού

Μάιος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Βελτίωση Ασφάλειας Δικτύου Με SDN : Software Defined Networks

Παντελής Μανάκας

Επιβλέπουσα Καθηγήτρια

Αδαμαντίνη Περατικού

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε

προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών

στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών

του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2019

Περίληψη

Τα Δίκτυα Καθορισμένα από Λογισμικό είναι μια νέα, αναπτυσσόμενη τεχνολογία η οποία έρχεται να αλλάξει πλήρως την αρχιτεκτονική και την λειτουργία των παραδοσιακών δικτύων. Διαχωρίζοντας το πεδίο ελέγχου από το πεδίο δεδομένων, κεντρικοποιείται η διαχείριση και δημιουργείται ένα δυναμικό, ευέλικτο και πλήρως προγραμματοποιήσιμο δίκτυο. Υπάρχουν ωστόσο δικαιολογημένες ανησυχίες σχετικά με τους νέους κινδύνους ασφαλείας που γεννιούνται και τις απαραίτητες ενέργειες που θα πρέπει να γίνουν ώστε να διασφαλισθεί η ακεραιότητα και η αξιοπιστία τους.

Στην παρούσα διπλωματική εργασία γίνεται μια παρουσίαση των βασικών συστατικών και της αρχιτεκτονικής των Δικτύων που είναι Καθορισμένα από Λογισμικό, ενώ μπορούν να εντοπιστούν και οι βασικές τους διαφορές σε σχέση με τα παραδοσιακά δίκτυα. Εξετάζονται οι απειλές που εντοπίζονται σε κάθε του επίπεδο και προτείνονται οι αντίστοιχες λύσεις που θα μπορούσαν να υλοποιηθούν ως μέτρα προστασίας. Στην συνέχεια καταγράφονται τα σημεία στα οποία υπερέχουν τα Δίκτυα που Καθορίζονται από Λογισμικό στον τομέα της ασφάλειας, έναντι των παραδοσιακών δικτύων.

Τέλος γίνονται οι υλοποιήσεις προσομοίωσης ενός παραδοσιακού δικτύου και ενός Δικτύου Καθορισμένου από Λογισμικό, μέσω της πλατφόρμας του Mininet, παρουσιάζεται η συμπεριφορά τους ενώ δέχονται μια κακόβουλη επίθεση και καταγράφεται η αντίδραση τους προς αυτήν.

Summary

Software Defined Networking is a new developing technology which completely changes the architecture and working of traditional networks. By seperating the control plane from the data plane, management is being centralized, and is created a dynamic, flexible and fully programmable network. However, there are justifiable concerns about the new security issues arising as well as the necessary actions that must be taken so that the network's integrity and reliability can be ensured.

This thesis presents the basic components and the architecture of Software Defined Networks, while their main differences compared to the traditional networks can be detected. Focusing on security, we analyse most common threats for each of the network's layers and suggest the security measures that should be implemented. In addition, we record the outcompeting aspects of the Software Defined Network security opposed to the traditional networks.

Finally, we simulate a traditional network and a Software Defined Network through the Mininet platform. We present their behavior while they are under malicious attack and we record their reaction.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την κυρία Περατικού που μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο αντικείμενο και για την υποστήριξη της στην διπλωματική μου διατριβή. Επίσης θα ήθελα να ευχαριστήσω τους δικούς μου ανθρώπους για την στήριξη και την κατανόηση τους όλο αυτό το διάστημα.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Οι ανάγκες των σύγχρονων εταιριών έχουν αυξηθεί ραγδαία τα τελευταία χρόνια με αποτέλεσμα οι παραδοσιακές υποδομές να δυσκολεύονται να ακολουθήσουν αυτό τον ρυθμό και να αναζητούνται νέοι τρόποι και τεχνολογίες οι οποίες θα λειτουργούν πιο αποτελεσματικά και κυρίως χωρίς να αυξάνουν το κόστος. Αυτός είναι και ένας από τους λόγους για τους οποίους το SDN (Software Defined Networking) έχει αποκτήσει τέτοια προβολή και οι μεγαλύτεροι οργανισμοί στον τομέα της τεχνολογίας στρέφονται προς τα εκεί. Εφαρμόζοντας την νέα και συνεχώς αναπτυσσόμενη αυτή τεχνολογία, δίνεται η δυνατότητα στους διαχειριστές των συστημάτων εκμεταλλευόμενοι την αρχιτεκτονική της να διαχειρίζονται, να συντηρούν και να αναβαθμίζουν τις υπηρεσίες των δικτύων τους με έναν κεντρικοποιημένο τρόπο όπου με μια κεντρική οντότητα , τον ελεγκτή έχουν τον πλήρη έλεγχο. Με την αρχιτεκτονική αυτή αλλάζει ουσιαστικά η οπτική με την οποία βλέπαμε ως τώρα το παραδοσιακό δίκτυο ενός οργανισμού και οδηγούμαστε σε μια νέα εποχή όπου μπορούμε να διαχειριζόμαστε τις υπηρεσίες και να εφαρμόζουμε πολιτικές χωρίς περιοριζόμαστε από τις ελλείψεις, τις ασυμβατότητες και την σύνθετη λειτουργία των υποδομών.

Ωστόσο, η ανάπτυξη αυτής της αρχιτεκτονικής γίνεται τόσο γρήγορα και είναι έντονες οι ανησυχίες σχετικά με το αν οι επίδοξοι εισβολείς θα καταφέρουν να δημιουργήσουν προβλήματα. Είναι δεδομένο ότι το πλήθος των οργανισμών που ενδιαφέρονται να υλοποιήσουν SDN στο δίκτυο τους θα αποτελεί πρόκληση για κακόβουλες παρεμβάσεις. Μάλιστα η ίδια η αρχιτεκτονική του SDN στην οποία εύκολα εντοπίζεται ο εύκολος "στόχος", ο ελεγκτής από όπου γίνεται όλη η διαχείριση, μπορεί να οδηγήσει σε επιθέσεις τύπου Denial of Service (DoS) με αποτέλεσμα να παραλύσουν όλες οι λειτουργίες ή και Man in the Middle όπου κάποιος μπορεί να πάρει τον έλεγχο ή να παρακολουθεί την κάθε κίνηση του δικτύου.
Έτσι, είναι ιδιαίτερα κρίσιμο όσο δίνεται έμφαση στην καινοτομία για την υλοποίηση νέων υπηρεσιών και την αύξηση της αποδοτικότητας, να αναπτύσσεται και ο τομέας της ασφάλειας ως ξεχωριστό πεδίο ανάπτυξης, στόχος του οποίου θα είναι να ξεπεράσει όλες τις αδυναμίες των παραδοσιακών δικτύων και να βελτιώσει την αξιοπιστία και την ακεραιότητα τους.
Σκοπός της διατριβής είναι η παρουσίαση των βασικών αρχών λειτουργίας ενός Software Defined Network, ο εντοπισμός των διαφορών του σε σχέση με ένα παραδοσιακό Ethernet δίκτυο και η κάλυψη που παρέχει αυτή η καινοτόμα τεχνολογία δικτύωσης ως προς τον τομέα της ασφάλειας. Ο κυρίως στόχος είναι να αναγνωριστεί αν το SDN έχει τις δυνατότητες να προσφέρει μια εναλλακτική λύση με επιπρόσθετη ασφάλεια.
Τα βασικά ερευνητικά ερωτήματα της διατριβής είναι:

- Ποιοι μηχανισμοί έχουν αναπτυχθεί για την παροχή ασφάλειας στους χρήστες ενός Software Defined Network;
- Είναι αρκετά αποτελεσματικοί για να αποτρέψουν τους διάφορους τρόπους επιθέσεων;
- Ποιες είναι οι προκλήσεις ασφάλειας που πρέπει να αντιμετωπιστούν με την χρήση του SDN;
- Συγκριτικά με ένα παραδοσιακό Ethernet δίκτυο ποια από τις δυο τεχνολογίες υπερέχει στον τομέα της ασφάλειας;

Η αναγκαιότητα και σπουδαιότητα της έρευνας έγκειται στο ότι η νέα τεχνολογία των Software Defined Networks αναπτύσσεται συνεχώς και ερευνάται από τους μεγαλύτερους οργανισμούς πληροφοριακών συστημάτων. Η χρήση της τα επόμενα χρόνια αναμένεται να αυξηθεί και μαζί με αυτήν θα αυξηθούν και οι επίδοξοι εισβολείς. Είναι απαραίτητο να εντοπιστούν οι ευπάθειες και να προταθούν τρόποι κάλυψης τους και αντιμετώπισης των επιθέσεων που γίνονται από τους εισβολείς. Είναι πολύ σημαντικό λοιπόν να γίνει μια

καταγραφή της υπάρχουσας κάλυψης που παρέχουν τα SDN στον τομέα της ασφάλειας αλλά και να γίνουν προτάσεις για την βελτίωση της.

Όσον αφορά την μεθοδολογία που ακολουθήθηκε, έγινε βιβλιογραφική επισκόπηση για την καταγραφή των σημαντικότερων πηγών οι οποίες παρείχαν τις πληροφορίες που συγκεντρώθηκαν και παρουσιάζονται στην παρούσα διατριβή. Σχεδιάστηκαν τα μοντέλα, ενός παραδοσιακού και ενός SDN δικτύου και υλοποιήθηκε προσομοίωση κακόβουλης επίθεσης σε αυτά. Συγκρίνεται η συμπεριφορά του καθενός στην επίθεση που δέχεται και παρουσιάζονται τα αντίστοιχα συμπεράσματα.

Στο κεφάλαιο 2 παρουσιάζονται βασικά στοιχεία της αρχιτεκτονικής των Software Defined Networks και γίνεται μια βιβλιογραφική ανασκόπηση ως προς τους υπάρχοντες μηχανισμούς ασφάλειας σε αυτά. Στο κεφάλαιο 3 γίνεται μια καταγραφή των σημαντικότερων ευπαθειών και προκλήσεων που εντοπίζονται στον τομέα της ασφάλειας. Στο κεφάλαιο 4 παρουσιάζονται οι σημαντικότεροι τομείς που δίνουν στα Software Defined Networks προβάδισμα ως προς την ασφάλεια και την υλοποίηση πολιτικών σε αυτή την κατεύθυνση. Στο κεφάλαιο 5 παρουσιάζεται ο σχεδιασμός των δυο μοντέλων, τα εργαλεία που χρησιμοποιήθηκαν και η υλοποίηση της προσομοίωσης.

Κεφάλαιο 2
Software Defined Networking

2.1 Παραδοσιακά Δίκτυα

Παραδοσιακά στις Ethernet συσκευές δικτύωσης τα πεδία δεδομένων και ελέγχου είναι συνδεδεμένα μεταξύ τους. Αυτό σημαίνει ότι το λειτουργικό σύστημα και τα χαρακτηριστικά του, μαζί με το υλικό (hardware) είναι υλοποιημένα σε μια μοναδική συσκευή. Τα συστατικά των πεδίων δεδομένων ελέγχου και διαχείρισης περιγράφονται από ένα λογικό, θεωρητικό μοντέλο διαστρωμάτωσης το οποίο είναι γνωστό ως Μοντέλο Ενδοσύνδεσης Ανοιχτών Συστημάτων (OSI Model) και έχει προτυποποιηθεί από τον Διεθνή Οργανισμό Προτυποποίησης (International Organization of Standardization). Δίνεται μέσω αυτού η δυνατότητα στους προγραμματιστές να αναπτύσσουν και να βελτιώνουν λειτουργίες σε κάποιο στρώμα χωρίς να λαμβάνονται υπόψη τα υπόλοιπα επίπεδα.

Όπως γνωρίζουμε, η κίνηση στο Ethernet βασίζεται στα Ethernet πλαίσια, στα οποία ενθυλακώνονται τα δεδομένων. Η πηγή και ο προορισμός περιέχονται στις κεφαλίδες του Ethernet οι οποίες πακετάρονται μαζί με τα πλαίσια. Η δημιουργία των πλαισίων πραγματοποιείται από τους υπολογιστές, ενώ οι μεταγωγείς επιθεωρούν τις κεφαλίδες, και διαχειρίζονται ανάλογα τα πλαίσια.

Ένας Ethernet μεταγωγέας έχει πολλαπλές διεπαφές (θύρες) για εσωτερική μεταγωγή. Η βασική του λειτουργία είναι να προωθεί πλαίσια από μία θύρα σε μια άλλη, λαμβάνοντας υπόψη τον προορισμό. Για αυτό το λόγο, ο μεταγωγέας διατηρεί καταγραφή όλων των MAC διευθύνσεων, που συνδέονται σε κάθε θύρα. Αυτός ο πίνακας ονομάζεται Διευθυνσιοδοτημένη Μνήμη Περιεχομένου (CAM). Αρχικά, ο μεταγωγέας δε γνωρίζει τις διευθύνσεις MAC όλων των συνδεδεμένων υπολογιστών. Αυτή η πληροφορία συλλέγεται όταν οι υπολογιστές στέλνουν πλαίσια και έτσι ο μεταγωγέας μαθαίνει τη διεύθυνση MAC της πηγής, τη στιγμή που πραγματοποιεί την προώθηση του πλαισίου. Οι MAC διευθύνσεις των συνδεδεμένων διεπαφών προστίθενται, αν δεν έχουν ήδη προστεθεί, και χρησιμοποιώντας τον ίδιο πίνακα, ο μεταγωγέας γνωρίζει σε ποια θύρα πρέπει να προωθήσει το πλαίσιο. Σε περίπτωση που δεν υπάρχει καμία αντιστοιχία, ο μεταγωγέας στέλνει το πλαίσιο σε όλες τις

θύρες του, εκτός από αυτήν από την οποία προήλθε. Βάσει των αρχών τους Ethernet, κάθε υπολογιστής θα πρέπει να δέχεται μόνο πλαίσια τα οποία προορίζονται για τον ίδιο και να απορρίπτει τα υπόλοιπα.

Όσον αφορά την δρομολόγηση των δεδομένων μεταξύ διαφορετικών δικτύων χρησιμοποιούνται οι δρομολογητές και τα παραπάνω Ethernet πλαίσια ενθυλακώνονται σε πακέτα. Πέρα από τις στατικές διαδρομές που καθορίζονται από τον διαχειριστή του συστήματος, υπάρχουν δυο τύποι πρωτοκόλλων δρομολόγησης, τα distance vector και τα link state πρωτόκολλα.

- Distance Vector

  Κάθε δρομολογητής προωθεί μια λίστα των γνωστών του προορισμών στους γειτονικούς του δρομολογητές και καθένας από αυτούς ενημερώνει τους εσωτερικούς του πίνακες δρομολόγησης σύμφωνα με τις νέες πληροφορίες. Η σύγκλιση των πινάκων δρομολόγησης ωστόσο μπορεί να είναι χρονοβόρα ειδικά στις περιπτώσεις αφαίρεσης-απώλειας κάποιου δρομολογητή από το δίκτυο. Στα distance vector πρωτόκολλα λογίζονται τα RIP και BGP
- Link state

  Κάθε δρομολογητής μοιράζεται την πληροφορία για τους γείτονες του με όλους τους δρομολογητές μέσω διαδικασίας flooding, αποστέλλοντας δηλαδή σε όλους τον πίνακα δρομολόγησης του ή κάθε αλλαγή που γίνεται σε αυτόν. Η πρακτική αυτή καταναλώνει φυσικά περισσότερη από την χωρητικότητα του δικτύου, ωστόσο η σύγκλιση των πινάκων δρομολόγησης γίνεται ταχύτερα και έχουν όλοι πλήρη εικόνα του δικτύου. Στα link state πρωτόκολλα εντάσσονται τα OSPF και ISIS.

Αναλύοντας την αρχή λειτουργίας των παραδοσιακών δικτύων ως προς τα τρία βασικά επίπεδα λειτουργίας έχουμε τα εξής:

- Επίπεδο Δεδομένων (Data Plane): Αφορά όλες τις λειτουργίες και τις διαδικασίες που είναι υπεύθυνες για την προώθηση και εναλλαγή των πακέτων από την μια διεπαφή στην άλλη. (QoS , access control lists (ACLs).)

- Επίπεδο Ελέγχου (Control Plane): Αφορά όλες τις λειτουργίες και τις διαδικασίες που καθορίζουν ποια διαδρομή θα χρησιμοποιηθεί. Στο επίπεδο αυτό επιτελούνται οι λειτουργίες των πρωτοκόλλων δρομολόγησης καθώς και λειτουργίες ανταλλαγής

πληροφοριών που απαιτεί η λειτουργία των πρωτοκόλλων στις διαδικτυακές συσκευές .

- Επίπεδο Διαχείρισης (Management Plane): Αφορά όλες τις λειτουργίες και διαδικασίες που απαιτούνται για την παρακολούθηση των δικτυακών συσκευών (Simple Network Management Protocol (SNMP), Telnet, File Transfer Protocol (FTP), Secure FTP, and Secure Shell (SSH)).

Εικόνα 1: Παραδοσιακή προσέγγιση λειτουργίας των τριών επιπέδων σε ένα δίκτυο.

Η χρήση των παραδοσιακών δικτύων όσο διαδεδομένη και αξιόπιστη κι αν είναι σήμερα, έχει κάποια χαρακτηριστικά τα οποία δεν επιτρέπουν στην αρχιτεκτονική αυτή να καλύψει πλήρως τις απαιτήσεις των χρηστών και των επιχειρήσεων.

Η υποχρεωτικά χειροκίνητη ρύθμιση κάθε μιας εκ των συσκευών του δικτύου είτε αυτές είναι υπολογιστές, μεταγωγείς, δρομολογητές ή firewall κάνει την προσέγγιση αυτή επιρρεπή σε σφάλματα και φυσικά χρονοβόρα.
Ειδικά σε πολύ μεγάλα δίκτυα, η εφαρμογή μιας ευρείας πολιτικής ασφάλειας ή QoS έχει αυξημένη πολυπλοκότητα αν αναλογιστούμε το πλήθος των συσκευών, τις σύνθετες λειτουργίες τους και τα εκατομμύρια γραμμών κώδικα που περιέχουν. Σε πολλές περιπτώσεις είναι αναγκαία η χρήση γραμμής εντολών για την παραμετροποίηση μιας δικτυακής συσκευής καθώς κάποιες δεν έχουν γραφικό περιβάλλον, ενώ και σε αυτές που υπάρχει είναι προφανές ότι δεν είναι δυνατόν να ενταχθούν όλες οι λειτουργίες του εξοπλισμού σε αυτό.

Οδηγούμαστε έτσι σε αποφάσεις αποφυγής του κινδύνου ώστε να μην διαταραχθούν οι υπηρεσίες που ήδη λειτουργούν αξιόπιστα. Η αδράνεια όμως αυτή καταλήγει σε στατικότητα και καθιστά τους οργανισμούς ευάλωτους σε κινδύνους ασφαλείας, και λιγότερο αποδοτικούς. Είναι απαραίτητο για ένα σύγχρονο δίκτυο να έχει την ευχέρεια να επεκτείνεται προς την κατεύθυνση που το απαιτούν οι ανάγκες των χρηστών του για κάθε

τύπο υπηρεσίας. Πέρα από αυτό όμως η εξάρτηση των δικτύων από τους εξοπλισμούς επιφέρει εξάρτηση και από τους κατασκευαστές τους, οι οποίοι κρατάνε το λογισμικό τους κλειστό και πατενταρισμένο με ότι φυσικά συνεπάγεται αυτή η εξάρτηση σε ζητήματα συμβατότητας αλλά και κόστους.

2.2 Η Αρχιτεκτονική SDN και τα βασικά της συστατικά

2.2.1 Ορισμός

Η ανάγκη να αντιμετωπιστεί η στατική προσέγγιση των παραδοσιακών δικτύων που αναφέρθηκε παραπάνω οδήγησε στην γέννηση των Software Defined Networks. Αν και τα πρώτα βήματα σε θεωρητικό επίπεδο ξεκίνησαν από το Internet Engineering Task Force (IETF) το 2004 και γίνονταν προσπάθειες υλοποίησης και από άλλους οργανισμούς αλλά και από την ακαδημαϊκή κοινότητα με την δημιουργία και ανάπτυξη των πρώτων APIs και πρωτοκόλλων, το SDN θεωρούμε ότι ξεκίνησε το 2011 με την ίδρυση του Open Networking Foundation (ONF).
Ο ορισμός [9] λοιπόν που δίνεται από τον ONF είναι ο εξής:
Το SDN είναι μια αναδυόμενη τεχνολογία η οποία είναι δυναμική, διαχειρίσιμη, αποδοτική οικονομικά και ευπροσάρμοστη. Τα χαρακτηριστικά αυτά την κάνουν ιδανική για την δυναμική φύση των σύγχρονων εφαρμογών με τις υψηλές απαιτήσεις σε bandwidth. Αυτή η αρχιτεκτονική αποσυνδέει τον έλεγχο του δικτύου και τις μεθόδους προώθησης από το δίκτυο δίνοντας τους την δυνατότητα να γίνουν απολύτως προγραμματίσιμα και την υποβόσκουσα υποδομή να γίνει πιο αφηρημένη προς τις εφαρμογές και τις υπηρεσίες του δικτύου.

Ο κύριος παράγοντας που κάνει το SDN να ξεχωρίζει είναι ο διαχωρισμός του data plane από το control plane στους δρομολογητές και τους μεταγωγείς. Η υλοποίηση του control plane γίνεται μέσω λογισμικού και διαχωρίζεται από τους δικτυακούς εξοπλισμούς, ενώ το data plane υλοποιείται εντός του δικτυακού εξοπλισμού.

Το βασικό και μοναδικό χαρακτηριστικό που κάνει το SDN ιδιαίτερα δημοφιλές είναι η ικανότητα του να καθιστά το δίκτυο προγραμματίσιμο. Έχει τις προοπτικές να φέρει επανάσταση στην βιομηχανία των δικτύων. Η εποχή πλέον έχει ανάγκη από ευελιξία και δυναμική διαχείριση των στοιχείων αποσφαλμάτωσης.

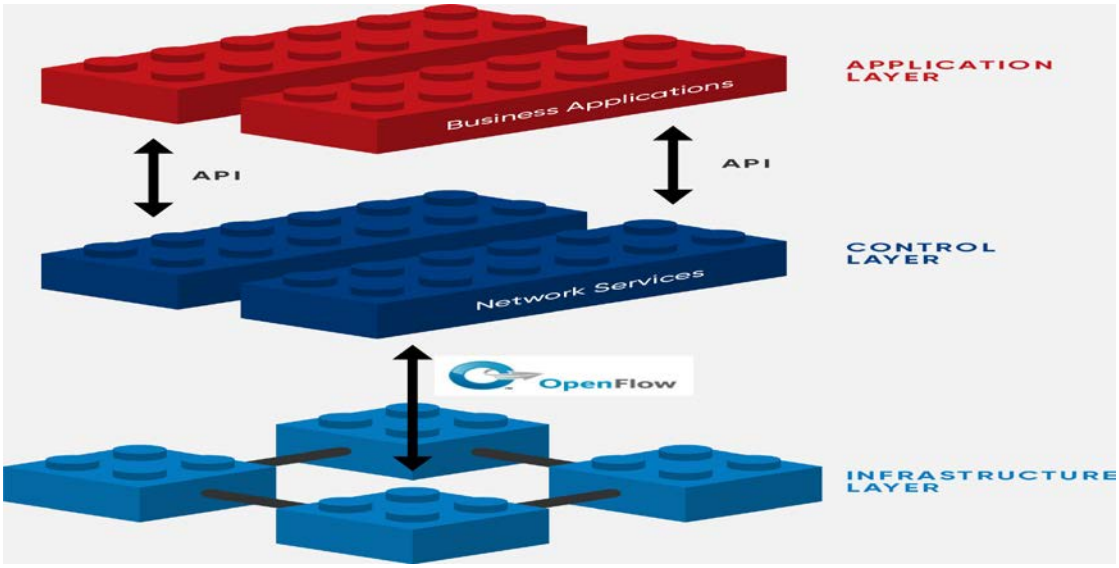
2.2.2 Η αρχιτεκτονική των Software Defined Networks

Η αρχιτεκτονική S.D.N. χαρακτηρίζεται από 3 κύρια επίπεδα, (Εικόνα 2):

- Επίπεδο Εφαρμογών (Application Layer): Σε αυτό το επίπεδο εντάσσονται όλες οι εφαρμογές οι οποίες καθορίζουν τα χαρακτηριστικά του δικτύου, τις πολιτικές που θα τηρούνται σε αυτό και οι υπηρεσίες που θα υλοποιούνται και θα παρέχονται στους χρήστες του συστήματος. Μέσω των APIs διακινούνται οι πληροφορίες που προέρχονται από τα υπόλοιπα επίπεδα και ανάλογα λαμβάνονται οι αποφάσεις και δίνεται η δυνατότητα να αλλάξει δυναμικά η συμπεριφορά του δικτύου.

- Επίπεδο Ελέγχου (Control Layer): Στο επίπεδο αυτό βρίσκεται το πιο σημαντικό στοιχείο της αρχιτεκτονικής, ο ελεγκτής (controller). Ο ελεγκτής λοιπόν έχει την πλήρη εικόνα για την τοπολογία του δικτύου ανά πάσα στιγμή, καθώς επικοινωνεί με όλες τις συσκευές δικτύωσης της υποδομής, και ενημερώνεται για κάθε αλλαγή που γίνεται σε αυτό. Από τις οδηγίες που λαμβάνει από τις εφαρμογές του προηγούμενου επιπέδου που αναφέραμε, μεταβιβάζει τις πληροφορίες εκτέλεσης των αντίστοιχων ενεργειών στους κατάλληλους δικτυακούς εξοπλισμούς. Εδώ ουσιαστικά είναι το σημείο στο οποίο λαμβάνονται όλες οι αποφάσεις για τον τρόπο που θα γίνει η διαχείριση των πακέτων, είτε αν αυτά θα προωθηθούν και σε ποιο μονοπάτι είτε αν θα απορριφθούν.

- Επίπεδο Υποδομής (Infastructure Layer): Αποτελείται από όλο το υλικό, τις συσκευές δικτύωσης που αποτελούν το υποκείμενο φυσικό δίκτυο, και υλοποιείται η φυσική διασύνδεση του. Στις συσκευές υλικού εκτελείται λογισμικό, το οποίο παρέχει μια διεπαφή ελέγχου του επιπέδου δεδομένων (Southbound API). Αυτή η διεπαφή χρησιμοποιείται για την επικοινωνία με το ανώτερο επίπεδο, δηλαδή το επίπεδο ελέγχου.



Εικόνα 2: Απεικόνιση της SDN Αρχιτεκτονικής (ONF)

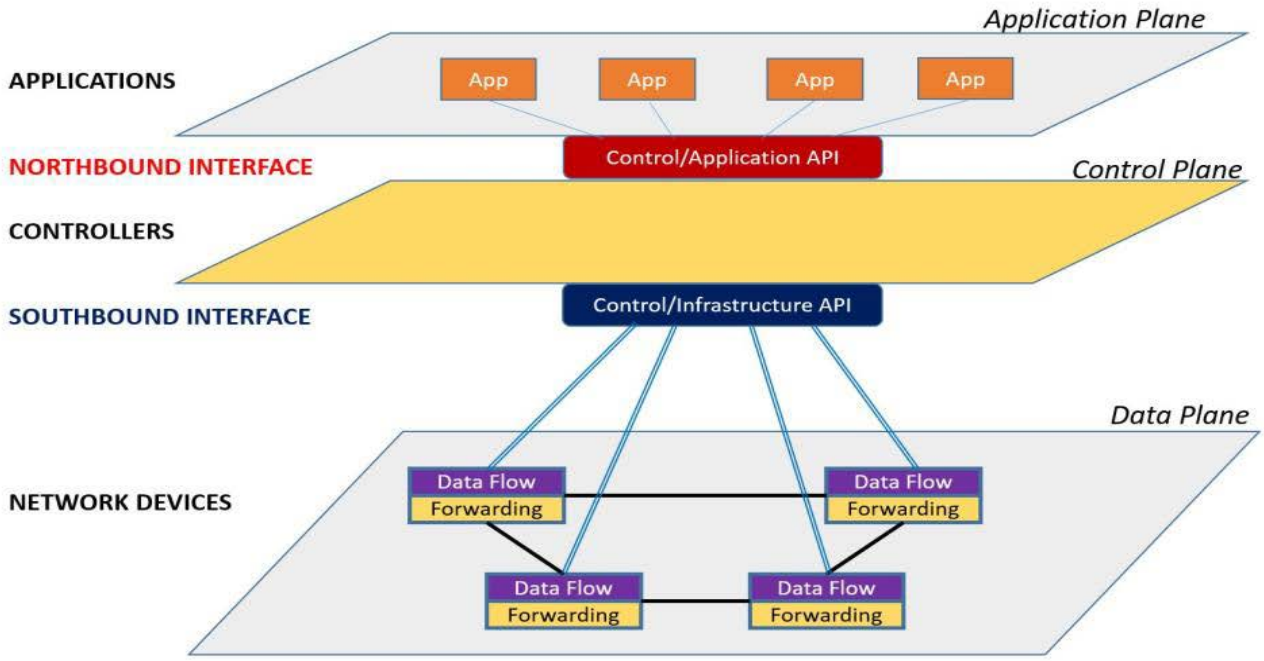
Τα προαναφερθέντα επίπεδα διασυνδέονται μεταξύ τους μέσω διεπαφών, είτε βρίσκονται στην ίδια συσκευή είτε όχι. Ανάλογα την περίπτωση είτε θα χρησιμοποιηθεί μια κλήση συστήματος ώστε να τα φέρει σε επικοινωνία υπό την ίδια συσκευή ή τον ρόλο θα αναλάβει ένα πρωτόκολλο επικοινωνίας ανάμεσα σε δυο διαφορετικές δικτυακές συσκευές. Οι διεπαφές (Εικόνα 3) που χρησιμοποιούνται για την αλληλεπίδραση στοιχείων των χωρίζονται σε Northbound και Southbound.

Μεταξύ αυτών των τριών επιπέδων υπάρχουν οι Διεπαφές Προγραμματισμού Εφαρμογών (APIs), οι οποίες παρέχουν τα απαραίτητα εργαλεία επικοινωνίας:

- Το Northbound API παρέχεται από τον ελεγκτή και οι εφαρμογές πρέπει να διαχειριστούν την επικοινωνία τους μαζί του, μέσω αυτής της διεπαφής. Δημιουργούν την διασύνδεση του πεδίου ελέγχου με το πεδίο εφαρμογών.

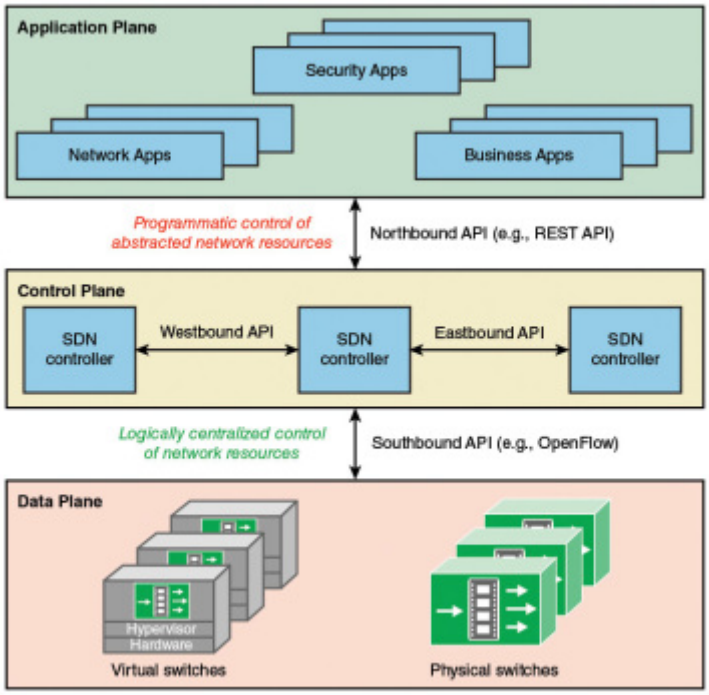
  Αν και υπάρχουν αρκετές γλώσσες προγραμματισμού οι οποίες χρησιμοποιούνται για την επικοινωνία με τον ελεγκτή μέσω αυτής της διεπαφής δεν έχει ξεχωρίσει κάποια ώστε να καθιερωθεί ως πρωτόκολλο.

- Το Southbound API είναι ο τρόπος επικοινωνίας μεταξύ του ελεγκτή και των δικτυακών συσκευών. Αντίστοιχα πραγματοποιούν την διασύνδεση μεταξύ του επιπέδου ελέγχου με το επίπεδο υποδομών. Το πιο διαδεδομένο από αυτά είναι το πρωτόκολλο OpenFlow, το οποίο θα αναλυθεί περισσότερο και παρακάτω.



Εικόνα 3 : Τα επίπεδα και οι διεπαφές της SDN αρχιτεκτονικής

Σε κατανεμημένες SDN αρχιτεκτονικές (Εικόνα 4) εμφανίζονται και τα Eastbound και Westbound APIs, τα οποία παρέχουν την διεπαφή μεταξύ των ελεγκτών, ορισμένες δυνατότητες παρακολούθησης και ενημέρωσης και την υλοποίηση αλγορίθμων συνοχής των δεδομένων.



Εικόνα 4: Κατανεμημένη SDN αρχιτεκτονική

Τα διαφορετικά στιγμιότυπα ελεγκτή, σε μια κατανεμημένη αρχιτεκτονική πρέπει να επικοινωνούν συχνά και να μεταφέρουν, μεταξύ τους, πληροφορίες ελέγχου και διαχείρισης.

Υπάρχουν πολλές διαφορετικές υλοποιήσεις σε κατανεμημένες SDN αρχιτεκτονικές που μπορεί να συναντήσουμε.
Σε μια από αυτές υπάρχουν διαφορετικά στιγμιότυπα ελεγκτών για κάποιες από τις σημαντικότερες λειτουργίες, ανάλογα με την κρίση των διαχειριστών του δικτύου, όπου για παράδειγμα σε ένα από αυτά μπορεί να γίνεται διαχείριση των ζητημάτων ασφάλειας, στο άλλο η βελτιστοποίηση διαχείρισης των πόρων, η εξισορρόπηση του φορτίου κατά βούληση κ.ο.κ. Επιπλέον, οι ελεγκτές μπορούν να διαμοιράζονται κάποιες από τις εργασίες τους ως κοινές και ο φόρτος να κατανέμεται σε νέα στιγμιότυπα ελεγκτών, σε πραγματικό χρόνο, με βάση τις ανάγκες σε πόρους, την κατανάλωση κτλ.

2.2.3 Βασικός τρόπος λειτουργίας

Από τα παραδοσιακά δίκτυα Ethernet, έχουμε συνηθίσει να διαχωρίζουμε τον δικτυακό εξοπλισμό σε επιπέδου 3 (δρομολογητές), πακέτα(packets), πίνακες δρομολόγησης ή επιπέδου 2 (μεταγωγείς), πλαίσια (frames) κτλ. Στην SDN τεχνολογία χρησιμοποιούνται οι όροι προώθησης(forwarding) και ροής(flow) καθώς δημιουργούνται πίνακες ροής (flow tables) βάση των οποίων το επίπεδο δεδομένων θα προωθήσει τα πακέτα. Αυτές οι ροές περιέχουν αντίστοιχα πεδία όπως MAC διευθύνσεις, IP διευθύνσεις, VLAN tags κ.ο.κ.

Όταν ένας μεταγωγέας ενός SDN δικτύου δεχτεί το πρώτο πακέτο μιας ροής ενός αποστολέα, γίνεται αναζήτηση στην κρυφή μνήμη (cache) του SDN μήπως υπάρχει ήδη κανόνας ροής για το πακέτο.

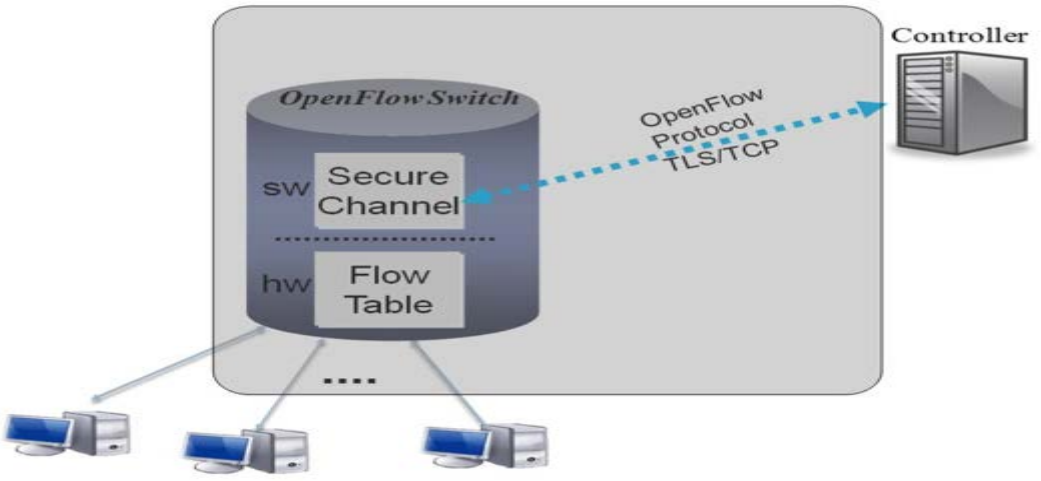
- Αν βρεθεί εγγραφή που να αντιστοιχεί, τότε εκτελούνται οι οδηγίες που σχετίζονται με την συγκεκριμένη εγγραφή ροής (π.χ. ανανέωση του μετρητή, σύνολο ενεργειών, μεταδεδομένα κτλ) και στην συνέχεια, προωθούνται και τα υπόλοιπα πακέτα στον παραλήπτη.
- Αν δεν βρεθεί κάποια καταχώρηση στον πίνακα ροών, τότε το πακέτο προωθείται στον ελεγκτή, μέσω ενός ασφαλούς καναλιού. Με την χρήση ενός Southbound API ο ελεγκτής μπορεί να προσθέσει, να ανανεώσει, ή να διαγράψει εγγραφές ροών, τόσο ως απόκριση στην ροή των πακέτων όσο και προληπτικά. Ο ελεγκτής εκτελεί τον αλγόριθμο δρομολόγησης και προσθέτει μια νέα εγγραφή προώθησης στον πίνακα

ροών του μεταγωγέα και όλων των σχετικών μεταγωγέων που ανήκουν στο μονοπάτι της ροής. Έπειτα ο μεταγωγέας προωθεί το πακέτο στην κατάλληλη θύρα για να αποσταλεί στον τελικό παραλήπτη.

2.2.4 Πρωτόκολλο Openflow

Το πρωτόκολλο OpenFlow χρησιμοποιείται για να επικοινωνούν ο ελεγκτής από το επίπεδο ελέγχου με τα στοιχεία του δικτύου στο επίπεδο υποδομών. Πρόκειται για μια από τις πιο σημαντικές τεχνολογίες της αρχιτεκτονικής SDN, σε σημείο που πολλοί λανθασμένα συγχέουν τις δύο έννοιες ως μια ενιαία. Πρόκειται για ένα πολύ καλά καθορισμένο API από το Open Networking Foundation και συγκεκριμένα το πιο διαδεδομένο Southbound API, το οποίο είναι υπεύθυνο για τον απομακρυσμένο έλεγχο του πίνακα προώθησης ενός μεταγωγέα ή ενός δρομολογητή.

Μια δικτυακή συσκευή που υποστηρίζει το πρωτόκολλο OpenFlow, είτε πρόκειται για μεταγωγέα είτε για δρομολογητή αποτελείται από έναν ή περισσότερους πίνακες ροής, έναν πίνακα ομάδας και ένα ασφαλές κανάλι επικοινωνίας μέσω του οποίου γίνεται η επικοινωνία με τον ελεγκτή. Ο ελεγκτής OpenFlow χρησιμοποιώντας το κανάλι αυτό διαχειρίζεται τον μεταγωγέα OpenFlow μέσω του πρωτοκόλλου OpenFlow. Το ασφαλές κανάλι κρυπτογραφείται μέσω του πρωτοκόλλου ασφαλείας TLS, ενώ μπορεί να λειτουργήσει και μέσω του πρωτοκόλλου TCP. Η απεικόνιση της επικοινωνίας αυτής παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 5: Openflow πρωτόκολλο

Με την χρήση του OpenFlow, μια δικτυακή συσκευή, θα διατηρεί τον πίνακα ροών της ο οποίος θα περιέχει όλες τις αντίστοιχες εγγραφές [17].

Match fields	Priority	Counters	Instructions	Timeouts	Cookies

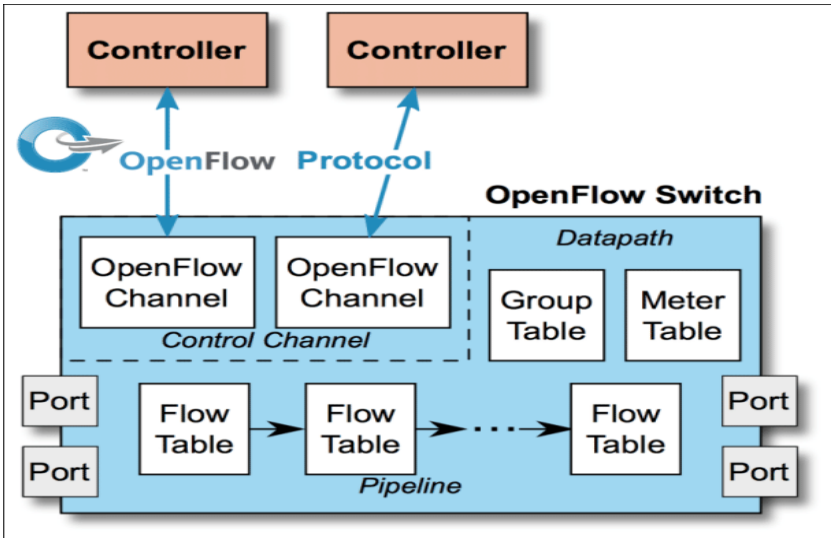
Πίνακας 1: Κύρια στοιχεία μιας εγγραφής ροής

- Match field (Αντιστοίχιση): Χρησιμοποιείται για την επιλογή πακέτων που ταιριάζουν με τις τιμές στα πεδία. (Tcp port , udp source ,vlan id κτλ). Κατά την είσοδο δηλαδή της ροής σε έναν μεταγωγέα, τα πακέτα ένα ένα ελέγχονται από κάθε πίνακα ροής βάση του match field και όσων τα πεδία αντιστοιχίζονται με κάποια από τις εγγραφές, εκτελεστούν οι ενέργειες που ορίζονται.
- Priority (Προτεραιότητα): Σχετική προτεραιότητα των καταχωρήσεων του πίνακα. Δίνεται για την περίπτωση όπου κάποιο πακέτο ροής αντιστοιχίζεται σε παραπάνω από έναν κανόνες, ώστε να εκτελεστούν οι ενέργειες με την σωστή σειρά.
- Counters (Μετρητές): Μετρητές γενικής και ειδικής χρήσης όπως ορίζει το whitepaper των προδιαγραφών της κάθε έκδοσης OpenFlow που χρησιμοποιείται, όπως πχ μετρητές ληφθέντων bytes, πακέτων ανά θύρα, διάρκεια ροής κτλ.
- Instructions (Οδηγίες): Ενέργειες που πρέπει να εκτελούνται σε περίπτωση εμφάνισης επιτυχούς αντιστοίχησης του ορισμένου πεδίου.
- Timeouts (Χρονικά όρια): Μέγιστος χρόνος αναμονής πριν από τη λήξη μιας ροής από το Switch.
- Cookie : Τιμές δεδομένων επιλεγμένες από τον ελεγκτή. Δεν χρησιμοποιούνται κατά την επεξεργασία του πακέτου. Στην ουσία αποτελεί τον τρόπο προσδιορισμού, στοιχείο μηχανισμού του OpenFlow, των ροών για τις εργασίες τροποποίησης και διαγραφής. Χρησιμοποιείται ως εργαλείο συλλογής και παραγωγής στατιστικών αλλά και ως στοιχείο διαδικασίας φιλτραρίσματος ροών.

Σε περίπτωση όπου κατά την διαδικασία αντιστοίχησης, ένα πακέτο που εισάγεται σε έναν μεταγωγέα, ενώ περνάει από όλους τους πίνακες ροής διαδοχικά (pipeline), ελέγχεται το match field του και δεν εντοπιστεί κάποια καταχώρηση σε έστω έναν από αυτούς, τότε έχουμε την διαδικασία table miss. Αναφέραμε και στον γενικό τρόπο λειτουργίας παραπάνω πως περίπου γίνεται η διαχείριση αυτών των πακέτων. Το πακέτο λοιπόν αυτό θα αποσταλεί

στον ελεγκτή, ο οποίος δημιουργεί μια ή και περισσότερες εγγραφές ροής και ενημερώνει τους μεταγωγείς. Το πακέτο διαχειρίζεται πλέον από τις νέες εγγραφές που δημιουργήθηκαν.

Πέρα από τους πίνακες ροής υπάρχει και ο πίνακας ομάδας(group table) ο οποίος χρησιμοποιείται για πιο μαζική επεξεργασία της κίνησης. Περιέχει αναφορές προς πίνακες ροής για ομαδοποίηση των εγγραφών τους με κοινά χαρακτηριστικά και χρησιμοποιείται για πιο σύνθετη προώθηση πακέτων, όπως π.χ. flooding.
Τέλος, υπάρχει και ο meter table στον οποίο συλλέγονται στατιστικά και χρησιμοποιείται για την καλύτερη υλοποίηση λειτουργιών QoS στο δίκτυο.



Εικόνα 6: Τα βασικά συστατικά ενός OpenFlow Switch

Τα μηνύματα επικοινωνίας που αποστέλλονται σε ένα δίκτυο που χρησιμοποιεί OpenFlow χωρίζονται σε τρείς κατηγορίες:

Controller to Switch μηνύματα

- Χρησιμοποιούνται για την διαχείριση των εγγραφών ροής
- Αιτούνται πληροφορίες σχετικά με τις δυνατότητες και τους μετρητές ενός μεταγωγέα
- Αποστέλλουν ένα πακέτο πίσω στον μεταγωγέα

Ασύγχρονα μηνύματα

- Μηνύματα τα οποία αποστέλλονται από τους μεταγωγείς χωρίς προηγουμένως να υπάρχει αίτημα από τον ελεγκτή.

- Αποστέλλουν στον ελεγκτή ένα πακέτο το οποίο δεν αντιστοιχήθηκε στον πίνακα ροής του μεταγωγέα.
- Ενημερώνουν τον ελεγκτή για το ότι ένας timer έχει λήξει ή ότι έχει προκύψει ένα λάθος.

Συμμετρικά μηνύματα

- Μηνύματα που αποστέλλονται και από τις δυο πλευρές
- Hello και Echo μηνύματα
- Άλλα μηνύματα τα οποία καθορίζονται από τους κατασκευαστές του μεταγωγέα

Τα σημαντικότερα από αυτά είναι τα

- Packet-in
  Ασύγχρονο μήνυμα, μεταφέρει τον έλεγχο ενός πακέτου στον controller.
- Packet-out
  Controller to switch μήνυμα, αποτελεί την απάντηση στο Packet-in, παρέχει οδηγίες στον μεταγωγέα ώστε να προωθήσει ένα πακέτο από μια καθορισμένη θύρα.
- Modify-State
  Controller to switch μήνυμα το οποίο προσθέτει, διαγράφει και μεταβάλλει εγγραφές σε πίνακες ροής και ρυθμίζει τις ιδιότητες των θυρών του μεταγωγέα.
- Flow Removed
  Ασύγχρονο μήνυμα, ενημερώνει τον ελεγκτή για την αφαίρεση μια εγγραφής από έναν πίνακα ροής.

2.2.5 SDN Controller

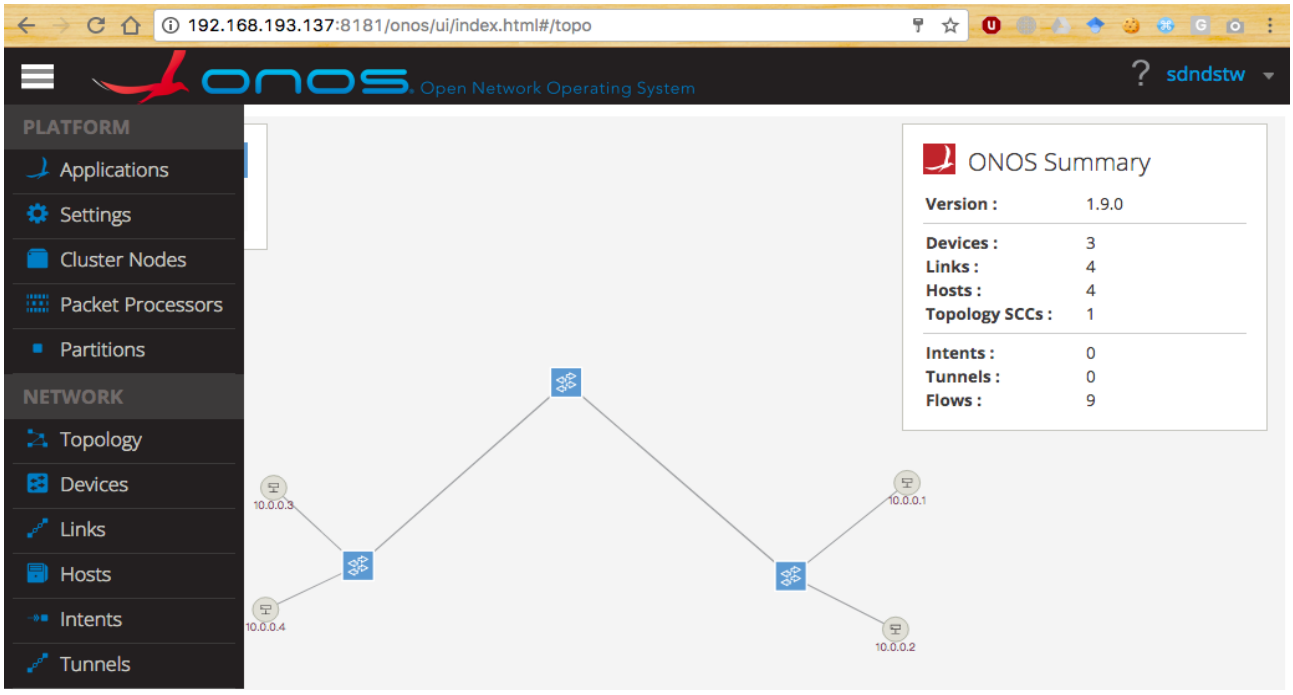
Ο ελεγκτής ή Controller αποτελεί κύρια οντότητα της αρχιτεκτονικής και όπως είπαμε ανήκει στο επίπεδο ελέγχου του SDN. Είναι η καρδιά και ο εγκέφαλος του δικτύου, οι ελεγκτές παρέχουν μια συγκεντρωτική εικόνα του συνολικού δικτύου και δίνουν τη δυνατότητα στους διαχειριστές δικτύων να διαβιβάζουν πληροφορίες όπως κανόνες δικτύωσης και πολιτικές στα υποκείμενα συστήματα (επίπεδο υποδομής , VMs, Virtual Routers & Switches).

Ένας SDN controller αποτελεί κατά κάποιο τρόπο το λειτουργικό σύστημα του δικτύου. "Απλώνεται" από τις δικτυακές συσκευές στο ένα άκρο ως τις εφαρμογές στο άλλο άκρο. Οποιαδήποτε επικοινωνία μεταξύ των δύο άκρων πρέπει να περάσει από τον controller.

Κάθε μια από τις πλατφόρμες SDN ελεγκτών που κυκλοφορούν περιέχει μια συλλογή από συνδέσιμες προεκτάσεις οι οποίες έχουν την δυνατότητα να υλοποιούν διαφορετικές δικτυακές εργασίες. Κάποιες από αυτές αφορούν την καταγραφή των συσκευών που είναι συνδεδεμένες στο δίκτυο και τις δυνατότητες της κάθε μιας από αυτές, την συλλογή στατιστικών κτλ. . Υπάρχει επίσης η δυνατότητα να εισαχθούν νέες επεκτάσεις οι οποίες ενισχύουν την λειτουργικότητα και υποστηρίζουν πιο εξελιγμένες δυνατότητες όπως αλγορίθμους οι οποίοι διεξάγουν ανάλυση της κίνησης και εντάσσουν νέους καταλληλότερους κανόνες στο δίκτυο.

Δυο από τα πιο γνωστά πρωτόκολλα που χρησιμοποιούνται από τους ελεγκτές για να επικοινωνούν με το Data Plane είναι το OpenFlow και το OVSDB. Η ομάδα εργασίας της Internet Engineering Task Force ανέπτυξε ένα πρότυπο για το SDN το οποίο δίνει την δυνατότητα σε έναν SDN controller να αξιοποιήσει γνωστά, αξιόπιστα πρωτόκολλα της παραδοσιακής αρχιτεκτονικής όπως τα OSPF, MPLS, BGP και IS-IS.

Έχουν υλοποιηθεί αρκετοί ελεγκτές, κάποιοι από αυτούς ανήκουν σε γνωστούς κατασκευαστές ενώ υπάρχουν και πολλοί οι οποίοι είναι ανοιχτού κώδικα όπως ο ONOS του Open Networking Foundation (ONF) που χρησιμοποιούμε παρακάτω στην υλοποίηση ενός SDN δικτύου. Οι δημοφιλέστεροι SDN ελεγκτές είναι οι Opendaylight, Beacon, NOX, POX, Project Floodlight κ.α.



Εικόνα 7: Το γραφικό περιβάλλον του ελεγκτή ανοιχτού κώδικα ONOS

2.3 Ανασκόπηση βιβλιογραφίας

Η εξάπλωση της αρχιτεκτονικής των Δικτύων που καθορίζονται από Λογισμικό πέρα από τα πλεονεκτήματα τα οποία προσφέρει στους διαχειριστές τους και τους προγραμματιστές εμφανίζει αρκετά ζητήματα ασφάλειας, είτε αυτά είναι παρόμοια με τα παραδοσιακά δίκτυα είτε αυτά είναι νέα και προκύπτουν από την διαφορετικότητα στην αρχιτεκτονική τους και τα νέα στοιχεία που την αποτελούν.
Παρακάτω καταγράφονται μηχανισμοί ασφαλείας που έχουν προταθεί από τους ερευνητές, έχουν υλοποιηθεί και χρησιμοποιούνται από την SDN κοινότητα.

Ο S. A. Mehdi με την ομάδα του [22], ενδιαφέρθηκαν για την ασφάλεια στα οικιακά και τα μικρά επιχειρηματικά δίκτυα. Παρουσιάζουν πώς υλοποιούνται τέσσερις γνωστοί αλγόριθμοι εντοπισμού ανωμαλιών στην κίνηση των δεδομένων σε περιβάλλον SDN με την χρήση μεταγωγέων συμβατών με το OpenFlow και τον NOX ως ελεγκτή. Τα αποτελέσματα των πειραμάτων δείχνουν ότι αυτοί οι αλγόριθμοι είναι σημαντικά πιο ακριβείς στον εντοπισμό κακόβουλων δραστηριοτήτων σε οικιακά δίκτυα σε σχέση με έναν ISP. Επίσης από την ανάλυση των SDN υλοποιήσεων ως προς την αποτελεσματικότητα τους σε ένα προγραμματίσιμο οικιακό δρομολογητή παρατηρείται ότι οι μηχανισμοί εντοπισμού των ανωμαλιών μπορούν να λειτουργήσουν χωρίς να δημιουργούν επιβάρυνση στην κίνηση του οικιακού δικτύου.

Ο J. R. Ballard και οι υπόλοιποι [23], ασχολούνται με την δρομολόγηση της κίνησης σε ένα δίκτυο ώστε να γίνει σωστά η διαχείριση της και να μην επιβαρυνθεί η απόδοση του δικτύου. Για τον λόγο αυτό προτείνουν το OpenSAFE, ένα σύστημα για την κατεύθυνση της κίνησης ώστε να παρακολουθείται από εφαρμογές παρακολούθησης για λόγους ασφάλειας . Περιγράφουν επίσης την ALARMS, μια γλώσσα καθορισμού των ροών η οποία απλοποιεί την διαδικασία παρακολούθησης της κίνησης από τους μηχανισμούς του δικτύου. Οι προτάσεις τους διευκολύνουν την παρακολούθηση δικτύων μεγάλης έκτασης και μάλιστα μειώνοντας το κόστος καθώς χρησιμοποιούν συσκευές συμβατές με OpenFlow.

Ο Syed Taha Ali και οι υπόλοιποι [24] παρουσιάζουν στην έρευνα τους τρόπους για την προστασία ενός SDN δικτύου από τις πιθανές απειλές ενώ προτείνουν την υλοποίηση και την πιθανή παροχή της ασφάλειας ως υπηρεσίας (NSaaS). Με τον τρόπο αυτό το πρότυπο του SDN διευκολύνει την υλοποίηση επιπρόσθετων μέτρων ασφάλειας στο δίκτυο τα οποία προχωρούν πέρα από την προστασία του δικτύου ενεργοποιώντας υπηρεσίες όπως την

ανωνυμία, την ενισχυμένη εμπιστοσύνη και την απομακρυσμένη διαχείριση. Η υπηρεσία αυτή έχει την δυνατότητα να παρέχεται σε μικρότερες εταιρίες και οργανισμούς οι οποίοι δεν έχουν την τεχνογνωσία και τους οικονομικούς πόρους να προστατευθούν από μόνοι τους και αναθέτουν σε τρίτους που διαθέτουν την επιχειρησιακή κατάρτιση να αναλάβουν την προστασία του δικτύου τους.

Ο R. Braga με την ομάδα του [25], παρουσιάζουν μια "ελαφριά" μέθοδο εντοπισμού DDoS επιθέσεων η οποία βασίζεται στα χαρακτηριστικά της κίνησης ροών, στην οποία η εξαγωγή πληροφοριών και συμπερασμάτων γίνεται με πολύ λιγότερη επιβάρυνση σε σχέση με τις παραδοσιακές προσεγγίσεις. Εμφανίζει επίσης πολύ υψηλό βαθμό επιτυχίας στον εντοπισμό των επιθέσεων και πολύ χαμηλό αριθμό λανθασμένων συναγερμών λόγω της ανάλυσης των ροών με την χρήση Self Organizing Maps, ενός νευρωνικού δικτύου τεχνητής νοημοσύνης, το οποίο εκπαιδεύεται με χαρακτηριστικά της κίνησης ροών.
Η μέθοδος υλοποιείται μέσω ενός δικτύου που βασίζεται στον ελεγκτή NOX, όπου οι OpenFlow μεταγωγείς διατηρούν πίνακες ροής με στατιστικά για όλες τις ενεργές ροές. Όλες οι πληροφορίες των χαρακτηριστικών που χρειάζονται διαχειρίζονται μέσω του NOX ελεγκτή με ευφυΐα και στην συνέχεια επεξεργάζονται με ένας έξυπνο μηχανισμό εντοπισμού των επιθέσεων. Η μέθοδος αυτή είναι σε ακριβώς αντίθετη κατεύθυνση με τις παραδοσιακές τεχνικές οι οποίες όλες απαιτούν αρκετή επεξεργασία των δεδομένων ώστε να εξάγουν πληροφορίες που χρειάζονται για την ανάλυση της κίνησης.

Ο Z. Hu και οι υπόλοιποι [26] παρουσιάζουν μια ευρεία και συνολική αρχιτεκτονική ασφάλειας για τα Software Defined Networks. Μέσω αυτής προτείνονται μηχανισμοί ασφάλειας οι οποίοι καλύπτουν τις προκλήσεις που καταγράφονται στην έρευνά τους. Συγκεκριμένα, προτείνεται η ενίσχυση προκαθορισμένων υποχρεωτικών πολιτικών στο δίκτυο και το πως αυτές θα υλοποιούνται με αποτελεσματικότητα. Παρουσιάζεται επίσης μια μέθοδος που σχεδίασαν με την οποία ελέγχονται και τα δεδομένα των πακέτων με τον ελεγκτή να μπορεί να περιορίσει επιθέσεις όπως από worms, Trojans και spam information, ενώ έχει την δυνατότητα να ελέγχει και τους headers των πακέτων και να εντοπίζει επιθέσεις και από αυτούς και να τους αντιμετωπίζει αυτόματα. Τέλος, προτείνει μεθόδους με τις οποίες ενισχύεται η ασφάλεια στην επικοινωνία μέσω του Northbound API, ώστε να περιορίζονται οι επιθέσεις από application servers οι οποίοι έχουν καταληφθεί από εισβολείς.

Οι S. Shirali-Shahreza and Y. Ganjali [27], παρουσιάζουν μια ευέλικτη δειγματοληπτική επέκταση για το OpenFlow, η οποία σχεδιάστηκε για να παρέχει πρόσβαση στην πληροφορία σε επίπεδο πακέτων. Η επέκταση αυτή ονομάζεται FleXam, και λόγω της απλότητας της είναι εύκολο να υλοποιείται σε μεταγωγείς που υποστηρίζουν OpenFlow και λειτουργεί στο line rate χωρίς να απαιτείται επιπρόσθετη μνήμη. Την ίδια στιγμή η ευελιξία της επιτρέπει την υλοποίηση διάφορων εφαρμογών παρακολούθησης και ασφάλειας στον ελεγκτή, διατηρώντας την ισορροπία ανάμεσα στην επιβάρυνση και την λεπτομέρεια στην συλλογή πληροφοριών. Χρησιμοποιεί εξίσου προληπτικούς αλλά και αντιδραστικούς μηχανισμούς δρομολόγησης παρέχοντας ένα συντονισμένο αντιστάθμισμα μεταξύ της ορατότητας των ανεξάρτητων ροών και του φορτίου του ελεγκτή.
Για να κάνουν μια επίδειξη με το πως χρησιμοποιείται το FleXam και πως μπορεί να υλοποιήσει εφαρμογές ασφαλείας, παρουσιάζουν την υλοποίηση του Threshold Random Walk port scan detection, η οποία είναι μία από τις πιο γνωστές μεθόδους port scan detection.

Παρά την επιτυχία του OpenFlow, η ανάπτυξη και η ανάπτυξη πολύπλοκων υπηρεσιών ασφαλείας παραμένει μια σημαντική πρόκληση. Ο S. Shin και η ομάδα του [28], παρουσιάζουν το FRESCO, ένα πλαίσιο ανάπτυξης εφαρμογών ασφαλείας OpenFlow το οποίο έχει σχεδιαστεί για να διευκολύνει την ραγδαία ανάπτυξη σχεδιασμού και την αρθρωτή σύνθεση OpenFlow-enabled modules για την ανίχνευση και τον περιορισμό των επιθέσεων. Το FRESCO εξάγει ένα scripting API που επιτρέπει στους επαγγελματίες ασφάλειας να κωδικοποιούν την παρακολούθηση της κίνησης και την ανίχνευση απειλών ως αρθρωτές βιβλιοθήκες. Αυτές οι αρθρωτές βιβλιοθήκες αντιπροσωπεύουν τις στοιχειώδεις μονάδες επεξεργασίας στο FRESCO και μπορούν να μοιραστούν και να συνδεθούν μεταξύ τους για να παρέχουν σύνθετες εφαρμογές προστασίας του δικτύου. Τα modules του FRESCO μπορούν επίσης να παράγουν κανόνες ροής, παρέχοντας έτσι ένα αποτελεσματικό μέσο για την εφαρμογή οδηγιών ασφαλείας που μπορεί να αναφερθούν από άλλα modules του FRESCO.
Οι αξιολογήσεις τους αποδεικνύουν ότι η FRESCO εισάγει ελάχιστη επιβάρυνση στο δίκτυο και ότι επιτρέπει την ταχεία δημιουργία δημοφιλών λειτουργιών ασφαλείας με σημαντικά λιγότερες γραμμές κώδικα. Μπορεί να προσφέρει ένα ισχυρό νέο πλαίσιο για την δημιουργία πρωτοτύπων και την παροχή καινοτόμων εφαρμογών ασφάλειας, ενώ σχεδιάζεται να απελευθερωθεί όλος ο ανεπτυγμένος κώδικας ως open source software για την SDN κοινότητα.

Το Cloudwatcher [29]αποτελεί ένα εργαλείο για την παρακολούθηση των cloud υπηρεσιών. Μερικά πακέτα με βάση υποθέσεις ασφαλείας, αναδρομολογούνται σε ένα σημείο αναφοράς ασφάλειας, για περαιτέρω επιθεώρηση. Η εφαρμογή περιλαμβάνει τρία στοιχεία : ένα διαχειριστή συσκευής και πολιτικών, μια γεννήτρια κανόνων δρομολόγησης, και έναν επιβολέα κανόνων ροών. Αυτή η εργασία ανέδειξε δύο θέματα, της παρακολούθησης cloud κίνησης: την ανάγκη να ληφθούν υπόψη τόσο οι εξωτερικές όσο και οι εσωτερικές απειλές και την ανάγκη να ληφθεί υπόψη το γεγονός ότι τα cloud δίκτυα είναι πολύ δυναμικά και οι υπολογιστές ή τα δικτυακά συστατικά τους μπορεί να αλλάζουν συχνά. Το Cloudwather προτάθηκε ως Northbound εφαρμογή του ελεγκτή και παρέχει την υπηρεσία παρακολούθησης σε διαφορετικούς μηχανισμούς ασφαλείας.

Ο Wen X και η ομάδα του [30], πρότειναν το PermOF, ένα σύστημα διαχείρισης ελέγχου πρόσβασης, το οποίο περιλαμβάνει επίπεδα πρόσβασης, για τον ελεγκτή και τους δικτυακούς πόρους. Τα απλά επίπεδα περιορισμένης εξουσιοδότησης με μόνο δύο ή τρία επίπεδα εξουσιοδότησης (συμπεριλαμβανομένου και του διαχειριστή) αποτελούν εύκολο στόχο για αλλοίωση ή για επιθέσεις ανύψωσης δικαιωμάτων. Η προτεινόμενη προσέγγιση παρέχει ένα σύνολο 18 πιθανών επιπέδων δικαιωμάτων. Οι εφαρμογές λαμβάνουν ένα, προεπιλεγμένο, ελάχιστο επίπεδο δικαιωμάτων. Τα ερωτήματα από το API του ελεγκτή είναι αυτά που πυροδοτούν την επικοινωνία με τις εφαρμογές.

Ο Mendonca με την ομάδα του [31], εισήγαγαν το AnonyFlow, μια ενδοδικτυακή υπηρεσία ανωνυμίας βασισμένη στο OpenFlow. Ο βασικός στόχος αυτής της προσέγγισης είναι η καταγραφή τελικού σημείου. Οι IP διευθύνσεις μεταφράζονται σε αναγνωριστικά ανωνυμίας (δηλαδή προορισμούς), τα οποία οι τρίτοι μπορούν απλώς να δουν. Το AnonyFlow είναι υπεύθυνο για την μετάφραση, μεταξύ αυτών των αναγνωριστικών και των IP διευθύνσεων.

Ένας πυρήνας επιβολής της ασφάλειας προτείνεται στο [32] ως λύση στο πρόβλημα των εφαρμογών από έναν κακόβουλο ελεγκτή. Το FortNOX εφαρμόζει έλεγχο ταυτότητας βάση ρόλων για τον καθορισμό της εξουσιοδότησης ασφαλείας κάθε εφαρμογής OpenFlow. Ο μηχανισμός επιβολής FortNox χειρίζεται πιθανές συγκρούσεις με την εισαγωγή κανόνων ροής, όπου η αποδοχή ή απόρριψη του κανόνα εξαρτάται από την εξουσιοδότηση ασφαλείας που διαθέτει ο δημιουργός της. Ένας νέος κανόνας ροής που έρχεται σε σύγκρουση με έναν υπάρχοντα κανόνα θα ανιχνεύεται από το FortNOX. Αν ο νέος κανόνας ροής δημιουργήθηκε από έναν συγγραφέα υψηλότερης προτεραιότητας, τότε ο υπάρχων κανόνας ροής θα

αντικατασταθεί. Ωστόσο, εάν ο νέος κανόνας ροής παράγεται από έναν συγγραφέα με χαμηλότερη προτεραιότητα, τότε θα αγνοηθεί. Διακρίνονται τρεις ρόλοι παραγωγών κανόνων ροής, του χειριστή, της ασφάλειας και της εφαρμογής. Ένας περιορισμός αυτής της προσέγγισης είναι ο προσδιορισμός του κατάλληλου επιπέδου έγκρισης ασφαλείας. Το FortNOX δεν επιλύει το ζήτημα της αναγνώρισης των εφαρμογών και της επιβολής της προτεραιότητας.

Το AVANT-GUARD [33] προτείνει μια λύση στην συμφόρηση της επικοινωνίας στο επίπεδο ελέγχου του SDN περιορίζοντας τα αιτήματα ροής που αποστέλλονται στο επίπεδο ελέγχου χρησιμοποιώντας ένα connection migration tool. Κατά μία έννοια, αποτελεί κάτι ανάμεσα σε λύση ενός ζητήματος ασφάλειας και σε μια βελτίωση SDN στην ασφάλεια του δικτύου. Η λύση για τις επιθέσεις DoS στο SDN επισημαίνονται, ενώ γίνεται επίσης ανάλυση της κίνησης και της λειτουργικότητας στην ενημέρωση των κανόνων ροής. Εστιάζοντας στην επίθεση SYN flood, το AVANT-GUARD χρησιμοποιεί ένα connection migration tool για να αφαιρέσει αποτυχημένα TCP sessions στο επίπεδο δεδομένων πριν από οποιαδήποτε ειδοποίηση στο επίπεδο ελέγχου. Αυτό αποτρέπει την δυνατότητα κορεσμού ή DoS επίθεσης στέλνοντας μόνο τα αιτήματα ροής στο επίπεδο ελέγχου τα οποία ολοκληρώνουν το TCP handshake.

Με το ROSEMARY [34], οι συγγραφείς προτείνουν ένα ισχυρό, ασφαλές, υψηλής απόδοσης λειτουργικό σύστημα δικτύου (NOS). Ο κεντρικός στόχος του ROSEMARY είναι να βελτιώσει την ανθεκτικότητα του επιπέδου ελέγχου στις κακόβουλες εφαρμογές. Για να επιτευχθεί αυτό, οι συγγραφείς προτείνουν μια μικρή NOS αρχιτεκτονική. Κάθε εφαρμογή OpenFlow εκτελείται μέσα σε ένα ανεξάρτητο στιγμιότυπο του ROSEMARY απομονώνοντας αποτελεσματικά την εφαρμογή για την προστασία του επιπέδου ελέγχου από οποιαδήποτε ευπάθεια ή κακόβουλη λειτουργία της εφαρμογής. Η λύση χωρίζει τις εφαρμογές του δικτύου από την αξιόπιστη υπολογιστική βάση του NOS, παρακολουθεί και ελέγχει τους πόρους του δικτύου που καταναλώνονται από κάθε εφαρμογή, παρακολουθεί και ελέγχει λειτουργίες εφαρμογής όπως προνομιούχες κλήσεις συστήματος και εφαρμόζει μια ασφαλή διαδικασία επανεκκίνησης του NOS για να επανεκκινήσει προσεκτικά κάθε υπηρεσία.

Απαντάται έτσι το πρώτο μας ερευνητικό ερώτημα κατά μεγάλο βαθμό με την βιβλιογραφική αυτή επισκόπηση, δηλαδή ποιοι μηχανισμοί έχουν αναπτυχθεί για την παροχή ασφάλειας στους χρήστες ενός Software Defined Network. Υπάρχουν ωστόσο κι άλλοι μηχανισμοί οι οποίοι θα μελετηθούν σε μελλοντική μας έρευνα.

Κεφάλαιο 3
Θέματα ασφαλείας στα SDN δίκτυα

Η πολυεπίπεδη αρχιτεκτονική SDN μπορεί να προσφέρει πολλά πλεονεκτήματα έναντι των παραδοσιακών δικτύων ακόμα και στον τομέα της ασφάλειας δημιουργεί όμως και νέες προκλήσεις όσον αφορά την αντιμετώπιση απειλών οι οποίες μπορούν να εμφανιστούν σε κάθε επίπεδο. Πηγές ευπάθειας μπορούν να αποτελούν οι SDN εφαρμογές και η κονσόλα διαχείρισης που χρησιμοποιούν τη διεπαφή northbound στο πεδίο εφαρμογών, ο ελεγκτής, οι southbound east/westbound διεπαφές στο πεδίου ελέγχου καθώς και οποιαδήποτε στοιχείο του δικτύου στο επίπεδο υποδομής. Παρακάτω παρουσιάζονται λεπτομερώς οι σημαντικότερες απειλές ταξινομημένες με βάση τα σημαντικότερα λειτουργικά στοιχεία του καθώς και κάποιες ιδέες για την αντιμετώπισή τους.

3.1 Επίπεδο Εφαρμογών (Application Plane)

Όπως αναφέρθηκε και προηγουμένως το βασικότερο χαρακτηριστικό της αρχιτεκτονικής SDN είναι η ύπαρξη ενός επιπρόσθετου επιπέδου αυτού της εφαρμογής (application layer) καθώς και η διεπαφή (interface) μεταξή του ελεγκτή και της εφαρμογής που ονομάζεται Northbound API. Το επιπλέον αυτό επίπεδο είναι που δίνει τη δυνατότητα στα δίκτυα να είναι προγραμματίσιμα (να ελέγχονται δηλαδή μέσω λογισμικού) αλλά και αυτοματοποιήσιμα (το λογισμικό έχει την δυνατότητα να εντοπίσει ένα πρόβλημα στο δίκτυο αλλά και να επέμβει για την επίλυσή του). Παρόλα τα πλεονεκτήματα της αρχιτεκτονικής αυτής γίνεται εύκολα αντιληπτό ότι εισάγει και σοβαρούς κινδύνους ασφάλειας.

Κακόβουλη Πρόσβαση σε SDN εφαρμογή

Μία είδους επίθεση θα μπορούσε για παράδειγμα να είναι η απόκτηση πρόσβασης από έναν κακόβουλο hacker στον πηγαίο κώδικα μιας Northbound εφαρμογής ή service όπως

αποκαλείται στην ορολογία του SDN. Θα μπορούσε λοιπόν να αλλάξει τον κώδικα της εφαρμογής ώστε να επιτύχει κάποιο συγκεκριμένο σκοπό (DoS, sniffing κτλ.).[21] Καταλαβαίνουμε λοιπόν ότι ακόμα και η γλώσσα προγραμματισμού που χρησιμοποιείται για την υλοποίηση της εφαρμογής παίζει το ρόλο της σε σχέση με την ασφάλεια του δικτύου. Για παράδειγμα μια μεταγλωττισμένη (compiled) εφαρμογή γραμμένη σε C είναι πιο δύσκολο να μελετηθεί και να "χτυπηθεί" αφού δεν υπάρχει ο πηγαίος κώδικας αλλά μόνο το εκτελέσιμο αρχείο (binary). Από την άλλη μια εφαρμογή που είναι γραμμένη σε μορφή script σε bash ή python θα μπορούσε να προσφέρει πολύτιμες πληροφορίες στον hacker σε σχέση με το API και φυσικά μπορεί πολύ πιο εύκολα να μετατρέψει τον κώδικα ώστε να επιτύχει το σκοπό του.

Κακόβουλη πρόσβαση στην Κονσόλα Ελέγχου του Ελεγκτή
Ακόμη χειρότερα για την ασφάλεια του δικτύου θα μπορούσε να είναι τα πράγματα εάν ο κακόβουλος εισβολέας αποκτήσει ολική πρόσβαση στο Northbound API. Πολλοί ελεγκτές δίνουν τη δυνατότητα εκτέλεσης εντολών σε κονσόλα (γνωστό και ως management API) από κάποιον χρήστη. Έτσι, ο χρήστης έχει στη διάθεση του όλες τις functions της διεπαφής (API calls) με τις οποίες μπορεί να αποκτήσει ολοκληρωμένη άποψη για την τοπολογία του δικτύου, να διακόψει τη λειτουργία του δικτύου (διαγράφοντας για παράδειγμα τους πίνακες OpenFlow όλων των συσκευών) ή να αποκτήσει πρόσβαση σε ευαίσθητα δεδομένα (evesdropping). Είναι φανερό ότι οποιοσδήποτε αποκτήσει πρόσβαση στο Northbound API είναι σαν να κατέχει τα κλειδιά του δικτύου με ότι αυτό συνεπάγεται για την ασφάλεια και τη σωστή του λειτουργία.

3.1.1 Τρόποι αντιμετώπισης

Όπως για όλα τα θέματα ασφαλείας δικτύου έτσι και για την ασφάλεια στο επίπεδο εφαρμογών μπορούν να χρησιμοποιηθούν διάφορες λύσεις ή και συνδυασμός αυτών ώστε να ελαχιστοποιηθεί η πιθανότητα απόκτησης κακόβουλης πρόσβασης στη διεπαφή. Παρακάτω προτείνουμε μερικές από αυτές:

1. Ισχυρή προστασία των servers που φιλοξενούν (hosting) τα SDN services. Εγκατάσταση ενημερωμένων patches, χρήση ισχυρών κωδικών για αποφυγή brute forcing και password guessing επιθέσεων, αποφυγή εκτέλεσης συγκεκριμένων εφαρμογών με δικαιώματα administrator κτλ.

2. Αποφυγή εφαρμογών που είναι γραμμένες σε πηγαίο κώδικα. Ακόμα και αν κάποιος αποκτήσει πρόσβαση στους εξυπηρετητές των εφαρμογών η εξαγωγή πληροφοριών από εκτελέσιμο αρχείο είναι πολύ δύσκολη (αν και όχι αδύνατη). Η απόκτηση πρόσβασης σε πηγαίο κώδικα είναι από μόνη της σοβαρό πρόβλημα καθώς μπορεί να δώσει πολλές πληροφορίες σε έναν ικανό κακόβουλο εισβολέα σε σχέση με την τοπολογία του δικτύου, τη λειτουργία του, τις υπάρχουσες ρυθμίσεις ασφάλειας κτλ.

3. Εγκατάσταση μηχανισμού αυθεντικοποίησης, χρήση ισχυρών συνθηματικών και κρυπτογράφηση (π.χ. με TLS) του Northbound API. Κανένας χρήστης ή service δε θα πρέπει να μπορεί να κάνει χρήση του Northbound API εάν προηγουμένως δεν έχει αποδείξει την ταυτότητα του στον ελεγκτή.

4. Εγκατάσταση συστήματος καταγραφής γεγονότων (event logging) και ανίχνευση κακόβουλων ενεργειών στο επίπεδο του ελεγκτή. Με τη συλλογή των logs και την ανάλυσή τους το σύστημα μπορεί να εντοπίσει και να μπλοκάρει τις απειλές αποτελεσματικότερα. Η καταγραφή βοηθάει επίσης στην αντιμετώπιση της απάρνησης (non-repudiation) καθώς καταγράφονται όλες οι ενέργειες ενός χρήστη ή μιας εφαρμογής.

3.2 Επίπεδο Ελέγχου (Control Plane)

Με τον όρο επίπεδο ελέγχου εννοούμε τον ελεγκτή του δικτύου καθώς και τη Νότια Διεπαφή (Southbound API). Είναι φανερό ότι το μεγάλο πλεονέκτημα της αρχιτεκτονικής SDN, δηλαδή η κεντρικοποίηση της διαχείρισης του δικτύου, μπορεί να αποτελέσει και το πιο αδύναμό του σημείο. Ο ελεγκτής του δικτύου αποτελεί την καρδιά του δικτύου αλλά ταυτόχρονα είναι και μοναδικό σημείο αποτυχίας (Single Point of Failure) και είναι λογικό να θεωρείται κύριος στόχος από επίδοξους κακόβουλους εισβολείς.

Denial of Service

Μία από τις πιο διαδεδομένες επιθέσεις στα παραδοσιακά δίκτυα, η επίθεση Άρνησης Εξυπηρέτησης μπορεί να έχει πολύ πιο καταστροφικές συνέπειες για τη λειτουργία του δικτύου εάν διεξαχθεί με επιτυχία στον ελεγκτή. Παραδοσιακά η επίθεση DoS στοχεύει είτε κάποιον εξυπηρετητή (για παράδειγμα έναν web server ο οποίος αδυνατεί να εξυπηρετήσει τους χρήστες) είτε κάποια συσκευή δικτύου (για παράδειγμα έναν δρομολογητή ο οποίος δεν

μπορεί πλέον να δρομολογήσει τη νόμιμη κίνηση). Οι συνέπειες είναι συνήθως ένα server downtime (ο χρόνος από την έναρξη της επίθεσης μέχρι την αντιμετώπιση της ώστε ο server να αρχίσει να εξυπηρετεί ξανά) μόνο των εξυπηρετητών που δέχονται την επίθεση ή στη χειρότερη περίπτωση απώλεια νόμιμης κίνησης εώς ότου τα routing πρωτόκολλα των δρομολογητών (π.χ. OSPF, RIP κτλ.) να επιλύσουν το πρόβλημα με επαναδρομολόγηση (rerouting) της κίνησης από κάποιον άλλο διαθέσιμο δρομολογητή αν φυσικά έχει προβλεφθεί κάτι τέτοιο από τους διαχειριστές. Μία DoS επίθεση στον ελεγκτή μπορεί να είναι καταστροφική για τη λειτουργία του δικτύου. Ένας ελεγκτής που δεν μπορεί να δεχτεί Packet-In μηνύματα ούτε να ενημερώσει τους πίνακες των μεταγωγέων θα οδηγήσει αργά ή γρήγορα σε διακοπή της λειτουργίας του δικτύου και συνεπώς κανένας εξυπηρετητής δε θα είναι πια διαθέσιμος στους χρήστες. Οι νέες συνδέσεις θα αποτυγχάνουν καθώς δε θα μπορούν να προωθηθούν από τους μεταγωγείς οι οποίοι δε θα γνωρίζουν τι να κάνουν και οι ήδη υπάρχουσες συνδέσεις θα συνεχίσουν να προωθούνται έως ότου επέλθει η λήξη του κανόνα στους πίνακες ροής (στις περισσότερες περιπτώσεις μερικές δεκάδες δευτερόλεπτα). Σημειώνεται ότι η επίθεση DoS είναι δυνατή σε όλα τα επίπεδα ενός SDN δικτύου με την επίθεση στον ελεγκτή να είναι η πιο σοβαρή. Με δεδομένο το ότι η διαδικασία χειραψίας του openflow δεν απαιτεί από τον ελεγκτή την αυθεντικοποίηση των μεταγωγέων πριν οι τελευταίοι συνδεθούν στον ελεγκτή η DoS είναι σχετικά εύκολη υπόθεση. Κακόβουλοι μεταγωγείς μπορούν να συνδεθούν στον ελεγκτή και να τον υπερχειλίσουν με Packet-In πακέτα.

Man in the Middle

Στην επίθεση MITM ένας εισβολέας παρεμβάλλεται ανάμεσα σε δύο επικοινωνούντες πλευρές (parties) χωρίς οι τελευταίες να το γνωρίζουν θεωρώντας ότι επικοινωνούν απευθείας η μία με την άλλη. Έτσι ο εισβολέας έχει την δυνατότητα να παρακολουθεί όλη την επικοινωνία και αποκτά πρόσβαση στα δεδομένα που ανταλλάσσονται. Σε ένα κλασικό δίκτυο ο εισβολέας έμπαινε συνήθως ανάμεσα σε ένα χρήστη κι έναν εξυπηρετητή ή σε ένα χρήστη κι ένα δρομολογητή. Υπάρχουν πολλοί τρόποι διεξαγωγής μιας τέτοιας επίθεσης με το ARP Spoofing να είναι ίσως η πιο διαδεδομένη. Στόχος της είναι να συσχετιστεί η διεύθυνση IP ενός νόμιμου host (για παράδειγμα ο default gateway ενός δικτύου) με την διεύθυνση MAC του επιτιθέμενου. Σε ένα SDN δίκτυο μία πιθανή τέτοια επίθεση θα μπορούσε να είναι η παρεμβολή ενός κακόβουλου εισβολέα ανάμεσα στους openflow μεταγωγείς και το νόμιμο ελεγκτή. Η επίθεση αυτή θα μπορούσε να θεωρήθει και spoofing attack στην οποία ο

επιτιθέμενος πλαστογραφεί την ταυτότητα του νόμιμου ελεγκτή και εγκαθιστά ένα κανάλι με τους μεταγωγείς μέσω του οποίου ανταλλάσσονται πακέτα OpenFlow. Έτσι. ο επιτιθέμενος λαμβάνει Packet-In πακέτα από τους μεταγωγείς (με αποτέλεσμα να έχει πλήρη πρόσβαση στο payload των πακέτων) και μπορεί να στέλνει κακόβουλα Packet-Out πακέτα και Flow Mod πακέτα τα οποία μπορούν να παραλύσουν το δίκτυο (για παράδειγμα σβήσιμο όλων των κανόνων ροής) ή να προωθήσουν την κίνηση σε ελεγχόμενες από τον ίδιο συσκευές. Η πρόσβαση στην διερχόμενη κίνηση από έναν κακόβουλο ελεγκτή είναι πολύ σοβαρή υπόθεση αν αναλογιστεί κανείς ότι η χρήση TLS (Transport Layer Security) είναι προαιρετική στο πρωτόκολλο OpenFlow (παρόλο που στην πρώτη έκδοση του v1.0 η χρήση ασφαλούς και κρυπτογραφημένου καναλιού ήταν υποχρεωτική στις επόμενες εκδόσεις έγινε προαιρετική).

3.2.1 Τρόποι αντιμετώπισης

1. Χρήση πλεοναζόντων ελεγκτών.

Η λύση αυτή έχει ως σκοπό να βελτιώσει την ασφάλεια του δικτύου μέσω αυξημένης διαθεσιμότητας (redundancy) των πόρων και συγκεκριμένα του ελεγκτή. Καθώς όπως ειπώθηκε ο ελεγκτής αποτελεί Μοναδικό Σημείο Αποτυχίας για το δίκτυο είναι αναμενόμενη η χρήση περισσότερων του ενός ελεγκτών ώστε σε περίπτωση επίθεσης εναντίον του (DoS, spoofing κτλ.) ένας άλλος νόμιμος ελεγκτής να αναλάβει τη διαχείριση του δικτύου.

2. Υλοποίηση του ελεγκτή σε διαφορετικού τύπου συστήματα (diversity)

Εκτός από την απλή αντιγραφή ενός ελεγκτή για την διατήρηση της αυξημένης διαθεσιμότητας, μπορεί να χρησιμοποιηθεί και μια ποικιλία (diversity) στην υλοποίηση των ελεγκτών ώστε να μετριάζονται οι επιθέσεις. Ένα προφανές παράδειγμα είναι η υλοποίηση του ελεγκτή σε διαφορετικά λειτουργικά συστήματα (π.χ. Linux και Windows) ώστε οι εγγενείς ευπάθειες (vulnerabilities) του ενός λειτουργικού να μετριάζονται από τις υπηρεσίες ασφαλείας ενός άλλου OS.

3. Ενίσχυση ασφάλειας και εμπιστοσύνης ανάμεσα στον ελεγκτή και τους μεταγωγείς. Η λύση αυτή απαιτεί την εγκατάσταση ενός συστήματος αυθεντικοποίησης μέσω του οποίου ο ελεγκτής θα επιτρέπει τη σύνδεση μόνο σε νόμιμους μεταγωγείς με τα κατάλληλα credentials. Επίσης, η κρυπτογράφηση του καναλιού επικοινωνίας θα πρέπει να είναι υποχρεωτική ώστε οι συνέπειες της υποκλοπής κίνησης να μετριάζονται.

3.3 Επίπεδο Δεδομένων (Data Plane)

Νέου τύπου επιθέσεις μπορούν να προκύψουν και στο επίπεδο δεδομένων του SDN. Ακολουθούν δύο παραδείγματα.

Denial of Service

Μία νέα πιθανή DoS επίθεση είναι η λεγόμενη data-to-control plane saturation attack [3]. Ένας επιτιθέμενος θα μπορούσε να κάνει χρήση πολλών compromised συσκευών ώστε να στέλνει μεγάλο όγκο πακέτων στους OpenFlow μεταγωγείς με τυχαία πεδία στην επικεφαλίδα κάθε πακέτου. Έτσι, η τυχαιοποίηση των επικεφαλίδων μειώνει την πιθανότητα να υπάρχουν εγκατεστημένοι κανόνες ροής στο μεταγωγέα κι έτσι ο τελευταίος θα αναγκάζεται να ανταλλάσει μεγάλο όγκο Packet-In πακέτων με τον ελεγκτή, γεγονός που μπορεί να σημαίνει υψηλή κατανάλωση πόρων του ελεγκτή αλλά και υπερχείλιση των πινάκων του μεταγωγέα.

Topology Poisoning Attacks

Ο ελεγκτής ενός δικτύου SDN βασίζεται στο πρωτόκολλο Link Layer Discovery Protocol (LLDP) για να μάθει την τοπολογία του δικτύου στο επίπεδο υποδομής. Στην επίθεση Topology Poisoning ένας επιτιθέμενος μπορεί αφού παρακολουθήσει για κάποιο διάστημα τη νόμιμη LLDP κίνηση, να συλλέξει πληροφορίες για την τοπολογία του δικτύου και στη συνέχεια να αποστείλει τροποποιημένα πακέτα στον ελεγκτή ώστε να δημιουργήσει στον τελευταίο λανθασμένη εντύπωση για την τοπολογία του δικτύου. Ένα παράδειγμα που αναφέρουν οι S. Gao, Z. Li, B. Xiao, G. Wei. [3] είναι η δημιουργία ψευδών συνδέσεων μεταξύ των μεταγωγέων. Δηλαδή, ενώ δεν υπάρχει φυσική σύνδεση μεταξύ δύο μεταγωγέων, τα κακόβουλα πακέτα που λαμβάνει ο ελεγκτής από τον επιτιθέμενο τον κάνουν να πιστέψει ότι οι δύο μεταγωγείς συνδέονται. Με αυτόν τον τρόπο ο επιτιθέμενος μπορεί να εξαπολύσει επιθέσεις DoS (μπλοκάροντας για παράδειγμα κάποιες νόμιμες θύρες του ελέγκτή) αλλά και επιθέσεις τύπου MITM.

3.3.1 Τρόποι αντιμετώπισης

Στην βιβλιογραφία [3],[4] δίνονται παραδείγματα και υλοποιήσεις αντιμετώπισης των προβλημάτων ασφαλείας στο επίπεδο δεδομένων. Για παράδειγμα, προτείνεται [3] η επέκταση ενός μεταγωγέα με ένα TCP Proxy ο οποίος θα έχει τη δυνατότητα να ελέγχει εάν ο

αποστολέας και παραλήπτης της κίνησης είναι υπαρκτοί ώστε να αποκλείεται η πιθανότητα DoS επίθεσης βασισμένη σε TCP. Επίσης, προτείνεται ο αποκλεισμός LLDP πακέτων που προέρχονται από hosts συνδεδεμένους σε ένα μεταγωγέα και να επιτρέπονται μόνο LLDP πακέτα από μεταγωγείς (αφού ένας host είναι πιο πιθανό να είναι κακόβουλος από έναν μεταγωγέα) για την αντιμετώπιση της Topology Poisoning επίθεσης. Τέλος, κάποιες καλές πρακτικές για τον μετριασμό των επιθέσεων στους μεταγωγείς είναι η ομαδοποίησης των κανόνων ροής (flow aggregation) ώστε να αποφεύγονται οι υπερχειλίσεις στους πίνακες και περιορισμός του όγκου των πακέτων ελέγχου (rate limiting) ώστε να μετριάζονται οι επιθέσεις DoS.

3.4 Συμπεράσματα

Στο συγκεκριμένο κεφάλαιο επιχειρήσαμε να απαντήσουμε στο ερώτημα ποιες είναι οι προκλήσεις ασφάλειας που πρέπει να αντιμετωπιστούν με την χρήση του SDN. Αναλύσαμε τα περισσότερα και πιο κοινά προβλήματα ασφαλείας που παρουσιάζει η νέα αρχιτεκτονική SDN. Είναι αντιληπτό ότι παρόλο που το SDN προσφέρει κάποια σοβαρά πλεονεκτήματα έναντι των κλασικών δικτύων, η κεντρική διαχείριση και η πολυεπίπεδη αρχιτεκτονική που εισάγει δημιουργεί νέα τρωτά σημεία και νέες προκλήσεις για τους μηχανικούς ασφαλείας των δικτύων. Τα προβλήματα αυτά πρέπει να ληφθούν σοβαρά υπόψη ώστε να μην επαναληφθούν τα λάθη του παρελθόντος με τα παραδοσιακά δίκτυα όπου η ασφάλεια άρχισε να λαμβάνεται υπόψη αρκετά αργότερα από την ανάπτυξή τους.

Κεφάλαιο 4

Υπεροχή των δικτύων SDN έναντι των παραδοσιακών σε θέματα ασφάλειας

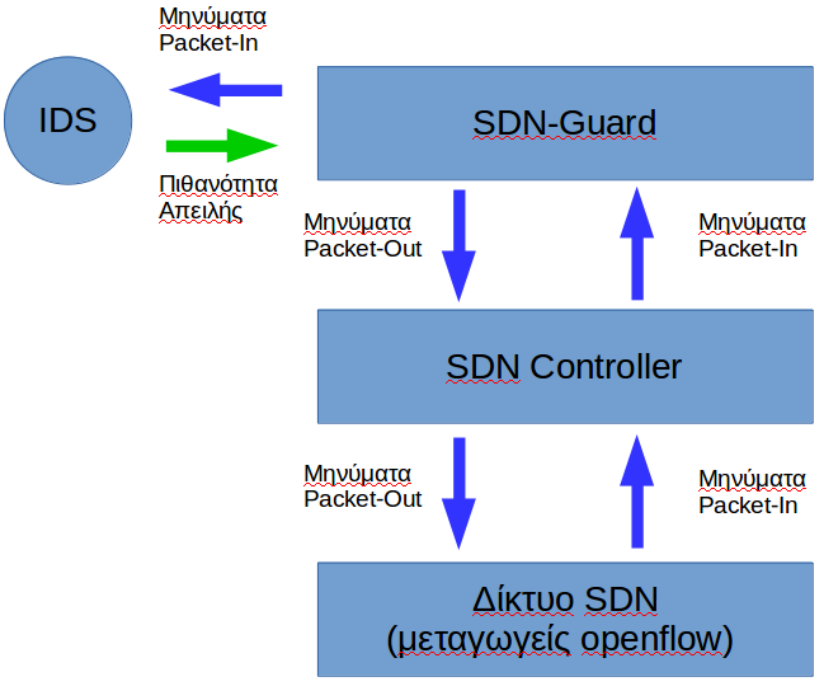
4.1 Ευκολότερη Ανάπτυξη IDPS λόγω καθολικής άποψης του δικτύου (Network-Wide Intrusion Detection Prevention System)

Ο παραδοσιακός τρόπος διαχείρισης κρίσεων σε ένα δίκτυο μέχρι τώρα περιελάμβανε συλλογή δεδομένων καταγραφής σε μορφή κειμένου (logs) από κάθε συσκευή του δικτύου, ανάλυση τους από τους διαχειριστές του δικτύου (άνθρωποι) και στην περίπτωση ανίχνευσης προβλήματος ή κακόβουλων ενεργειών γινόταν λήψη μέτρων αντιμετώπισης (reconfiguration του δικτύου, διακοπή κακόβουλης κίνησης κτλ.). Η διαδικασία αυτή της ανάλυσης αρχείων τόσου όγκου, μπορούν να φτάσουν και πολλά GB, από ανθρώπους είναι διαδικασία χρονοβόρα αν όχι αδύνατη σε περιπτώσεις μεγάλων δικτύων που εξυπηρετούν μεγάλους όγκους κίνησης. Αυτό έδινε σε κακόβουλους εισβολείς άπλετο χρόνο για την διεξαγωγή επιθέσεων πριν αυτές εντοπιστούν. Αλλά ακόμη και η έγκαιρη ανίχνευση προβλημάτων/επιθέσεων δε σημαίνει πάντα και έγκαιρη αντιμετώπιση τους καθώς αυτό εξαρτάται από το είδος της επίθεσης, το εύρος της επίθεσης (για παράδειγμα ο αριθμός των συσκευών που έχουν πληγεί) καθώς και από την ικανότητα και την εμπειρία των διαχειριστών. Η SDN αρχιτεκτονική με την εισαγωγή του μοντέλου κεντρικής εποπτείας και διαχείρισης δικτύου από λογισμικό μπορεί να δώσει τη λύση στο χρόνιο αυτό πρόβλημα των παραδοσιακών δικτύων.

Με τον όρο IDS εννοούμε ένα σύστημα (υλοποιημένο είτε με υλικό είτε με λογισμικό) το οποίο εποπτεύει ένα δίκτυο και είναι επιφορτισμένο με την ανίχνευση ύποπτης κίνησης. Σε

περίπτωση ανίχνευσης τέτοιου είδους κίνησης ειδοποιεί τον διαχειριστή του δικτύου ο οποίος λαμβάνει κατάλληλα μέτρα για την αντιμετώπιση της επίθεσης.

Οι τρόποι χρήσης του IDS σε παραδοσιακά δίκτυα είναι δύο. Συγκεκριμένα, συστήματα βασισμένα σε host (Host-based Intrusion Detection System) στα οποία η εφαρμογή τρέχει σε συγκεκριμένες συσκευές του δικτύου (π.χ. εξυπηρετητές) και συστήματα δικτύου (Network IDS) στα οποία το σύστημα εποπτεύει στρατηγικά σημεία του δικτύου (π.χ. switches, routers). Είναι φανερό, ότι και στους δύο τύπους IDS απαιτούνται συνήθως περισσότερες από μία instances του IDS και ενδεχομένως συνδυασμός και αντιπαραβολή των αποτελεσμάτων τους για επιτυχή ανίχνευση εισβολής. Αυτό μπορεί να οδηγήσει σε λανθασμένα αποτελέσματα (false positive, true negative) και να αυξήσει το χρόνο ανίχνευσης. Την λύση σε αυτό το πρόβλημα φαίνεται να μπορεί να δώσει η κεντρική διαχείριση που προσφέρει το SDN, καθώς ο ελεγκτής του δικτύου έχει πρόσβαση σε όλη την κίνηση (με τη μορφή Packet-In μηνυμάτων). Αντί λοιπόν το ίδιο το σύστημα ανίχνευσης να είναι επιφορτισμένο με τη συλλογή της κίνησης του δικτύου για την ανάλυση, το ρόλο αυτό αναλαμβάνει ο ελεγκτής. Με τον τρόπο αυτό, το σύστημα IDS μπορεί να υλοποιηθεί σαν μία ακόμη SDN εφαρμογή η οποία έχει πρόσβαση στην κίνηση του δικτύου με τη μορφή Packet-In μηνυμάτων. Προχωρώντας ακόμη παραπέρα, η IDS εφαρμογή μπορεί να συνδυαστεί με μία άλλη εφαρμογή SDN η οποία θα έχει την δυνατότητα να επεμβαίνει για την αποτροπή/αντιμετώπιση των επιθέσεων. Ο συνδυασμός αυτός IDS με ένα σύστημα επέμβασης ονομάζεται IDPS (Intrusion Detection Prevention System) και είναι ευκολότερος να υλοποιηθεί με τη βοήθεια της αρχιτεκτονικής SDN. Ένα εξαιρετικό παράδειγμα τέτοιας εφαρμογής είναι το SDN-Guard που παρουσιάζεται στην εργασία των L. Dribi, M. Zhani[5]. Στην εργασία αυτή οι συγγραφείς κάνουν χρήση ενός ήδη υπάρχοντος IDS συστήματος το οποίο βρίσκεται σε συνεχή επικοινωνία με την εφαρμογή SDN που ανέπτυξαν. Η εφαρμογή επαναπροωθεί όλα τα Packet-In μηνύματα στο σύστημα IDS και το τελευταίο τροφοδοτεί την εφαρμογή SDN-Guard με μια “πιθανότητα απειλής” για κάθε ροή της κίνησης. Εάν αυτή η πιθανότητα είναι πάνω από ένα συγκεκριμένο κατώφλι (threshold) τότε η ροή αυτή κρίνεται ως απειλή και το SDN-Guard αναλαμβάνει να την αντιμετωπίσει.



Εικόνα 8 : Υλοποίηση εφαρμογής εντοπισμού απειλών, SDN-Guard

Οι συγγραφείς παρουσιάζουν τρεις τρόπους με τους οποίους αντιμετωπίζονται οι απειλές με την εφαρμογή

1. Δρομολόγηση βασισμένη σε απειλές

Η εφαρμογή επιλέγει να ανακατανείμει τις πιθανά κακόβουλες ροές δικτύου σε συνδέσμους που υπο-χρησιμοποιούνται. Γίνεται δηλαδή load-balancing κι έτσι αποφεύγονται οι καθυστερήσεις σε ήδη υπερφορτωμένους συνδέσμους. Έτσι επιτυγχάνεται μείωση του αντίκτυπου μιας πιθανής επίθεσης στις επιδόσεις του δικτύου.

2. Επιλογή κατάλληλων χρόνων timeout για τους κανόνες ροής

Η επιλογή σύντομων χρόνων λήξης (timeout) για τις ροές σημαίνει ότι οι μεταγωγείς θα επικοινωνούν πολύ συχνότερα με τον ελεγκτή για να ανανεώσουν τους πίνακές τους. Αυτό πρακτικά μπορεί να οδηγήσει σε υπερβολική κίνηση και υπερφόρτωση των καναλιών μεταγωγέα - ελεγκτή. Για τον λόγο αυτό, οι συγγραφείς επιλέγουν τους χρόνους λήξης κάθε ροής με βάση την πιθανότητα απειλής. Όσο μεγαλύτερη είναι η πιθανότητα να πρόκειται για απειλή τόσο μεγαλύτερος κι ο χρόνος λήξης της ώστε η συγκεκριμένη ροή να μην μπορεί να δημιουργήσει υπερφόρτωση μεταξύ μεταγωγέα και ελεγκτή.

3. Συσσωμάτωση (grouping) κακόβουλων κανόνων ροής

Επειδή οι κακόβουλες ροές μένουν για περισσότερη ώρα στους πίνακες (λόγω υψηλού timeout) υπάρχει ο κίνδυνος υπερφόρτωσης των πινάκων ενός μεταγωγέα. Για την αποφυγή του ενδεχομένου αυτού, η εφαρμογή αναλαμβάνει να ομαδοποιήσει τις πιθανές κακόβουλες ροές με βάση κάποια στοιχεία (π.χ. κοινή διεύθυνση IP αποστολέα και παραλήπτη). Έτσι, επιτυγχάνεται μείωση των καταχωρίσεων στους πίνακες ροής.

4.2 Ενσωμάτωση Τεχνητής Νοημοσύνης

Η αρχιτεκτονική SDN θεωρείται ελκυστική για την ενσωμάτωση μοντέλων τεχνητής νοημοσύνης (στον τομέα των δικτύων με τον όρο "τεχνητή νοημοσύνη" αναφερόμαστε κυρίως σε Machine Learning τεχνικές). Η κεντρική συλλογή των δεδομένων (και η καθολική άποψη του δικτύου) προσφέρει τη δυνατότητα τροφοδότησης των μοντέλων (π.χ. νευρωνικών δικτύων) με τα δεδομένα αυτά με στόχο την ανάπτυξη μια ευρείας γκάμας τεχνικών ανίχνευσης και αντιμετώπισης των απειλών. Κάποια παραδείγματα που αναφέρονται στη βιβλιογραφία [6] είναι:

1. Κατηγοριοποίηση Κίνησης (Traffic Classification). Η κίνηση κατηγοριοποιείται όχι μόνο ανάλογα με τη θύρα προορισμού όπως γινόταν παλιότερα αλλά έξυπνα με βάση τα χαρακτηριστικά της κίνησης. Αυτό κάνει τον εντοπισμό ύποπτης κίνησης ευκολότερο.

2. Ανίχνευση Ανωμαλιών (Anomaly Detection). Με αυτό τον όρο αναφερόμαστε κυρίως σε ανίχνευση ύποπτων ή γενικά σπάνιων συμπεριφορών. Στα δίκτυα αυτό μπορεί να σημαίνει προσπάθεια εισβολής κτλ. Η τεχνική αυτή χρησιμοποιείται για την ανάπτυξη έξυπνων IDS που ονομάζονται Anomaly-based IDS.

3. Δρομολόγηση βασισμένη σε Machine Learning (ML-based multipath flow routing). Εξυπνότερη προώθηση και δρομολόγηση των πακέτων η οποία θα μπορούσε να μετριάσει DoS επιθέσεις μέσω load balancing.

4.3 Δυνατότητα αυτο-ίασης

Τα δίκτυα SDN δίνουν τη δυνατότητα εγκατάστασης μηχανισμών αυτο-ίασης. Ίσως το πιο χαρακτηριστικό παράδειγμα ενός τέτοιου μηχανισμού είναι οι κανόνες υπό προϋποθέσεις (conditional rules) [2]. Οι κανόνες αυτοί εγκαθίστανται στους μεταγωγείς από το πεδίο ελέγχου και φροντίζουν να λαμβάνουν κάποια μέτρα αντιμετώπισης εάν παρατηρηθεί επικίνδυνη συμπεριφορά. Συνήθως απαιτούν και τη λήψη στατιστικών από τον μεταγωγέα. Ένα παράδειγμα θα μπορούσε να είναι η άρνηση πακέτων (drop) από κάποιον αποστολέα που στέλνει μεγάλο όγκο (πιθανή επίθεση) ή η προώθηση τους σε κάποιο honeypot ώστε να συλλεχθούν πληροφορίες σε σχέση με τον επιτιθέμενο και την επίθεση.

4.4 Αυξημένη Δυνατότητα Ελέγχου

Το σύστημα προώθησης πακέτων βασισμένο σε ροές (flow-based scheme) που χαρακτηρίζει το SDN δίνει περισσότερες δυνατότητες ελέγχου της κίνησης στο δίκτυο από τα παραδοσιακά δίκτυα όπου η προώθηση/δρομολόγηση γίνεται απλά με βάση τη διεύθυνση IP προορισμού. Στο flow-based scheme μπορεί να γίνει χρήση πολλαπλών πεδίων της επικεφαλίδας του πακέτου για τη λήψη αποφάσεων. Αυτό δίνει συγκριτικό πλεονέκτημα στα δίκτυα SDN για καλύτερο έλεγχο της κίνησης του δικτύου και προστασία από επιθέσεις.

4.5 Ταχύτερη και δυναμική υλοποίηση συστημάτων ασφαλείας στο δίκτυο

Ένας από τους βασικούς λόγους που η SDN αρχιτεκτονική τράβηξε το ενδιαφέρον της ερευνητικής κοινότητας ήταν το γεγονός ότι ο διαχωρισμός των πεδίων ελέγχου και δεδομένων έδωσε τη δυνατότητα ανάπτυξης και δοκιμής εφαρμογών και πρωτοκόλλων δικτύου εύκολα, γρήγορα αλλά κυρίως χρησιμοποιώντας τις ήδη υπάρχουσες υποδομές δικτύου χωρίς την ανάγκη δημιουργίας ξεχωριστών testbed. Για παράδειγμα οι ερευνητές ενός πανεπιστημίου μπορούν να κάνουν χρήση του δικτύου του πανεπιστημίου για τις δοκιμές καινούριων πρωτοκόλλων χωρίς να επηρεάζονται οι υπόλοιποι χρήστες του δικτύου. Ακόμη σημαντικότερο είναι το γεγονός ότι μετά την υλοποίηση και την επαλήθευση της σωστής λειτουργίας του νέου πρωτοκόλλου ασφαλείας, το τελευταίο μπορεί πολύ ευκολότερα να ενσωματωθεί στο υπάρχον δίκτυο χωρίς την ανάγκη ενημέρωσης δεκάδων ή και χιλιάδων συσκευών δικτύου όπως στα παραδοσιακά δίκτυα. Επίσης, νέες εφαρμογές και

πρωτόκολλα ασφαλείας μπορεί να ενεργοποιούνται on-demand ανάλογα με την κατάσταση του δικτύου.

4.6 Σχεδιασμός με προτεραιότητα την ασφάλεια

Είναι κοινή παραδοχή ότι τα παραδοσιακά Ethernet δίκτυα δεν λάμβαναν υπόψη τους το θέμα της ασφάλειας στην αρχή της ανάπτυξής τους. Αυτό το θέμα άρχισε να απασχολεί τους ερευνητές, τους διαχειριστές και τους χρήστες πολύ αργότερα όταν τα δίκτυα άρχισαν να χρησιμοποιούνται για τη μεταφορά ευαίσθητων δεδομένων γεγονός που τράβηξε το ενδιαφέρον κακόβουλων εισβολέων. Συνεπώς, μπορούμε να πούμε ότι τα αναπτυσσόμενα πρωτόκολλα και συστήματα ασφαλείας λειτουργούν κατά κάποιο τρόπο σαν add-ons ή patches για τα μακροχρόνια κενά ασφαλείας των παραδοσιακών δικτύων. Το γεγονός ότι η αρχιτεκτονική SDN είναι σχετικά καινούρια μπορεί να αποτελεί πλεονέκτημα έναντι παραδοσιακών αρχιτεκτονικών δικτύων αρκεί να αποφευχθούν τα λάθη του παρελθόντος. Οι ειδικοί έχουν πια την πολυτέλεια να σχεδιάσουν τη νέα αρχιτεκτονική με την ασφάλεια όχι σαν επιπρόσθετο αλλά σαν βασική προτεραιότητα χρησιμοποιώντας τη γνώση και την εμπειρία πολλών ετών.

4.7 Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάσαμε τα κυριότερα πλεονεκτήματα της αρχιτεκτονικής SDN έναντι των παραδοσιακών δικτύων σε σχέση με την ασφάλεια. Τα περισσότερα από αυτά οφείλονται στην κεντρική διαχείριση του δικτύου και στην καθολική άποψη της τοπολογίας του δικτύου από τον ελεγκτή. Παρόλα αυτά, αυτή η κεντρική διαχείριση του δικτύου είναι και το πιο τρωτό σημείο της αρχιτεκτονικής καθώς εισάγει ένα Μοναδικό Σημείο Αποτυχίας. Έτσι, η ερώτηση εάν τα δίκτυα SDN είναι ασφαλέστερα από τα παραδοσιακά είναι δύσκολο να απαντηθεί χωρίς να ληφθούν υπόψη πολλοί και διαφορετικοί παράγοντες. Για παράδειγμα, ένα παραδοσιακό δίκτυο καλά παραμετροποιημένο, που κάνει χρήση ισχυρής κρυπτογράφησης και διατηρείται ενημερωμένο με τα τελευταία patches ασφαλείας μπορεί να είναι πολύ πιο ασφαλές από ένα δίκτυο SDN που αμελεί τα παραπάνω. Το βασικότερο, ίσως, πλεονέκτημα του SDN είναι ότι είναι ακόμα υπό ανάπτυξη κι έτσι δίνει τη δυνατότητα στους μηχανικούς να σχεδιάσουν την αρχιτεκτονική με την ασφάλεια σαν προαπαιτούμενο και σαν βασικό συστατικό της αρχιτεκτονικής κι όχι σαν ένα ακόμα patch που προσπαθεί να κλείσει τρύπες. Αν τα λάθη του παρελθόντος δεν επαναληφθούν η αρχιτεκτονική SDN μπορεί να αποτελέσει επανάσταση και στον τομέα της ασφάλειας.

Κεφάλαιο 5
Υλοποίηση Προσομοίωσης

5.1Παρουσίαση, Ανάλυση και Εγκατάσταση των Εργαλείων

Virtualbox

Το Virtualbox είναι ένας υπερεπόπτης (hypervisor) ο οποίος χρησιμοποιείται για τη δημιούργια εικονικών μηχανών (virtual machines) τα οποία προσομοιώνουν ένα ολόκληρο λειτουργικό σύστημα. Για παράδειγμα μπορούμε να τρέξουμε ένα εικονικό περιβάλλον linux το οποίο ονομάζεται φιλοξενούμενο λειτουργικό (guest OS) πάνω από το φυσικό μας λειτουργικό (αυτό το οποίο είναι εγκατεστημένο στο υλικό του υπολογιστή μας για παράδειγμα windows) το οποίο ονομάζεται λειτουργικό βάσης (host OS). Με αυτό τον τρόπο μπορούμε να δοκιμάσουμε εφαρμογές και συστήματα υλοποιημένα σε συστήματα διαφορετικά από το υπάρχον λειτουργικό μας χωρίς να χρειάζεται περίπλοκη εγκατάσταση. Επίσης, μπορούμε να τρέχουμε ταυτόχρονα πλήθος εικονικών μηχανημάτων (ο αριθμός των οποίων περιορίζεται φυσικά από τους πόρους του μηχανήματος μας) ώστε να μπορούμε να προσομοιώσουμε μεγαλύτερα συστήματα (π.χ. ένα δίκτυο). Διαθέτει γραφικό περιβάλλον αλλά και γραμμή εντολών για τη δημιουργία και τη διαχείριση των εικονικών μηχανών. Η επιλογή του έγινε για δύο βασικούς λόγους: κατά πρώτον γιατί η διατήρηση αρχείου backup της δουλειάς μας είναι πολύ ευκολότερη και δεν κινδυνεύουμε να χάσουμε πολύτιμο χρόνο σε περίπτωση βλάβης του υπολογιστή μας αφού το εικονικό μηχάνημα μπορεί εύκολα να αντιγραφεί και να τρέξει σε άλλο φυσικό μηχάνημα. Επιπλέον, η υλοποίηση μας απαιτεί την εγκατάσταση πολλών εργαλείων κι έτσι δεν κινδυνεύουμε να βλάψουμε το φυσικό μας σύστημα.

Εγκατάσταση του Virtualbox

Για την εγκατάσταση της τελευταίας έκδοσης του virtualbox σε Ubuntu 18.04 χρειάζεται να προσθέσουμε το αποθετήριο «Multiverse» καθώς δε βρίσκεται στα επίσημα αποθετήρια. Αρκεί η παρακάτω εντολή:

$ sudo add-apt-repository multiverse && sudo apt-get update

και στη συνέχεια την εντολή για την εγκατάσταση:

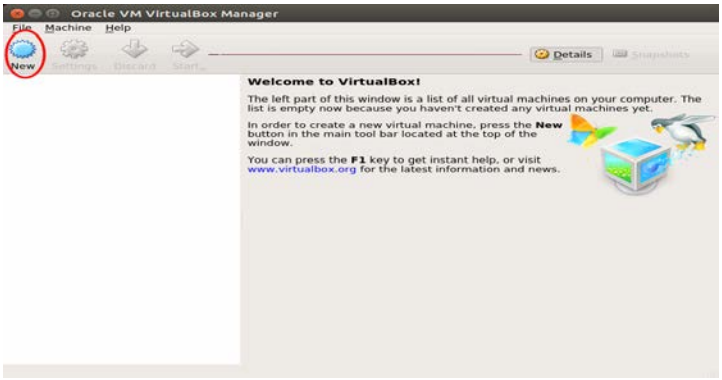
$ sudo apt install virtualbox

Τέλος, για να εκκινήσουμε τον hypervisor εκτελούμε απλά:

$ virtualbox

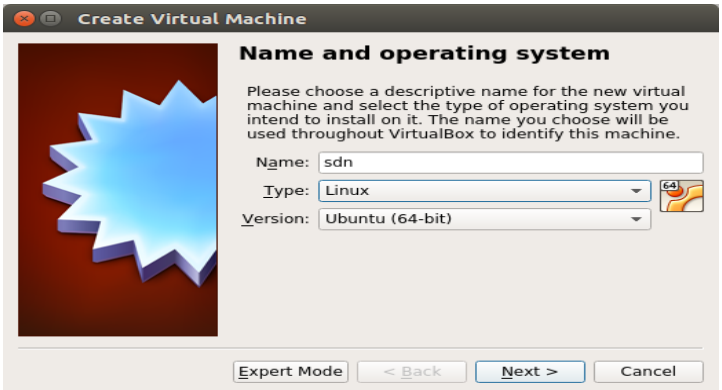
Εγκατάσταση εικονικού μηχανήματος στο Virtualbox

Βήμα 1. Από την υποδοχή του Virtualbox επιλέγουμε 'New' για τη δημιουργία νέου εικονικού μηχανήματος



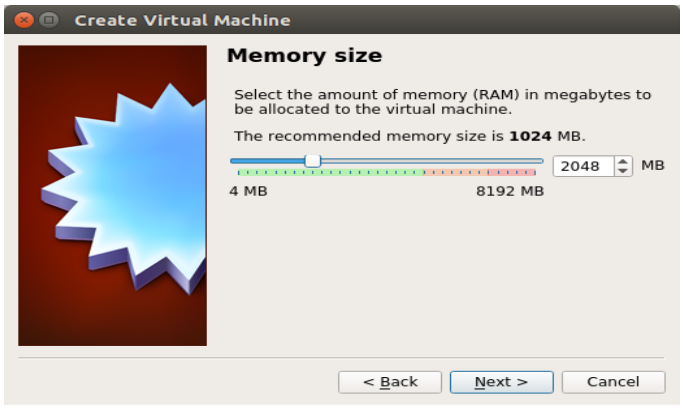
Εικόνα 9 : Εγκατάσταση εικονικού μηχανήματος Βήμα 1

Βήμα 2. Επιλέγουμε όνομα του μηχανήματος καθώς και τύπο λειτουργικού και διανομή. Στη δική μας περίπτωση Linux και διανομή Ubuntu 64-bit



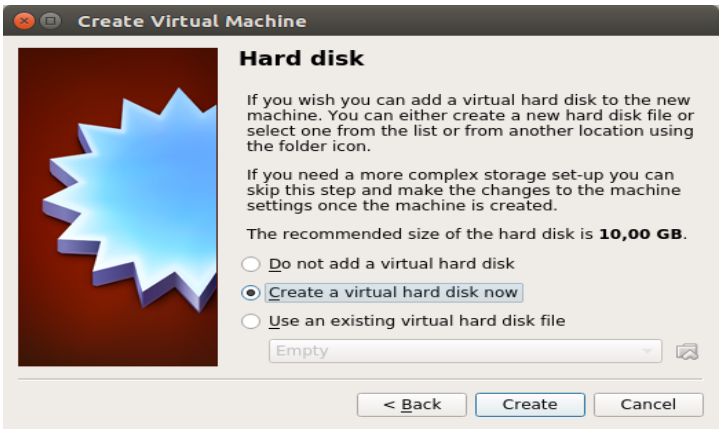
Εικόνα 10 : Εγκατάσταση εικονικού μηχανήματος Βήμα 2

Βήμα 3. Επιλογή μεγέθους μνήμης RAM (στην περίπτωση μας 2GB)



Εικόνα 11 : Εγκατάσταση εικονικού μηχανήματος Βήμα 3

Βήμα 4. Επιλέγουμε δημιουργία εικονικού σκληρού δίσκου



Εικόνα 12 : Εγκατάσταση εικονικού μηχανήματος Βήμα 4

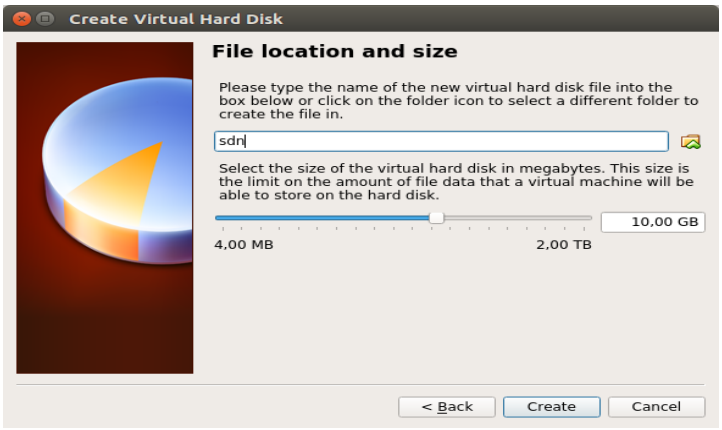
Βήμα 5. Επιλέγουμε τύπου σκληρού δίσκου VDI (το native format του Virtualbox)

Εικόνα 13 : Εγκατάσταση εικονικού μηχανήματος Βήμα 5

Βήμα 6. Επιλογή δυναμικά κατανεμημένου σκληρού δίσκου

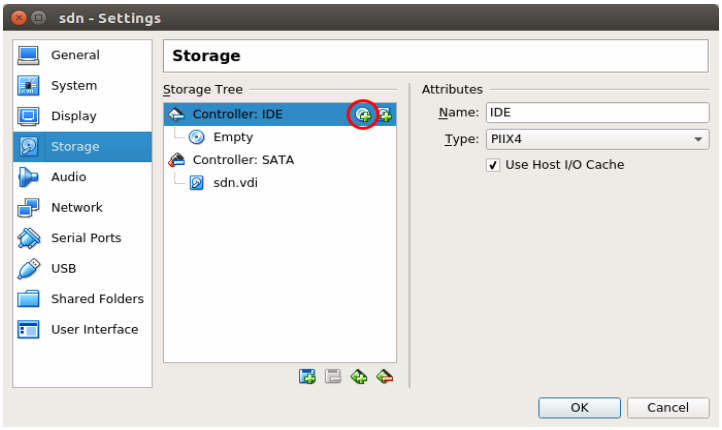
Εικόνα 14 : Εγκατάσταση εικονικού μηχανήματος Βήμα 6

Βήμα 7. Επιλέγουμε όνομα και χωρητικότητα για τον εικονικό δίσκο μας (στην περίπτωσή μας 20GB)



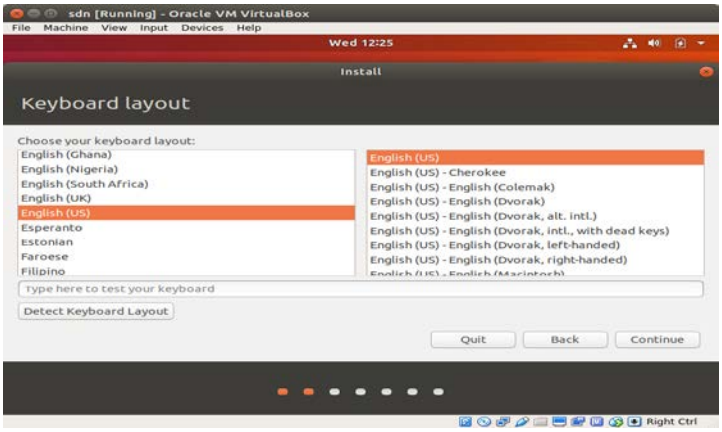
Εικόνα 15 : Εγκατάσταση εικονικού μηχανήματος Βήμα 7

Βήμα 8. Κάνουμε δεξί κλικ στο εικονίδιο του νέου VM κι επιλέγουμε ‘Settings’ κι από εκεί ‘Storage’. Στη συνέχεια επιλέγουμε ‘Add Optical Drive’ (εικόνα) και διαλέγουμε το iso του Ubuntu 18.04.2 LTS (το οποίο έχουμε κατεβάσει εδώ: https://www.ubuntu.com/download/desktop)



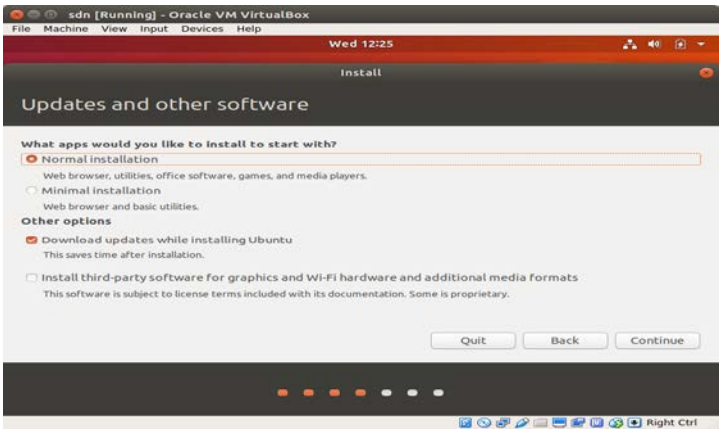
Εικόνα 16 : Εγκατάσταση εικονικού μηχανήματος Βήμα 8

Βήμα 9. Επιλέγουμε γλώσσα και διάταξη πληκτρολογίου



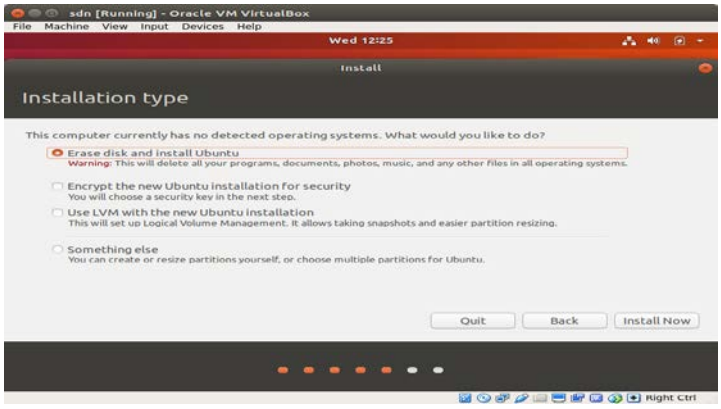
Εικόνα 17 : Εγκατάσταση εικονικού μηχανήματος Βήμα 9

Βήμα 10. Διαλέγουμε 'Κανονική Εγκατάσταση' κι επιλέγουμε εγκατάσταση των ενημερώσεων μαζί με την εγκατάσταση

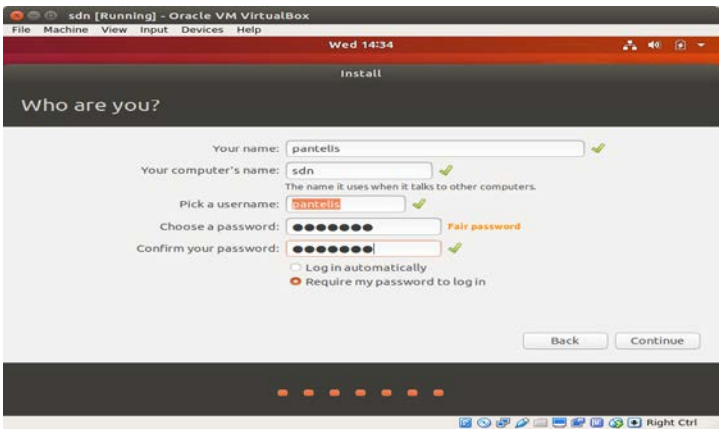


Εικόνα 18 : Εγκατάσταση εικονικού μηχανήματος Βήμα 10

Βήμα 11. Τέλος, επιλέγουμε 'Διαγραφή ολόκληρου του δίσκου και εγκατάσταση Ubuntu' (εννοώντας φυσικά τον εικονικό μας δίσκο). Κάνουμε κλικ στην επιλογή 'Install Now' και περιμένουμε (περίπου 15 λεπτά). Εισάγουμε όνομα χρήστη και όνομα υπολογιστή όταν μας ζητηθούν και η εγκατάσταση ολοκληρώνεται.



Εικόνα 19 : Εγκατάσταση εικονικού μηχανήματος Βήμα 11



Εικόνα 20 : Εγκατάσταση εικονικού μηχανήματος Βήμα 12

Mininet

Το Mininet είναι ένας προσομοιωτής δικτύου που χρησιμοποιείται ευρέως για λόγους εκπαιδευτικούς και ερευνητικούς πάνω στα SDN δίκτυα. Δίνει τη δυνατότητα δημιουργίας εικονικών τερματικών μηχανών (hosts), μεταγωγέων (switches), ελεγκτών (controllers) και συνδέσμων (links) και υποστηρίζει το ευρέως διαδεδομένο πρωτόκολλο OpenFlow. Το μεγάλο του πλεονέκτημα είναι ότι οι συσκευές δικτύου που προσομοιώνει το mininet «τρέχουν» πραγματικό λειτουργικό linux. Αυτό πρακτικά σημαίνει ότι μπορούμε να εγκαταστήσουμε και να δοκιμάσουμε οποιαδήποτε εφαρμογή linux στους hosts του mininet σαν αυτές να έτρεχαν σε φυσικό linux περιβάλλον. Έτσι, αφότου μια εφαρμογή δοκιμαστεί στον προσομοιωτή και ελεγχθεί ότι δουλεύει σωστά μπορεί στη συνέχεια να μεταφερθεί «ως έχει» σε ένα πραγματικό linux περιβάλλον. Αν και έχει πολλές ομοιότητες με ένα fully virtualized σύστημα προσφέρει και κάποια πλεονεκτήματα έναντι του τελευταίου όπως ταχύτερο boot (δευτερόλεπτα αντί για λεπτά), μεγαλύτερη επεκτασιμότητα (δυνατότητα δημιουργίας εκατοντάδων εικονικών συσκευών πράγμα που θα απαιτούσε τεράστια resources σε fully virtualized σύστημα), περισσότερο bandwidth και ευκολότερη εγκατάσταση. Είναι προφανές ότι είναι και πολύ πιο εύχρηστο από ένα testbed με πραγματικό υλικό καθώς είναι πολύ πιο φτηνό από το πραγματικό υλικό και ευκολότερα παραμετροποιήσιμο.

Εγκατάσταση

Η εγκατάσταση του mininet είναι απλούστατη καθώς το πακέτο βρίσκεται στα επίσημα αποθετήρια του Ubuntu. Έτσι αρκεί η εντολή:

$ sudo apt-get install mininet

μέσα στο εικονικό μας μηχάνημα για να εγκατασταθεί. Υπάρχει, επίσης, η δυνατότητα εγκατάστασης ενός προκατασκευασμένου εικονικού μηχανήματος αλλά εμείς επιλέξαμε την πρώτη επιλογή καθώς θα χρησιμοποιήσουμε το δικό μας εικονικό μηχάνημα Ubuntu.

hping3

Το hping3 είναι ένας αναλυτής πακέτων δικτύου αλλά και γεννήτρια πακέτων που χρησιμοποιείται ευρέως για penetration testing δικτύων. Μερικές από τις δυνατότητες του hping3 είναι:

Τεστ για firewalls

Σκανάρισμα θυρών δικτύου (port scanning)

Δυνατότητα αναγνώρισης λειτουργικού ενός host (OS fingerprinting)

Δυνατότητα καθορισμού χρόνου λειτουργίας ενός host (uptime guessing)

Έλεγχος στοίβας TCP/IP (TCP/IP stack auditing)

Και πολλές άλλες λειτουργίες όχι απαραίτητα χρήσιμες μόνο στον τομέα της ασφάλειας όπως traceroute, ανακάλυψη MTU ενός συνδέσμου, κανονική ping λειτουργία και πολλά άλλα.

Η δυνατότητα του εργαλείου να στέλνει πακέτα σε μεγάλο όγκο και με συγκεκριμένα πεδία στην TCP επικεφαλίδα (TCP flags) το καθιστά ιδανικό για τη διεξαγωγή διαφορετικών τύπων DoS.

Συγκεκριμένα, το hping3 μπορεί να χρησιμοποιηθεί για:

1. Επιθέσεις στο επίπεδο εφαρμογών. Για παράδειγμα:

- Αργή HTTP σύνδεση
- Πλημμύρες (flood) HTTP POST και GET
- SIP invite πλημμύρα

2. Επιθέσεις πρωτοκόλλου

- SYN, ACK, RST, TCP floods, Land επίθεση
- TCL state exhaustion επίθεση, TCP window size
- Ping of Death

3. Επιθέσεις Όγκου (volumetric/bandwidth attack)

- ICMP flood
- UDP flood και άλλες

Για την εγκατάσταση του αρκεί η εντολή:
$ sudo apt-get install hping3

tcpdump

Το tcpdump είναι ένας διαδεδομένος αναλυτής πακέτου που τρέχει αποκλειστικά σε κονσόλα χωρίς να χρειάζεται γραφικό περιβάλλον, όπως για παράδειγμα το Wireshark. Αυτό τον καθιστά εξαιρετικά χρήσιμο για ανάλυση της κίνησης δικτύου σε συστήματα που διαθέτουν μόνο τερματικό όπως για παράδειγμα οι servers. Στη συγκεκριμένη εργασία χρησιμοποιείται ο tcpdump αντί για το Wireshark καθώς θέλουμε να αναλύσουμε την κίνηση σε hosts του mininet οι οποίοι δε διαθέτουν γραφικό περιβάλλον. Στις περισσότερες διανομές Linux ο tcpdump είναι προεγκατεστημένος.
Για την απλή εκκίνησή του αρκεί η εντολή:
tcpdump -n -i <όνομα κάρτας δικτύου προς ανάλυση>

Η εντολή αυτή δείχνει σε πραγματικό χρόνο όλα τα εισερχόμενα και εξερχόμενα πακέτα από τη συγκεκριμένη κάρτα δικτύου. Μπορούμε φυσικά με τη χρήση φίλτρων να εξαιρέσουμε συγκεκριμένα πακέτα ή να εμφανίσουμε μόνο πακέτα με συγκεκριμένα χαρακτηριστικά (πρωτόκολλο, IP προέλευσης/προορισμού, θύρα προέλευσης/προορισμού κ.α.).

iperf3

Το iperf3 είναι ένα ανοιχτού κώδικα εργαλείο το οποίο επιτρέπει μετρήσεις ταχύτητας (throughput) μιας σύνδεσης μεταξύ ενός πελάτη κι ενός εξυπηρετητή. Στην εργασία αυτή το χρησιμοποιούμε για να προσομοιώσουμε μια σύνδεση πελάτη-εξυπηρετητή με συγκεκριμένο bandwidth και για να δείξουμε πώς αυτή επηρεάζεται από μια DDoS επίθεση στον εξυπηρετητή. Αρκεί η εντολή
$ sudo apt-get install iperf3
για την εγκατάστασή του.

ONOS SDN controller

Επιλέξαμε να χρησιμοποιήσουμε τον ελεγκτή ONOS καθώς διαθέτει ένα ιδιαίτερα εύχρηστο γραφικό περιβάλλον (GUI) για την οπτικοποίηση και συνεπώς την καλύτερη κατανόηση της τοπολογίας του δικτύου μας. Ο ONOS είναι ένας ανοιχτού κώδικα controller γραμμένος κυρίως σε Java του οποίου οι developers υποστηρίζουν ότι προσφέρει κάποια πλεονεκτήματα έναντι άλλων ελεγκτών όπως επεκτασιμότητα, υψηλές επιδόσεις, ελαστικότητα και υποστήριξη πολλών legacy αλλά και next generation συσκευών.

Εγκατάσταση

Ο Onos έχει πολλές εξαρτήσεις από άλλα πακέτα για την εγκατάσταση του. Τα παρακάτω βρίσκονται στα επίσημα αποθετήρια του Ubuntu συνεπώς μπορούν να εγκατασταθούν απλά με apt-get install:

- git
- zip
- curl
- python

Απαιτείται, επίσης, εγκατάσταση της Java ως εξής:

```
sudo apt-get install software-properties-common -y && \
sudo add-apt-repository ppa:webupd8team/java -y && \
sudo apt-get update && \
echo "oracle-java8-installer shared/accepted-oracle-license-v1-1 select true" | sudo debconf-set-selections && sudo apt-get install oracle-java8-installer oracle-java8-set-default -y
```

Για το build (δημιουργία ενός εκτελέσιμου αρχείο από πηγαίο κώδικα) του Onos απαιτείται το Bazel, ένα εργαλείο της Google που χρησιμοποιείται όλο και περισσότερο στη βιομηχανία. Εκτελούμε:

sudo apt-get install pkg-config zip g++ zlib1g-dev unzip python

για την εγκατάσταση των εξαρτήσεων και στη συνέχεια κατεβάζουμε το script εγκατάστασης από https://github.com/bazelbuild/bazel/releases

Στη συνέχεια:
chmod +x bazel-0.24.1-installer-linux-x86_64.sh

για να κάνουμε το script εκτελέσιμο και εκτέλεσή του με (όπου –user το όνομα χρήστη μας)
./bazel-0.24.1-installer-linux-x86_64.sh --user

Αφού εγκαταστήσουμε το bazel μπορούμε να προχωρήσουμε με την εγκατάσταση του Onos αφού πρώτα το κατεβάσουμε με:
```
$ git clone https://gerrit.onosproject.org/onos
```

Στη συνέχεια:
$ cd onos
και
$ bazel build onos
Μόλις ολοκληρωθεί η εγκατάσταση μπορούμε να τρέξουμε τον ελεγκτή με:
```
$ bazel run onos-local -- clean debug
```

sFlow-RT
Το sFlow-RT είναι ένα εργαλείο το οποίο χρησιμοποιεί το πρωτόκολλο sFlow για να συλλέξει δεδομένα από το δίκτυο και να τα αναλύσει για την εξαγωγή χρήσιμων συμπερασμάτων. Αυτού του είδους τα εργαλεία αποκαλούνται Real-Time Analytics και στην περίπτωσή μας θα το χρησιμοποιήσουμε για να ανιχνεύσουμε κακόβουλη κίνηση κατά την προσομοίωση μιας DoS επίθεσης σε ένα SDN δίκτυο. Το sFlow-RT σε συνεργασία με τον ελεγκτή του δικτύου μας (ONOS) θα μας βοηθήσουν να μετριάσουμε την επίθεση. Η εγκατάστασή του γίνεται ως εξής:
Κατεβάζουμε το συμπιεσμένο πακέτο με:
$ wget https://inmon.com/products/sFlow-RT/sflow-rt.tar.gz
το αποσυμπιέζουμε:
$ tar -xvzf sflow-rt.tar.gz
και εκκινούμε το πρόγραμμα με:
$ cd sflow-rt
$ ./start.sh

Μπορούμε, επίσης, να εγκαταστήσουμε και μια επιπλέον εφαρμογή που μας επιτρέπει να οπτικοποιήσουμε τα δεδομένα που συλλέγουμε από το mininet δίκτυο μας όπως για

παράδειγμα την κίνηση κάθε switch, την τοπολογία κτλ. Η εφαρμογή ονομάζεται mininet-dashboard και εγκαθίσταται με:

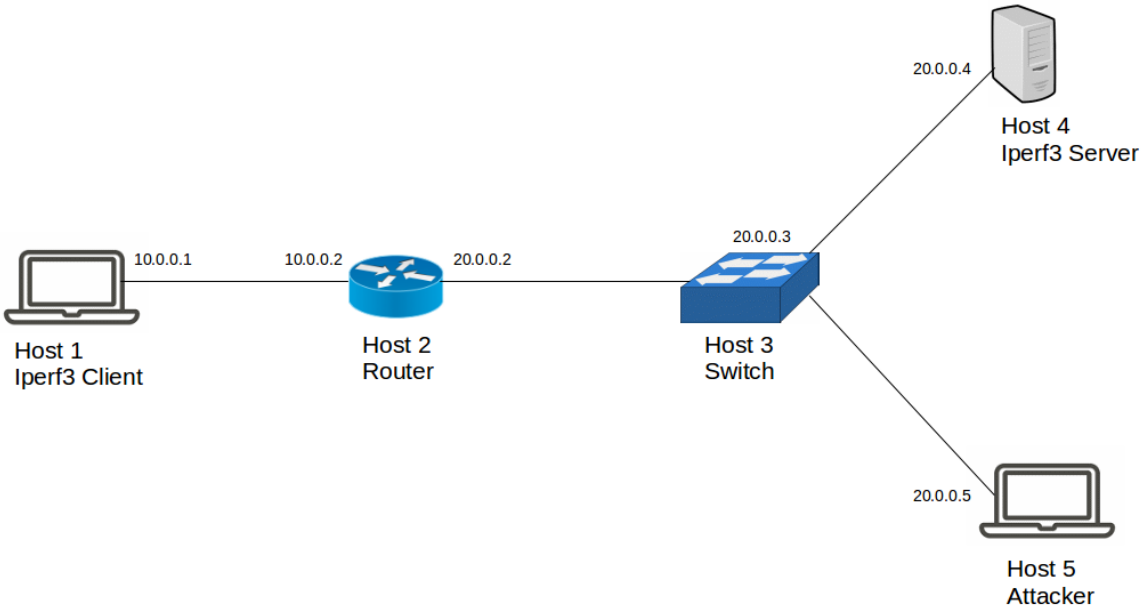
$ cd sflow-rt

$ get-app.sh sflow-rt mininet-dashboard

5.2 DoS Επίθεση σε Παραδοσιακό Ethernet δίκτυο

5.2.1 Περιγραφή μοντέλου

Παρόλο που το mininet χρησιμοποιείται κυρίως για την προσομοίωση SDN δικτύων μπορεί εύκολα να χρησιμοποιηθεί και για παραδοσιακά Ethernet δίκτυα αφού δίνεται η δυνατότητα δημιουργίας κανονικών 'linux-bridges' (δηλαδή εικονικοί μεταγωγείς) κι όχι μόνο OVS switches. Επίσης, ένας κανονικός host του mininet μπορεί να παραμετροποιηθεί ώστε να λειτουργεί σαν router ή switch. Το παρακάτω python script που συντάξαμε δημιουργεί την τοπολογία δικτύου που φαίνεται στην παρακάτω εικόνα. Να σημειωθεί ότι παρόλο που χρησιμοποιούνται τα εικονίδια δρομολογητή και μεταγωγέα στην εικόνα, στην πραγματικότητα πρόκειται για κανονικούς hosts του mininet που έχουν παραμετροποιηθεί έτσι ώστε να παίξουν τους συγκεκριμένους ρόλους. Ο πηγαίος κώδικας δίνεται σε πράσινη γραμματοσειρά και τα επεξηγηματικά σχόλια σε μαύρη.



Εικόνα 21 :Τοπολογία του παραδοσιακού δικτύου

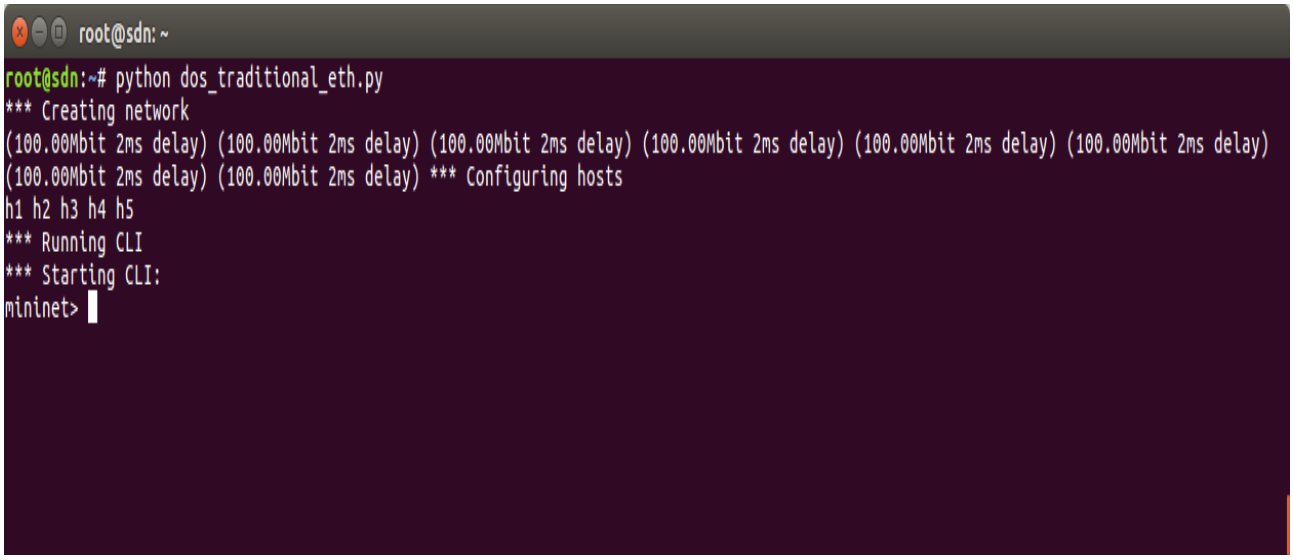
```
#!/usr/bin/python

# Εισαγωγή απαραίτητων python πακέτων
import re
import sys
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.log import setLogLevel, info, error
from mininet.link import Link, TCLink, Intf
from mininet.topolib import TreeTopo
from mininet.util import quietRun
from mininet.nodelib import LinuxBridge

def topology():
    # Δημιουργία του object 'Δίκτυο'
    net = Mininet(link=TCLink)

    # Δημιουργία των hosts
    print "*** Creating nodes"

    # Δημιουργία του host1. Δίνεται η δυνατότητα παραμετροποίησης της
    # MAC καθώς και της IP. Ο host αυτός θα παίξει το ρόλο του πελάτη iperf3
    # ο οποίος θα συνδεθεί στον εξυπηρετητή iperf3 (host4)
    h1 = net.addHost('h1', mac='00:00:00:00:01:00', ip='10.0.0.1/24')

    # Δημιουργία του host2. Ο host αυτός θα παίξει το ρόλο του δρομολογητή με κατάλληλη
    # παραμετροποίηση που εξηγείται παρακάτω
    h2 = net.addHost('h2', mac='00:00:00:00:02:00', ip='10.0.0.2/24')

    # Δημιουργία του host3. Ο host αυτός θα παίξει το ρόλο του μεταγωγέα με κατάλληλη
    # παραμετροποίηση που εξηγείται παρακάτω
    h3 = net.addHost('h3', mac='00:00:00:00:03:00', ip='20.0.0.3/24')
```

```
# Δημιουργία του host4. Ο host αυτός θα παίξει το ρόλο του
# εξυπηρετητή iperf3
h4 = net.addHost('h4', mac='00:00:00:00:04:00', ip='20.0.0.4/24')

# Δημιουργία του host5. Ο host αυτός θα παίξει το ρόλο του κακόβουλου εισβολέα
h5 = net.addHost('h5', mac='00:00:00:00:05:00', ip='20.0.0.5/24')

print "*** Creating links"
# Δημιουργία των συνδέσεων. Η σύνταξη
# addLink(h1, h2,0,0,cls=TCLink,bw=100,delay='2ms') σημαίνει ότι η κάρτα δικτύου
# eth0 του h1 θα συνδεθεί με το eth0 του host2. TCLink είναι ο τύπος του συνδέσμου
# με τον οποίο μας δίνεται η δυνατότα να κάνουμε traffic control. Στην περίπτωση μας
# ορίζουμε το maximum bandwidth κάθε συνδέσμου στα 100Mbps και το delay στα 2ms.
# Οι τιμές αυτές μας δίνουν ένα πιο ρεαλιστικό περιβάλλον (από τα 40Gbps που έχουμε
# για παράδειγμα αφού η προσομοίωση τρέχει στον localhost)
net.addLink(h1, h2,0,0,cls=TCLink,bw=100,delay='2ms')
net.addLink(h2, h3,1,0,cls=TCLink,bw=100,delay='2ms')
net.addLink(h4, h3,0,1,cls=TCLink,bw=100,delay='2ms')
net.addLink(h5, h3,0,2,cls=TCLink,bw=100,delay='2ms')

print "*** Starting network"
# Εκκίνηση του δικτύου
net.build()

# Η function h1.cmd μας επιτρέπει να τρέξουμε εντολές σε κάθε host μέσα στο
# script αντί να χρειαστεί να ανοίξουμε κονσόλα. Εδώ, προσθέτουμε τη διαδρομή
# προς το subnet του h4 ώστε ο client (h1) να μπορεί να επικοινωνήσει με το server (h4)
# μέσω του router (10.0.0.2)
h1.cmd("ip route add 20.0.0.0/24 via 10.0.0.2 dev h1-eth0")

# Ο host h2 παραμετροποιείται ώστε να λειτουργεί σαν δρομολογητής. Ενεργοποιείται η
# ικανότητα
# forwarding και προστίθεται το δεύτερο eth interface για το δεύτερο subnet
h2.cmd("echo 1 > /proc/sys/net/ipv4/conf/all/forwarding")
```

```
    h2.cmd("ifconfig h2-eth1 0")
    h2.cmd("ifconfig h2-eth1 20.0.0.2/24 up")

    # Ο host h3 παραμετροποιείται ώστε να λειτουργεί σαν switch. Προστίθεται ένας
    # linux-bridge
    # και σ'αυτόν προσθέτουμε τα τρία interfaces του. Στη συνέχεια θα συμπεριφέρεται σαν
    #ένα interface με το όνομα br0
    h3.cmd("ifconfig h3-eth0 0")
    h3.cmd("ifconfig h3-eth1 0")
    h3.cmd("ifconfig h3-eth2 0")
    h3.cmd("brctl addbr br0")
    h3.cmd("brctl addif br0 h3-eth0")
    h3.cmd("brctl addif br0 h3-eth1")
    h3.cmd("brctl addif br0 h3-eth2")
    h3.cmd("ifconfig br0 up")

    # Προσθέτουμε στον host4 τη διαδρομή προς το subnet του h1 ώστε ο server (h4)
    #να μπορεί να απαντήσει στον client (h1) μέσω του  (20.0.0.2).
    h4.cmd("ip r add 10.0.0.0/24 via 20.0.0.2 dev h4-eth0")



    print "*** Running CLI"
    # Αυτό θα μας δώσει μια γραμμή εντολών mininet με την οποία θα μπορούμε
    #να εκτελούμε
    # διάφορες εντολές όπως για παράδειγμα να ανοίγουμε ξεχωριστές κονσόλες για
    #κάθε host
    CLI( net )
    net.stop()

    # Η main function
if __name__ == '__main__':
    setLogLevel( 'info' )
    info( '*** Creating network\n' )
    topology()
```

5.2.2 Εκκίνηση εφαρμογών

Για να τρέξουμε το script εκτελούμε 'python dos_traditional_eth.py' σε ένα τερματικό του εικονικού μας μηχανήματος (σαν sudo). Μόλις η δημιουργία της τοπολογίας ολοκληρωθεί βλέπουμε το παρακάτω:



Εικόνα 22 : Δημιουργία τοπολογίας παραδοσιακού δικτύου

Βλέπουμε λοιπόν ότι οι hosts και οι συνδέσεις τους πραγματοποιήθηκαν με τις συγκεκριμένες παραμέτρους. Στο τερματικό mininet δίνουμε την εντολή xterm h1 h2 h3 h4 h5 ώστε να ανοίξουμε μια κονσόλα για κάθε host (για εγκατάσταση του xterm στο vm αρκεί η apt-get install xterm).

Στην κονσόλα του h4 δίνουμε την εντολή

iperf3 -s -i1 -p9090

για να εκκινήσουμε έναν iperf3 server που ακούει στη θύρα 9090.

Στην κονσόλα του h1 εκκινούμε έναν iperf3 client ώστε να διαπιστώσουμε εάν η σύνδεση είναι επιτυχής. Ο client εκκινεί ως εξής:

iperf3 -c 20.0.0.4 -p 9090 -b200M -i1 -t100

όπου:

-c: IP προορισμού

-p: θύρα προορισμού

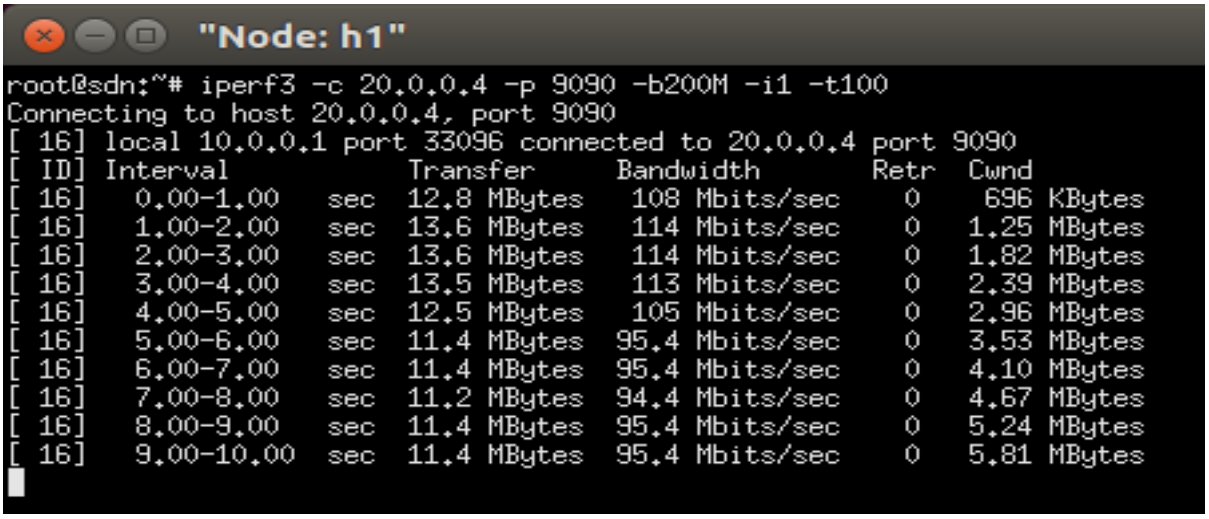
-b: target bandwidth δηλαδή τη ταχύτητα επιθυμούμε για την ανταλλαγή δεδομένων μεταξύ client και server (εδώ έχουμε επιλέξει 200M αλλά υπενθυμίζουμε ότι έχουμε θέσει όριο στις συνδέσεις μας τα 100M. Περιμένουμε, λοιπόν, αυτό να γίνει φανερό στα αποτελέσματά μας)

-i1: εμφάνιση αποτελεσμάτων κάθε 1 δευτερόλεπτο

-t100: τρέξιμο του client για 100 δευτερόλεπτα

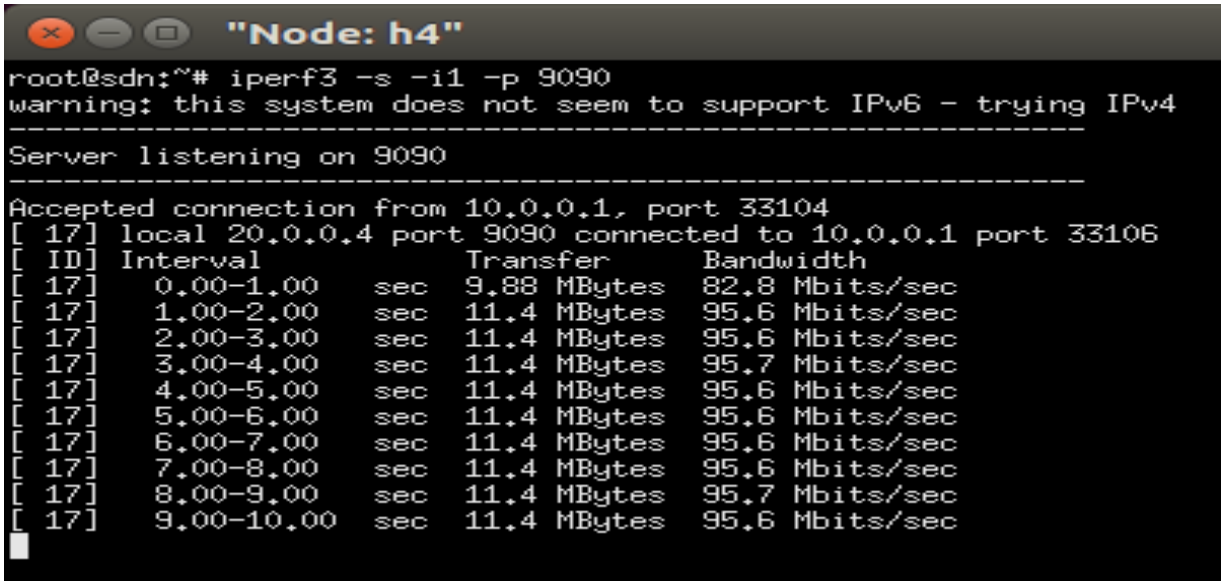
Η default σύνδεση είναι TCP

Βλέπουμε τα εξής στον client:



Εικόνα 23 : εκτέλεση iperf3 στον h1

και στον server:



Εικόνα 24:εκτέλεση iperf στον h4

Είναι αντιληπτό ότι παρόλο που έχουμε επιλέξει ταχύτητα 200Mbps η σύνδεσή μας φτάνει μέχρι τα 100Mbps καθώς τόσο έχουμε ορίσει το bandwidth στις διασυνδέσεις μας. Επίσης,

από τη στιγμή που οι server και client ανταλλάσουν δεδομένα σημαίνει ότι η παραμετροποίηση του router και του switch έχει γίνει σωστά.

5.2.3 Εκκίνηση επίθεσης

Είμαστε έτοιμοι να ξεκινήσουμε μια επίθεση DDoS την οποία θα προσομοιώσουμε με τη βοήθεια του hping3 [15]. Στην κονσόλα του h5 ο οποίος παίζει το ρόλο του attacker δίνουμε:

hping3 -c 10000 -d 120 -S -w 64 -p 80 –flood –rand-source 20.0.0.4

όπου:

-c 10000: αριθμός πακέτων για αποστολή

-d 120: μέγεθος κάθε πακέτου για αποστολή

-S: χρήση SYN πακέτων

-w 64: μέγεθος TCP παραθύρου

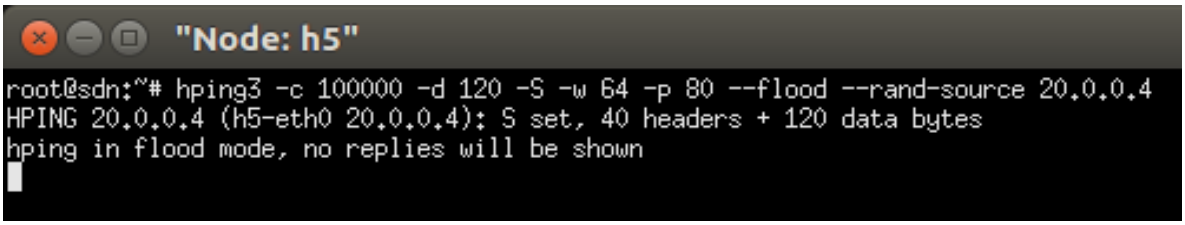
-p 80: θύρα προορισμού

--flood: λειτουργία “πλημμύρας”. Αυτό σημαίνει ότι ο αποστολέας στέλνει πακέτα όσο γρηγορότερα μπορεί χωρίς να ενδιαφέρεται για ενδεχόμενες απαντήσεις από τον server

--rand-source: επιλογή τυχαίας IP προέλευσης ώστε να κρυφτεί η πραγματική μας IP

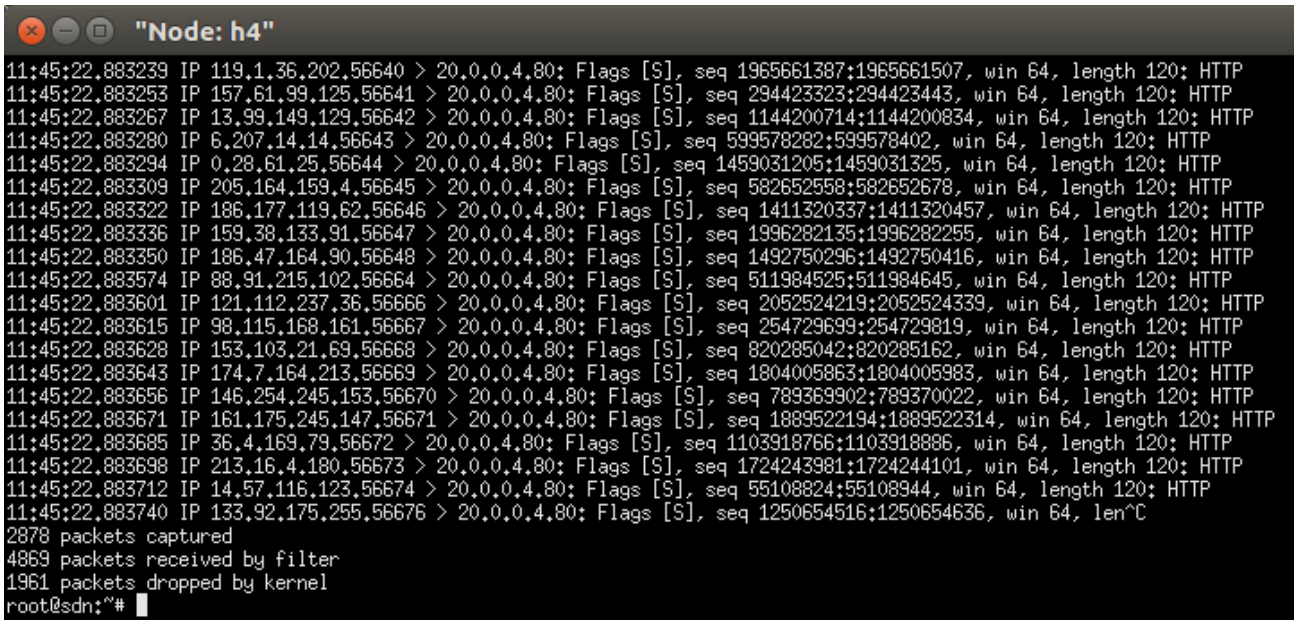
Με τις συγκεκριμένες παραμέτρους του hping3 επιχειρούμε να εκτελέσουμε μια TCP SYN Flood επίθεση στην οποία ο επιτιθέμενος στέλνει πολλές αιτήσεις για σύνδεση στον στόχο (το πακέτο SYN είναι το πρώτο στο 3-way handshake του TCP) χωρίς να περιμένει απάντηση. Έτσι, καταναλώνει τους πόρους του στόχου και τον καθιστά ανίκανο να ικανοποιήσει αιτήματα από κανονικούς χρήστες.

Στον h5 βλέπουμε:



Εικόνα 25 : Εντολή επίθεσης

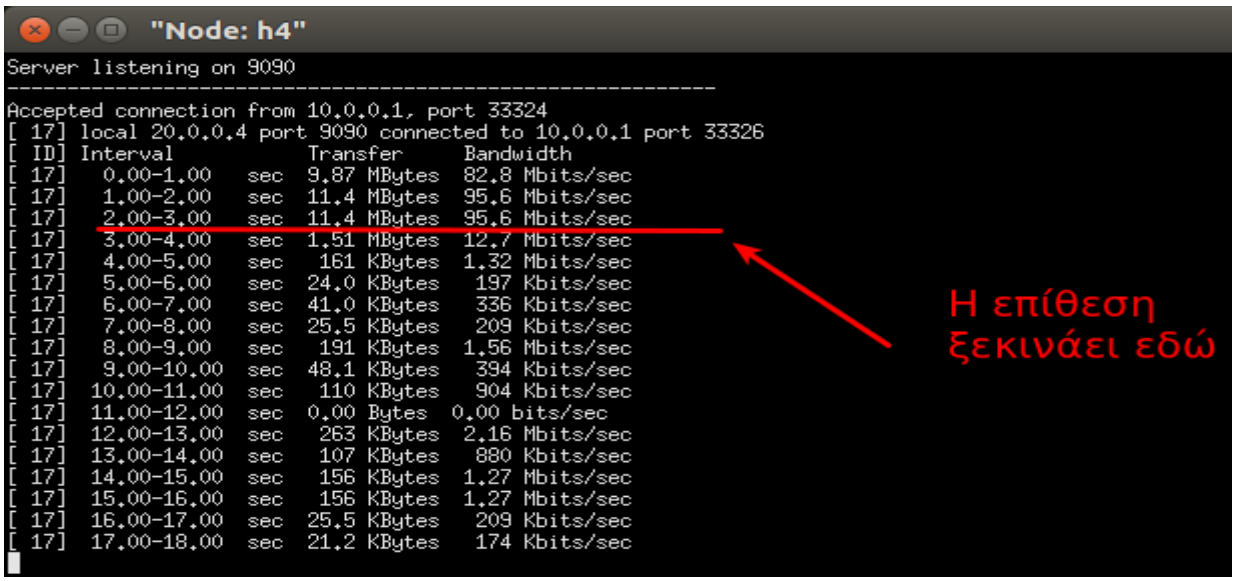
και χρησιμοποιώντας το tcpdump στον host h4 βλέπουμε:



Εικόνα 26 : ο h4 δέχεται τα πακέτα της επίθεσης

Παρατηρούμε ότι ο στόχος h4 δέχεται πάρα πολλά πακέτα σε μικρό χρονικό διάστημα κάθε ένα από τα οποία όντως έχει διαφορετική IP προέλευσης.

Αν τώρα κοιτάξουμε τα αποτελέσματα του iperf3 server μετά το ξεκίνημα βλέπουμε τα εξής:



Εικόνα 27 : Επιτυχία επίθεσης

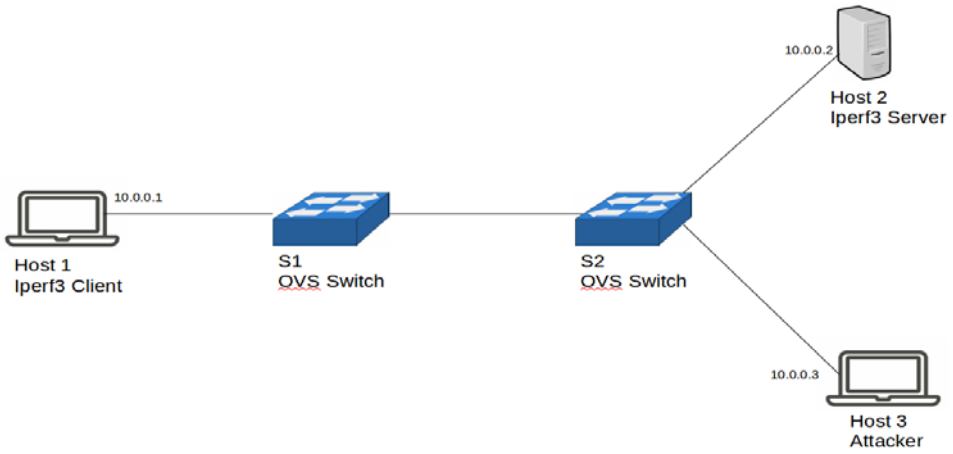
Ενώ η ταχύτητα της σύνδεσης iperf3 ξεκινάει στα 100Mbps, στο τρίτο δευτερόλεπτο εκκινούμε την hping3 επίθεση από τον h5. Βλέπουμε ότι ενώ η σύνδεση ξεκινά στα περίπου 100Mbps στη συνέχεια σταδιακά πέφτει σε μερικά Kbps. Θεωρούμε, λοιπόν, την επίθεση επιτυχημένη. Μπορούμε να φανταστούμε τον αντίκτυπο που θα είχε μια τέτοια επίθεση σε

ένα ρεαλιστικό σενάριο εάν στη θέση του iperf3 έχουμε για παράδειγμα έναν FTP server από τον οποίο διάφοροι clients κατεβάζουν αρχεία. Είναι κατανοητό με ταχύτητα μερικών δεκάδων Kbps ακόμα και το κατέβασμα μικρών αρχείων θα ήταν μια χρονοβόρα διαδικασία. Η συγκεκριμένη επίθεση θα μπορούσε να αντιμετωπιστεί -αφού εντοπιστεί- με διάφορους τρόπους (π.χ. εγκατάσταση mirror server για load balancing, κατάλληλο configuration του firewall κτλ.) αλλά συνήθως απαιτεί την επέμβαση ενός διαχειριστή και δεν είναι δυναμική. Για δυναμική αντιμετώπιση υπάρχουν αρκετές εφαρμογές οι οποίες όμως πρέπει να εγκατασταθούν σε κάθε συσκευή του δικτύου, γεγονός που κάνει τη διαδικασία της προστασίας χρονοβόρα και πολυέξοδη. Θα επιχειρήσουμε να δείξουμε ότι η ίδια επίθεση αντιμετωπίζεται δυναμικά και γρηγορότερα σε ένα SDN δίκτυο με χρήση μιας κατάλληλης εφαρμογής.

5.3 DoS Επίθεση σε SDN δίκτυο

5.3.1 Περιγραφή μοντέλου

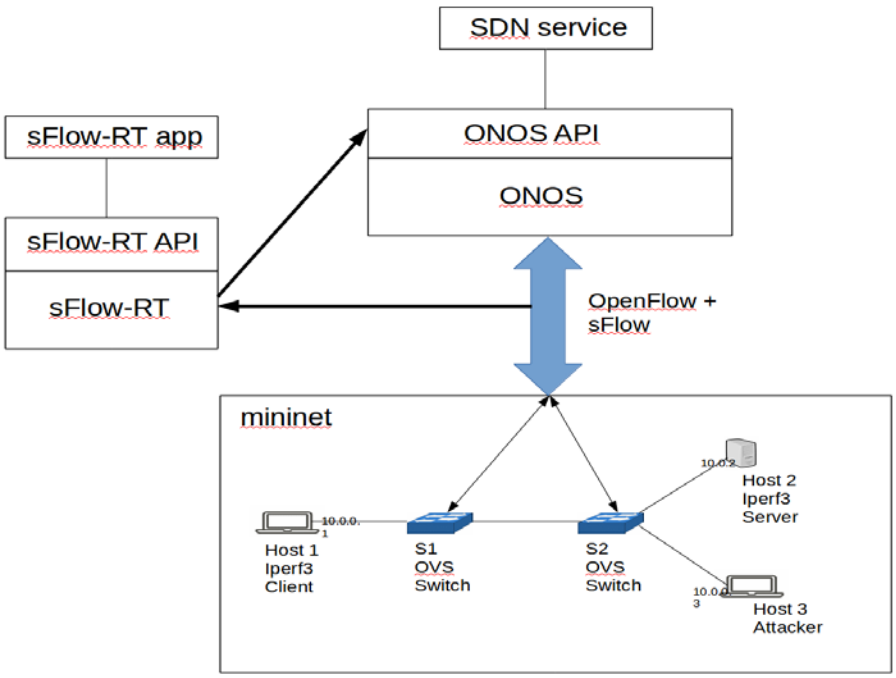
Για τη διεξαγωγή της επίθεσης DoS στο SDN δίκτυο μας[8] θα δημιουργήσουμε την παρακάτω απλή τοπολογία συγκρίσιμη με την τοπολογία του κλασικού Ethernet δικτύου του προηγούμενου κεφαλαίου. Έχουμε αντικαταστήσει τον router (h2) και τον μεταγωγέα (h3) με OVS μεταγωγείς. Οι hosts είναι κι εδώ τρεις, ο iperf3 client, ο iperf3 server και ο εισβολέας.



Εικόνα 28 : Τοπολογία SDN δικτύου

Στην περιγραφή του software και των εργαλείων που θα χρησιμοποιηθούν αναφέραμε πως θα χρησιμοποιήσουμε τον ελεγκτή ONOS ο οποίος επικοινωνεί με τους μεταγωγείς μέσω του OpenFlow. Επίσης, αναφέραμε πως θα χρησιμοποιήσουμε τον sFlow-RT για τη συλλογή και ανάλυση δεδομένων από τους μεταγωγείς. Ο sFlow-RT controller επικοινωνεί με τους

μεταγωγείς μέσω του πρωτοκόλλου sFlow. Έχουμε, δηλαδή, την παρακάτω αρχιτεκτονική του συστήματός μας:



Εικόνα 29 : Αρχιτεκτονική SDN προσομοίωσης

Στο υπόλοιπο της εργασίας υποθέτουμε ότι όλοι οι φάκελοι των εργαλείων (onos, mininet κτλ.) καθώς και τα αρχεία του παραπάνω αποθετηρίου βρίσκονται στον home folder μας.

5.3.2Εκκίνηση εφαρμογών

Για το ξεκίνημα της επίδειξης απαιτείται συνεργασία διάφορων εφαρμογών που εξηγούνται παρακάτω:

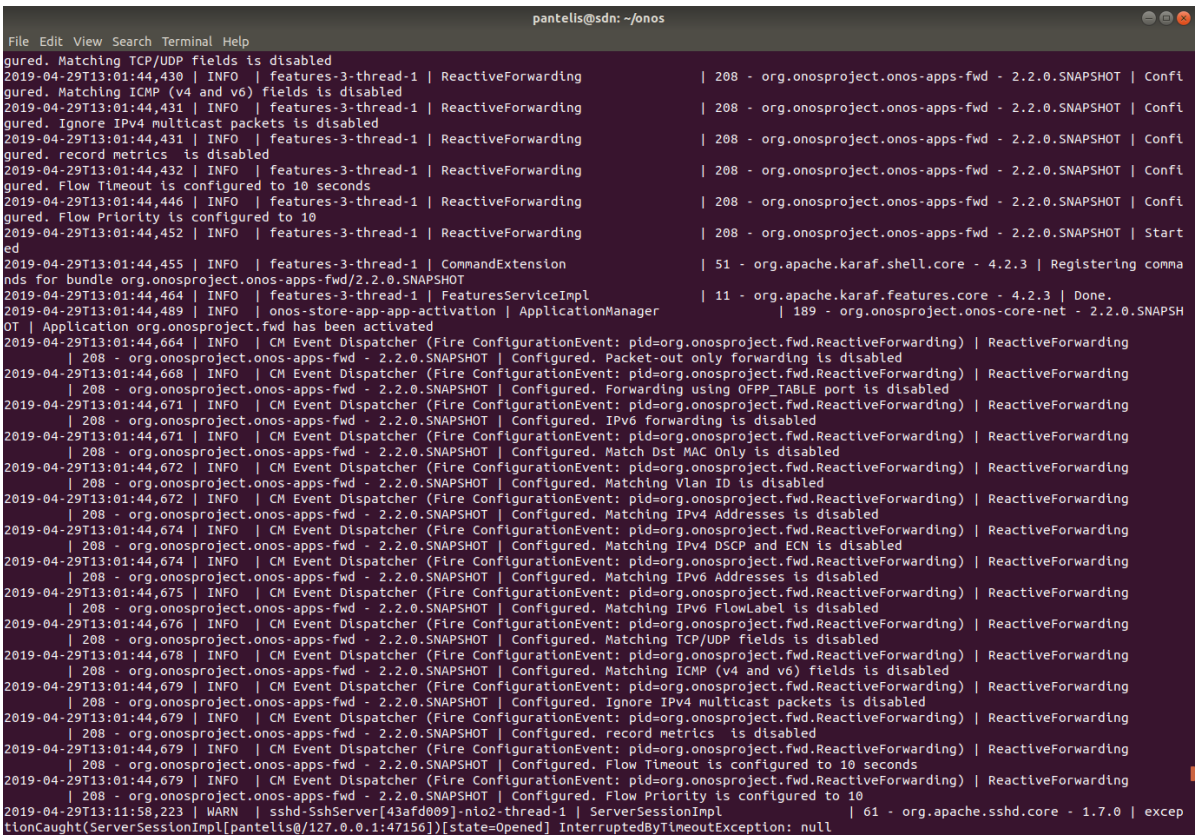
Εκκίνηση του ONOS Controller

$ cd onos

$ export ONOS_ROOT=$(pwd)

$ bazel run onos-local – clean

Η εκκίνηση μπορεί να πάρει αρκετά λεπτά.



Εικόνα 30: Εκκίνηση ONOS Controller

Εκκίνηση των απαιτούμενων εφαρμογών του ONOS

$ cd onos

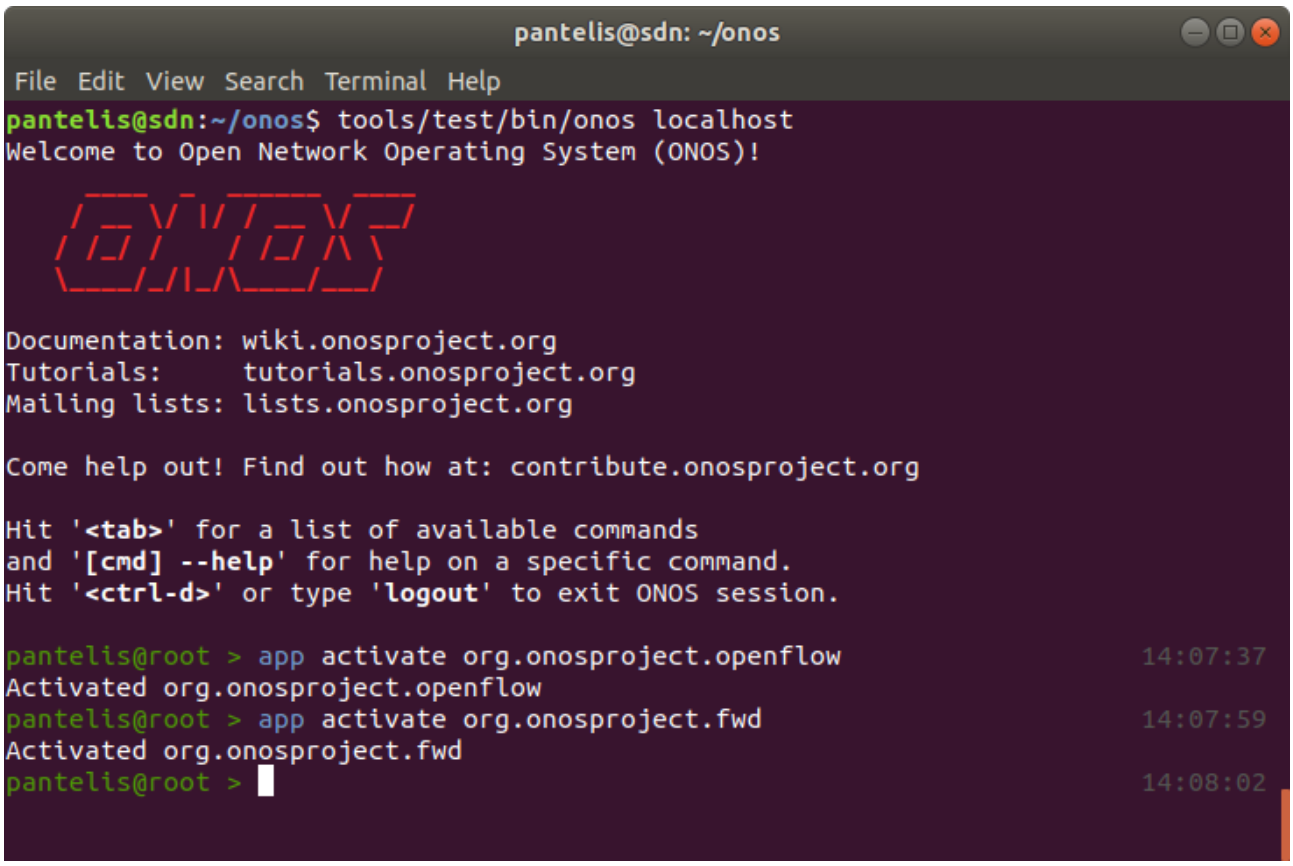
$ tools/test/bin/onos localhost

Αυτό θα ξεκινήσει τη γραμμή εντολών του ONOS στην οποία δίνουμε τις παρακάτω εντολές για το ξεκίνημα των εφαρμογών

$ ONOS > app activate org.onosproject.openflow

$ ONOS > app activate org.onosproject.fwd

Η εφαρμογή OpenFlow χρειάζεται για κάποια ελάχιστη αρχική παραμετροποίηση των μεταγωγέων και αναγνώριση της τοπολογίας από τον ελεγκτή (σημείωση: δεν είναι η εφαρμογή που καθιστά τον ελεγκτή openflow-enabled. Η εφαρμογή που χρησιμοποιείται γι'αυτό το σκοπό ονομάζεται openflow-base και είναι προαπαιτούμενη για την ενεργοποίηση της openflow) και η εφαρμογή fwd (forwarding) η οποία απαιτείται για την εγκατάσταση των κανόνων ροής δυναμικά ανάλογα με την εισερχόμενη κίνηση ώστε οι hosts να μπορούν να επικοινωνήσουν μεταξύ τους. Οι δύο αυτές εφαρμογές κάνουν χρήση του Northbound API του ONOS.



Εικόνα 31 : Εκκίνηση εφαρμογών από ONOS

Μπορούμε, επίσης, να δούμε τη λίστα όλων των διαθέσιμων επίσημων εφαρμογών δίνοντας την εντολή

$ ONOS> apps -s

Example of the apps, ONOS CLI command

```
onos> apps -s
    4 org.onosproject.scalablegateway      1.9.0.SNAPSHOT Scalable GW App
    5 org.onosproject.distributedprimitives 1.9.0.SNAPSHOT Distributed Primitives Test App
    6 org.onosproject.patchpanel           1.9.0.SNAPSHOT Patch Panel
    7 org.onosproject.netcfglinksprovider  1.9.0.SNAPSHOT Network Config Link Provider
    8 org.onosproject.isis                 1.9.0.SNAPSHOT ISIS Provider
    9 org.onosproject.cip                  1.9.0.SNAPSHOT Cluster IP alias App
   10 org.onosproject.openflow-message     1.9.0.SNAPSHOT Control Message Stats Provider
   11 org.onosproject.segmentrouting       1.9.0.SNAPSHOT Segment Routing App
   12 org.onosproject.virtualbng           1.9.0.SNAPSHOT Virtual Broadband Gateway App
   13 org.onosproject.metrics              1.9.0.SNAPSHOT OpenStack Interface App
   14 org.onosproject.ovsdb-base           1.9.0.SNAPSHOT OVSDB Provider
   15 org.onosproject.drivers.ovsdb        1.9.0.SNAPSHOT OVSDB Device Drivers
   16 org.onosproject.yms                  1.9.0.SNAPSHOT YANG Management System App
   17 org.onosproject.influxdbmetrics      1.9.0.SNAPSHOT InfluxDB Report and Query App
   18 org.onosproject.bgp                  1.9.0.SNAPSHOT BGP Provider
   19 org.onosproject.restsb               1.9.0.SNAPSHOT REST Provider
*  20 org.onosproject.hostprovider         1.9.0.SNAPSHOT Host Location Provider
*  21 org.onosproject.lldpprovider         1.9.0.SNAPSHOT LLDP Link Provider
*  22 org.onosproject.optical-model        1.9.0.SNAPSHOT Optical information model
*  23 org.onosproject.openflow-base        1.9.0.SNAPSHOT OpenFlow Provider
*  24 org.onosproject.openflow             1.9.0.SNAPSHOT OpenFlow Meta App
```

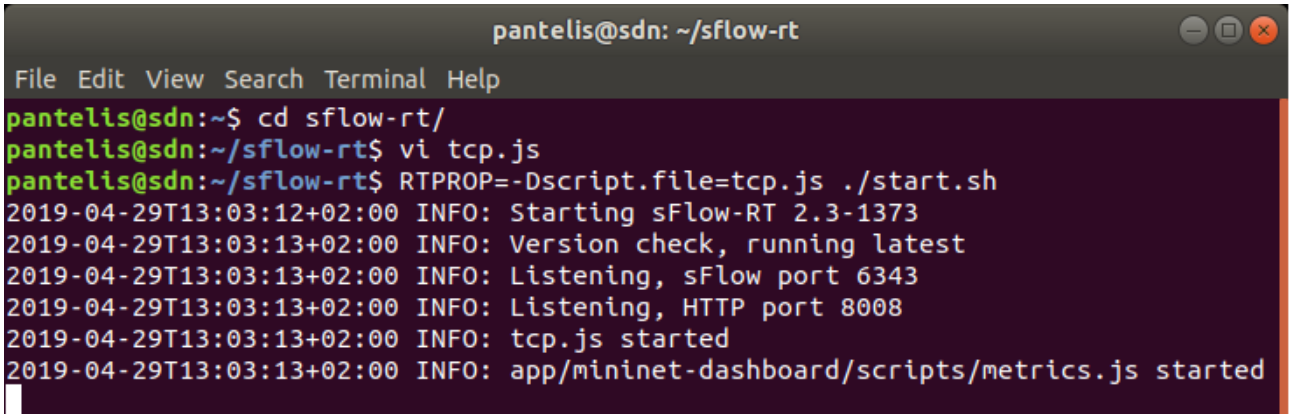
Εικόνα 32: ONOS application list

Εκκίνηση του sFlow-RT controller και της εφαρμογής TCP monitoring

$ cd sflow-rt

$ env RTPROP=-Dscript.file=tcp.js ./start.sh

Το script start.sh χρησιμοποιείται για την εκκίνηση του sFlow-RT controller και δίνεται το αρχείο tcp.js σαν configuration file. Η εφαρμογή αυτή είναι που θα επιβλέπει το δίκτυο για DoS επιθέσεις βασισμένες σε TCP (είναι δηλαδή η Intrusion Detection εφαρμογή) και θα ειδοποιήσει μια εφαρμογή του ελεγκτή που θα επιχειρήσει να τις μετριάσει.



Εικόνα 33 : Εκκίνηση sFlow-RT controller

Η εφαρμογή αυτή (ανάλογα με το configuration file που παίρνει σαν παράμετρο) θέτει κάποια όρια στον αριθμό των πακέτων που δέχεται σε κάποιο χρονικό διάστημα. Αν ο αριθμός αυτός ξεπεραστεί θεωρεί ότι δέχεται επίθεση. Για παράδειγμα στο αρχείο tcp.js σαν όριο ορίζονται οι 100 TCP requests το δευτερόλεπτο. Έτσι, θα χρησιμοποιήσουμε την εφαρμογή για να ανιχνεύσουμε μια SYN flood επίθεση.

Εκκίνηση του mininet και δημιουργία της τοπολογίας

Στη συγκεκριμένη επίδειξη, δεν θα χρησιμοποιήσουμε δικό μας script για να δημιουργήσουμε την επιθυμητή τοπολογία mininet, αλλά το script sflow.py του συγγραφέα. Με το συγκεκριμένο script οι μεταγωγείς είναι sFlow-enabled και αποστέλλουν δεδομένα στον ελεγκτή sFlow που θα χρησιμοποιηθούν για την ανίχνευση απειλής (βλέπε παρακάτω).

Με την παρακάτω εντολή:

$ sudo mn –custom ~/sflow-rt/extras/sflow.py –controller remote –topo linear,2

(πρέπει να εγκατασταθεί το πακέτο της Python “requests” με την εντολή pip install requests)

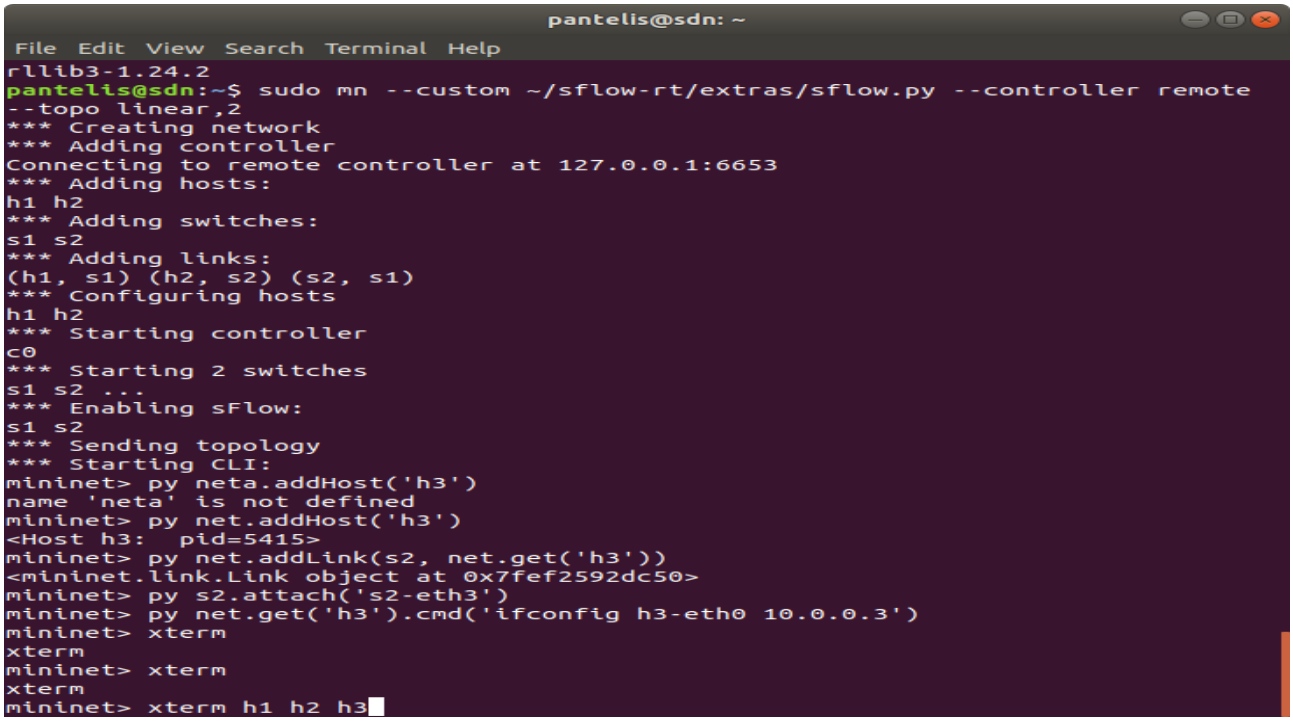
Η παραπάνω εντολή χρησιμοποιεί το script sflow.py που θα δημιουργήσει μια "γραμμική" τοπολογία με δύο OVS μεταγωγείς συνδεδεμένους μεταξύ τους και στον καθένα συνδεδεμένος από ένας host. Με την ολοκλήρωση της δημιουργίας της τοπολογίας mininet θα ανοίξει η γραμμή εντολών του mininet. Εδώ θα πρέπει να δημιουργήσουμε και έναν τρίτο host που θα παίξει το ρόλο του επιτιθέμενου όπως στην εικόνα 28. Στη γραμμή εντολών του mininet δίνουμε:

$ py net.addHost('h3') *# Δημιουργία του host*

$ py net.addLink(s2, net.get('h3') *# Συνδεσή του στον switch s2*

$ py s2.attach('s2-eth3') *# Τον συνδέουμε στην κάρτα δικτύου eth3 του switch*

$ py net.get('h3').cmd('ifconfig h3-eth0 10.0.0.3) # Δίνουμε μια IP στην κάρτα δικτύου του host

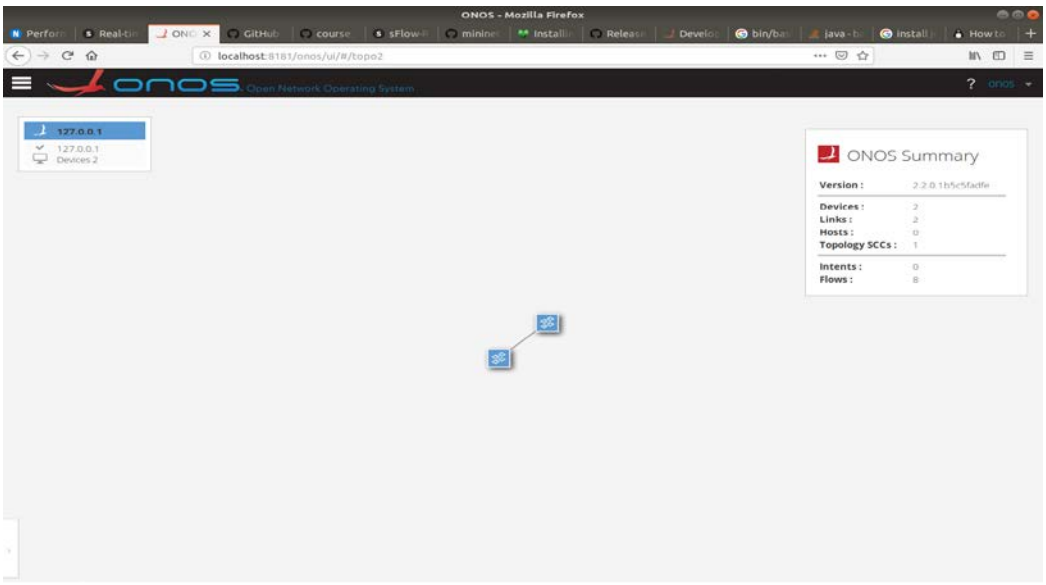


Εικόνα 34 : Δημιουργία τοπολογίας SDN στο Mininet

Από την webpage του onos μπορούμε να έχουμε μια οπτική περιγραφή της τοπολογίας του δικτύου. Σε έναν οποιονδήποτε περιηγητή:

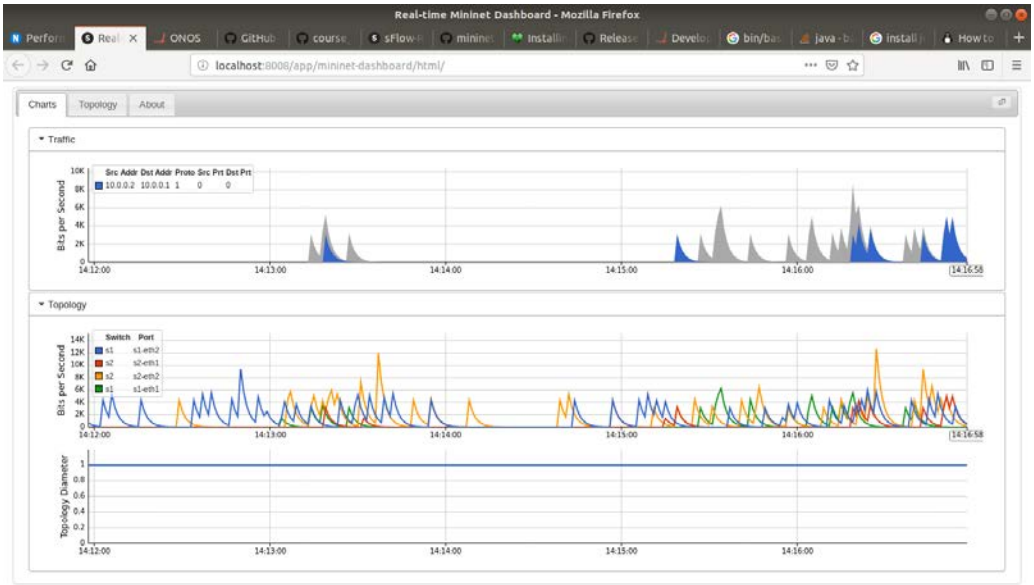
localhost:8181/onos/ui

και βλέπουμε την παρακάτω εικόνα (σημείωση: στην τοπολογία εμφανίζονται μόνο οι μεταγωγείς κι όχι οι hosts):



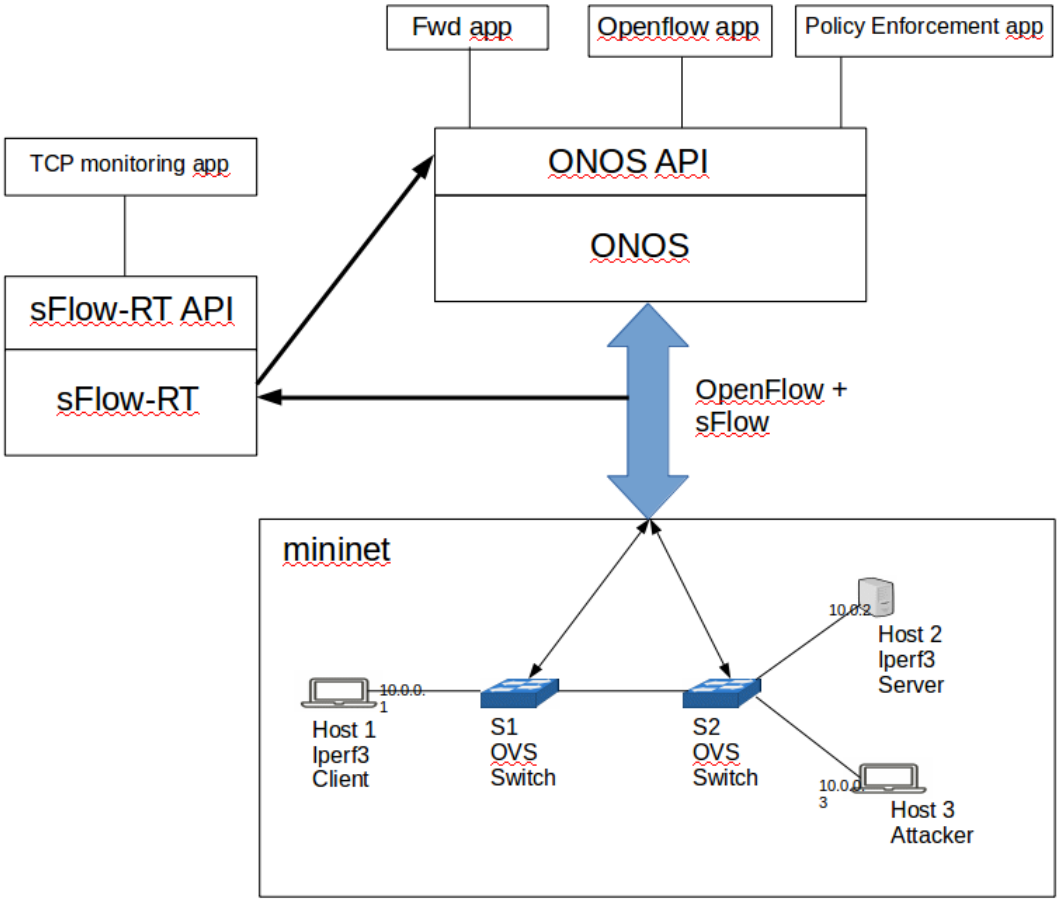
Εικόνα 35 : Η τοπολογία από το GUI του ONOS

Η εφαρμογή mininet-dashboard μας δίνει επίσης κάποια στοιχεία σε σχέση με την κίνηση που ανταλλάσσεται στο δίκτυο μεταξύ των hosts και των switches. Επισκεπτόμαστε τη σελίδα:
localhost:8008/app/mininet-dashboard/html
και βλέπουμε την παρακάτω εικόνα:



Εικόνα 36 : Απεικόνιση της κίνησης στο δίκτυο

Η παρακάτω εικόνα δείχνει την τελική αρχιτεκτονική του συστήματος μας.



Εικόνα 37 : Η τοπολογία του δικτύου μας και η αρχιτεκτονική του συστήματος

5.3.3Εκκίνηση επίθεσης και αντιμετώπιση

Αφού έχουμε εκκινήσει τις απαραίτητες εφαρμογές κι έχουμε δημιουργήσει την απλή τοπολογία δικτύου μας, στη συνέχεια ανοίγουμε μία κονσόλα για κάθε host με την εντολή

mininet> xterm h1 h2 h3

Υπενθυμίζουμε ότι ο h1 είναι ο iperf3 client, ο h2 ο iperf3 server και ο h3 ο επιτιθέμενος.

Στον h2 εκκινούμε τον iperf3 server:

$ iperf3 -s -i1

στον h1 εκκινούμε τον iperf3 client:

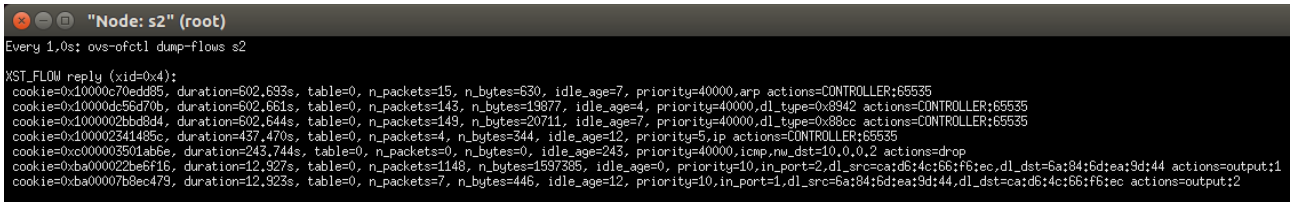
$ iperf3 -c 10.0.0.2 -i1 -b100M -u -t1000

(Σημείωση: εδώ χρησιμοποιούμε μια σύνδεση UDP για το iperf3 καθώς μια σύνδεση TCP μπορεί να θεωρηθεί απειλή και να μπλοκαριστεί απ'το IDS μας. Η επίθεση θα γίνει με TCP SYN flood).

Για να δούμε τους εγκατεστημένους κανόνες ροής στον μεταγωγέα s2 ανοίγουμε μια κονσόλα στον s2 και δίνουμε:

$ ovs-ofctl dump-flows s2

Το αποτέλεσμα είναι:



Εικόνα 38: οι flow tables του s2

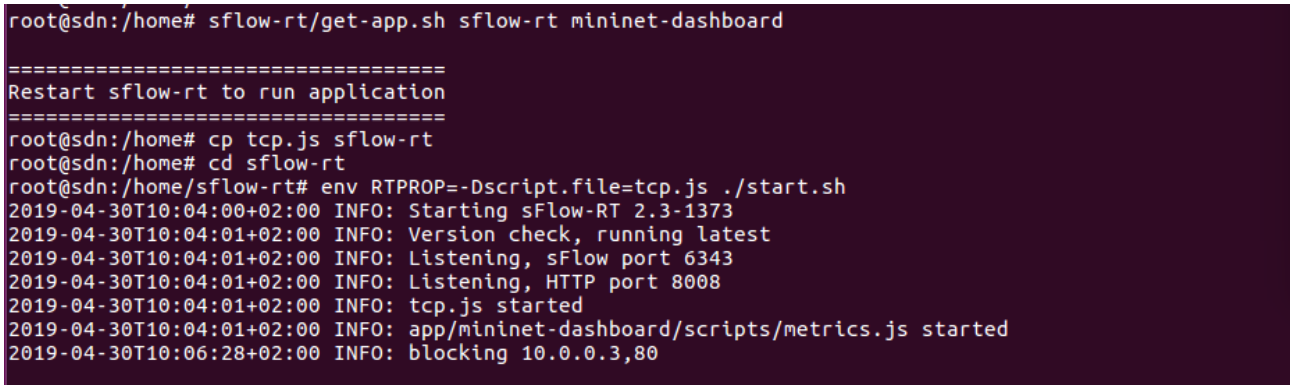
Οι δύο τελευταίες καταχωρίσεις του πίνακα είναι η iperf3 ροή μας.

Τέλος, μετά από λίγο, εκκινούμε και την επίθεσή μας από τον h3:

$ hping3 -S –flood -p 80 10.0.0.2

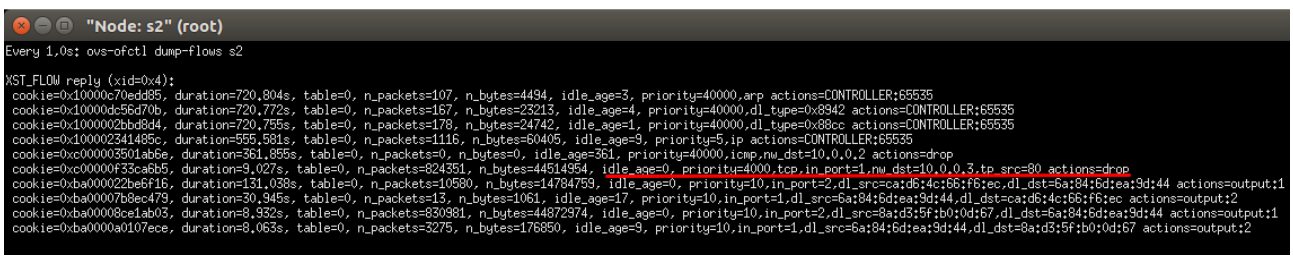
Παρατηρούμε λοιπόν τα εξής:

Η εφαρμογή μας, εντόπισε την επίθεση που προέρχεται από την IP του επιτιθέμενου (10.0.0.3) προς τη θύρα 80 του θύματος (10.0.0.2) και εγκατέστησε έναν κανόνα ροής στον s2 για να μπλοκάρει την κακόβουλη κίνηση.



Εικόνα 39: Εντοπισμός της επίθεσης

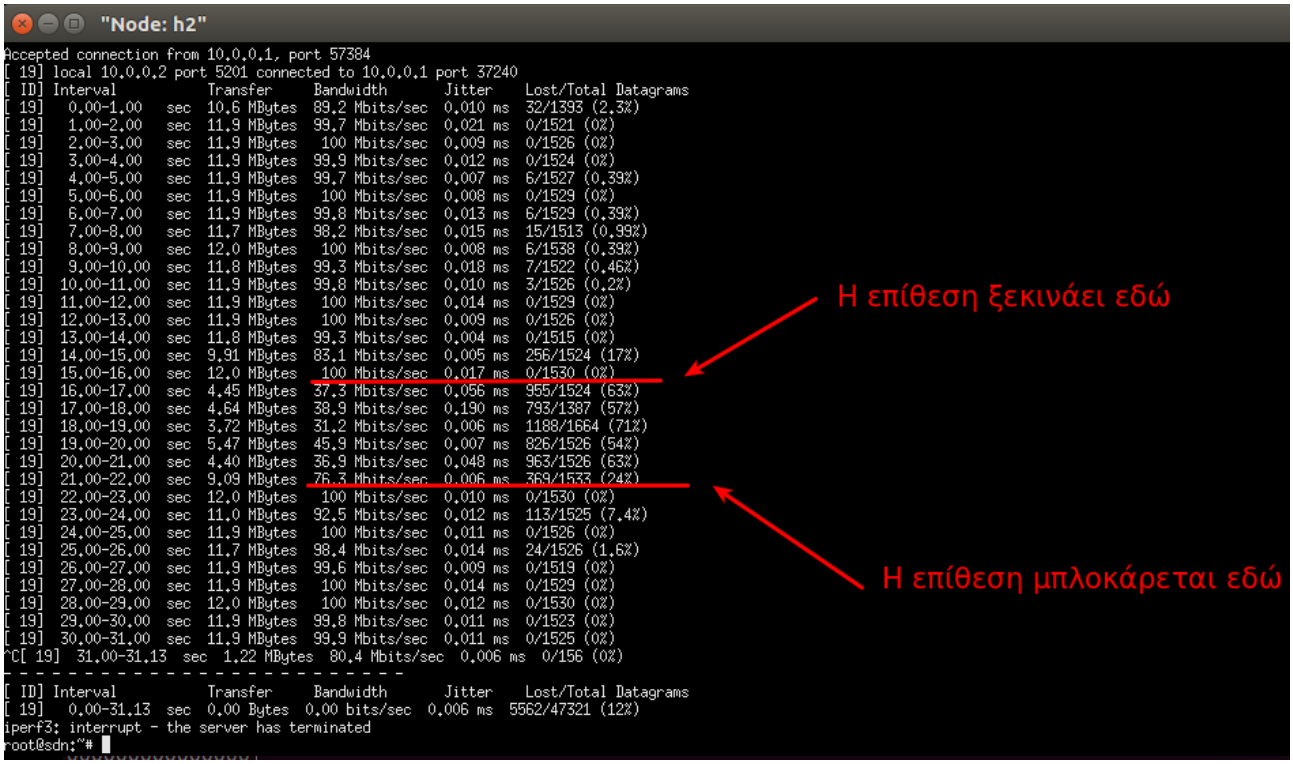
Οι κανόνες ροής στον s2 είναι τώρα:



Εικόνα 40: οι νέοι flow tables του s2

Η εφαρμογή μας εγκατέστησε έναν κανόνα στον μεταγωγέα s2 που μπλοκάρει τις TCP RST προς τον επιτιθέμενο (IP προορισμού 10.0.0.3 και θύρα προέλευσης 80) ώστε ο επιτιθέμενος να παραπλανηθεί πιστεύοντας ότι ο εξυπηρετητής δεν είναι διαθέσιμος και να σταματήσει την επίθεσή του.

Στον iperf3 server βλέπουμε μια σύντομη πτώση του bandwidth το οποίο επανέρχεται στα κανονικά επίπεδα (~100Mbps) μόλις η επίθεση μπλοκαριστεί.



Εικόνα 41: εντοπισμός και αποτροπή της επίθεσης

Συνεπώς, με τη χρήση του ελεγκτή ONOS και sFlow-RT και των κατάλληλων εφαρμογών τους, καταφέραμε να δημιουργήσουμε ένα σύστημα IDPS policy-based το οποίο ανιχνεύει μια επίθεση TCP SYN flood και την αντιμετωπίζει σε σύντομο χρονικό διάστημα αποτελεσματικά.

5.4 Συμπέρασματα

Στο κεφάλαιο αυτό επιχειρήσαμε να δείξουμε τη διαφορά του αντίκτυπου που έχει μια από τις πιο κοινές επιθέσεις, η DoS, σε ένα παραδοσιακό δίκτυο Ethernet και σε ένα SDN δίκτυο και να τονίσουμε πως τα χαρακτηριστικά της αρχιτεκτονικής του SDN μπορούν να του δώσουν προβάδισμα. Στο παραδοσιακό δίκτυο Ethernet η αντιμετώπιση της θα απαιτούσε κατάλληλη και στατική παραμετροποίηση κάθε συσκευής που συμμετέχει στο δίκτυο. Δηλαδή, κανόνες firewall σε κάθε δρομολογητή, μεταγωγέα, host κτλ. Η εγκατάσταση ενός συστήματος IDPS σε ένα παραδοσιακό δίκτυο απαιτεί επίσης εγκατάσταση λογισμικού σε

κάθε συσκευή ξεχωριστά για να είναι αποτελεσματική. Στη μικρή τοπολογία που δημιουργήσαμε (μόνο για λόγους επίδειξης) αυτό ίσως να μην αποτελεί πρόβλημα. Σε ένα πραγματικό δίκτυο, όμως, με εκατοντάδες ή και χιλιάδες συσκευές, είναι φανερό ότι η εγκατάσταση ενός τέτοιου συστήματος μπορεί να είναι διαδικασία χρονοβόρα, δαπανηρή και ενδεχομένως αναποτελεσματική. Από την άλλη, είδαμε ότι η κεντρική διαχείριση του δικτύου από τον ελεγκτή αποτελεί ένα ισχυρό πλεονέκτημα έναντι τέτοιου είδους απειλών. Αντί για την εγκατάσταση προληπτικών μέτρων σε κάθε συσκευή, το σύστημα IDPS με τη βοήθεια του ελεγκτή μπορεί δυναμικά να εφαρμόσει συγκεκριμένες firewalling policies πολύ πιο στοχευμένα και να αντιμετωπίσει την επίθεση αποτελεσματικότερα χωρίς να επηρεάζει το υπόλοιπο δίκτυο και τη νόμιμη κίνηση. Για παράδειγμα, αναφέρουμε ότι στη δική μας περίπτωση το μπλοκάρισμα της κίνησης έγινε μόνο στο μεταγωγέα s2 καθώς εκεί εντοπίστηκε η κακόβουλη κίνηση χωρίς να επηρεάζει την κίνηση στο μεταγωγέα s1. Είναι φανερό, ότι για μεγάλα δίκτυα η στοχευμένη αυτή δράση του συστήματος είναι πιο κατάλληλη από τη γενικευμένη παραμετροποίηση των συσκευών του δικτύου.

Κεφάλαιο 6
Συμπεράσματα Διατριβής και μελλοντική έρευνα

Τα χαρακτηριστικά του SDN δίνουν εξαιρετικά μεγάλες δυνατότητες ανάπτυξης εφαρμογών και εργαλείων στους οργανισμούς και τους developers. Η αφαιρετικότητα με την οποία μπορούν πλέον να διαχειριστούν την υποδομή του δικτύου απλοποιώντας την σύνθετη λειτουργία τους μπορεί να επιφέρει σπουδαία αποτελέσματα.

Ωστόσο πέρα από τους νεοτερισμούς και κάποια σοβαρά πλεονεκτήματα έναντι των κλασικών δικτύων που προσφέρει το SDN, η κεντρική διαχείριση και η πολυεπίπεδη αρχιτεκτονική που εισάγει, δημιουργεί νέα τρωτά σημεία και νέες προκλήσεις για τους μηχανικούς ασφαλείας των δικτύων. Μην ξεχνάμε το Μοναδικό Σημείο Αποτυχίας που εισάγει ο ελεγκτής στους κινδύνους ασφαλείας. Το πλεονέκτημα όμως του SDN να είναι ακόμα υπό ανάπτυξη δίνει τη δυνατότητα στους μηχανικούς να σχεδιάσουν την αρχιτεκτονική με την ασφάλεια σαν προαπαιτούμενο και να μην επαναληφθούν τα λάθη του παρελθόντος με τα παραδοσιακά δίκτυα όπου η ασφάλεια άρχισε να λαμβάνεται υπόψη αρκετά αργότερα από την ανάπτυξή τους.

Στο κεφάλαιο 2 καταγράψαμε τους σημαντικότερους μηχανισμούς ασφαλείας που έχουν προταθεί και υλοποιηθεί ωστόσο δεν υπάρχει ούτε ένα σημείο στην βιβλιογραφία που να αναφέρει ότι οι μηχανισμοί αυτοί είναι επαρκείς και ότι η ασφάλεια στα SDN δίκτυα έχει καλυφθεί ως τομέας κάτι που είναι απόλυτα φυσιολογικό. Πέρα από την φύση του τομέα της ασφάλειας όπου ποτέ και κανένα σύστημα δεν θεωρείται αρκετά ασφαλές, αν υπολογίσει κανείς τον ανθρώπινο παράγοντα και τις νέες ευπάθειες που δημιουργούνται όσο μεταβάλλονται τα συστήματα, ειδικά για τα Software Defined Networks που είναι μια σχετικά καινούργια τεχνολογία υπάρχουν ακόμα πολλά ζητήματα που θα πρέπει να καλυφθούν. Είναι αναγκαίο λοιπόν να υπάρχει ξεχωριστός τομέας ανάπτυξης στα SDN που να εξετάζει την ασφάλεια ως ζητούμενο.

Στο κεφάλαιο 3 εντοπίζονται οι συχνότερες ευπάθειες και οι πιθανές επιθέσεις που μπορεί να εξαπολυθούν στα SDN και δίνονται προτάσεις που θα παρείχαν στους διαχειριστές την δυνατότητα να τις αντιμετωπίσουν, ενώ στο κεφάλαιο 4 παρουσιάζονται οι τομείς στους οποίους υπερέχουν τα SDN στον τομέα της ασφάλειας και τα χαρακτηριστικά τους αυτά είναι εξαιρετικά κρίσιμο να χρησιμοποιηθούν από τους ερευνητές του μέλλοντος.
Όσον αφορά την σύγκριση των δυο προσεγγίσεων δεν είναι εφικτό να συγκριθούν σε απόλυτο βαθμό ως προς τα αποτελέσματα της ασφάλειας τα παραδοσιακά δίκτυα με τα SDN, λόγω της διαφορετικής αρχιτεκτονικής τους και των διαφορετικών συστατικών τους, καθώς και του ότι πολλές από τις ευπάθειες τους δεν είναι ίδιες. Το δεδομένο είναι όμως ότι λόγω του κεντρικοποιημένου χαρακτήρα του SDN είναι πολύ πιο εύκολο και πιο ευέλικτο να εφαρμοστεί μια πολιτική ασφαλείας σε όλο το δίκτυο, ειδικά αν πρόκειται για ένα μεγάλο δίκτυο ενός οργανισμού. Αυτό ήταν και το ζητούμενο στην υλοποίηση που παρουσιάζεται στην διατριβή στο κεφάλαιο 5, να είναι όσο το δυνατόν ίδια τα δεδομένα για τις δύο προσεγγίσεις και μέσα από την εκτέλεση των επιθέσεων να αναδειχθούν τα πλεονεκτήματα που μπορεί να προσφέρουν τα Software Defined Networks. Ως προς την υλοποίηση των προσομοιώσεων λοιπόν με την πλατφόρμα του Mininet, αναπτύξαμε ένα python script το οποίο μας επέτρεψε να χρησιμοποιήσουμε κάποιους από τους εικονικούς hosts ώστε να λειτουργούν ως απλοί μεταγωγείς και να έχουμε μια προσέγγιση παραδοσιακών δικτύων και να καταφέρουμε με την χρήση των κατάλληλων εργαλείων να εκτελέσουμε μια DoS επίθεση στον server. Στην προσέγγιση του SDN για τον ίδιο τύπο επίθεσης εφαρμόσαμε ένα IDPS μέσω του ελεγκτή sFlow το οποίο ανίχνευσε την επίθεση μας και την αντιμετώπισε αποτελεσματικά και στοχευμένα δίνοντας της πλεονέκτημα. Τα χαρακτηριστικά δηλαδή της προγραμματιστικότητας και της ευελιξίας που παρέχει τη SDN αρχιτεκτονική λειτούργησαν προς όφελος της ασφάλειας χωρίς να απαιτείται χρονοβόρα διαδικασία και χωρίς να επιβαρύνεται το δίκτυο.

Στα σχέδια για μελλοντική έρευνα υπάρχει η πληρέστερη καταγραφή των μηχανισμών αντιμετώπισης των προκλήσεων στα SDN δίκτυα, η κατηγοριοποίηση τους και η παρουσίαση τους σε ένα πλήρη οδηγό καλών πρακτικών που θα πρέπει να τηρούνται κατά την υλοποίηση SDN. Θα γίνει επίσης η προσπάθεια για την δημιουργία ενός framework το οποίο θα ενσωματώνει τους καταλληλότερους από τους υπάρχοντες μηχανισμούς ασφαλείας και θα παρέχει ένα πλαίσιο μέσω του οποίου θα μπορούν άλλοι ερευνητές να συμπληρώνουν τις δικές τους τεχνικές ή να το χρησιμοποιούν ως βασικό εργαλείο ενσωμάτωσης ασφάλειας στο SDN δίκτυο τους.

Βιβλιογραφία

[1] Benton, K., Camp, L.J. and Small, C. (2013) OpenFlow Vulnerability Assessment. Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ACM, New York,

[2] Dabbagh, M., Hamdaoui, B., Guizani, M. and Rayes, A. (2015). Software-defined networking security: pros and cons. IEEE Communications Magazine, 53(6).

[3] Gao, S., Li, Z., Xiao, B. and Wei, G. (2018). Security Threats in the Data Plane of Software-Defined Networks. IEEE Network, 32(4), pp.108-113.

[4] Shaghaghi, Arash & Kaafar, Mohamed Ali & Buyya, Rajkumar & Jha, Sanjay. (2018). Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions.

[5] L. Dribi, M. Zhani,(2016) SDN-Guard: DoS Attacks Mitigation in SDN Networks, 5th IEEE International Conference on Cloud Networking

[6] T. Nguyen, The Challenges in SDN/ML Based Network Security: A Survey, (2018) 2nd Cyber Security in Networking Conference (CSNet)

[7]Potluri. A. (n.d.). [online] Scis.uohyd.ac.in. Available at: https://scis.uohyd.ac.in/~apcs/talks/sdn-sec.pdf

[8] Varadharajan, V., Karmakar, K., Tupakula, U. and Hitchens, M. (2019). A Policy-Based Security Architecture for Software-Defined Networks. IEEE Transactions on Information Forensics and Security, 14(4), pp.897-912.

[9] Open Networking Foundation. (2019). Open Networking Foundation is an operator led consortium leveraging SDN, NFV and Cloud technologies to transform operator networks and business models. [online] Available at: https://www.opennetworking.org/

[10] Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet. [online] Mininet.org. Available at: http://mininet.org

[11] *Hping - Active Network Security Tool*. [online] Available at: http://hping.org/.

[12] Tcpdump.org. (2019). TCPDUMP/LIBPCAP public repository. [online] Available at: https://www.tcpdump.org/

[13] GUEANT, V. (2019). iPerf - The TCP, UDP and SCTP network bandwidth measurement tool. [online] Iperf.fr. Available at: https://iperf.fr

[14] ONOS. (2019). Open Network Operating System. [online] Available at: https://onosproject.org

[15] Policy, P. and Policy, P. (2015). Denial-of-service Attack – DoS using hping3 with spoofed IP in Kali Linux. [online] blackMORE Ops. Available at: https://www.blackmoreops.com/2015/04/21/denial-of-service-attack-dos-using-hping3-with-spoofed-ip-in-kali-linux/ .

[16] sFlow-rt -Real-time analytics for actionable intelligence to support DevOps, Orchestration and SDN applications , (2019). [online] Available at: https://www.sflow-rt.com/

[17] Opennetworking.org. (2015). OpenFlow Switch Specification Version 1.5.1. [online] Available at: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[18] L. Ertaul, K. Venkatachalam (2017), Security of Software Defined Networks (SDN),International Conf. Wireless Networks ICWN '17

[19] A. AlEroud, I. Alsmadi,(2017), Identifying cyber-attacks on software defined networks An inference-based intrusion detection approach, Journal of Network and Computer Applications, Volume 80

[20] Scott-Hayward, S., Natarajan, S. and Sezer, S. (2016). A Survey of Security in Software Defined Networks. *IEEE Communications Surveys & Tutorials*, 18(1), pp.623-654.

[21] Ahmad, I., Namal, S., Ylianttila, M. and Gurtov, A. (2015). Security in Software Defined Networks: A Survey. IEEE Communications Surveys & Tutorials, 17(4), pp.2317-2346.

[22] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in Recent Advances in Intrusion Detection. Springer, 2011, pp. 161-180.

[23] J. R. Ballard, I. Rae, and A. Akella, "Extensible and scalable network monitoring using opensafe," Proc.INM/WREN, 2010.

[24]Syed Taha Ali, Vijay Sivaraman, Adam Radford, and Sanjay Jha, "A Survey of Securing Networks Using Software Defined Networking", IEEE TRANSACTIONS ON RELIABILITY, VOL. 64, NO. 3, SEPTEMBER 2015

[25] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in IEEE 35th Conference on Local Computer Networks (LCN). IEEE, 2010, pp. 408-415.

[26] Z. Hu, M. Wang, X. Yan, Y. Yin, Z. Luo, "A comprehensive security architecture for SDN", Proc. 18th Int. Conf. Intell. Next Gener. Netw. (ICIN), pp. 30-37, 2015.

[27] S. Shirali-Shahreza and Y. Ganjali, " Efficient Implementation of Security Applications in OpenFlow Controller with FleXam", in 21st IEEE Annual Symposium on High-Performance Interconnects. IEEE, 2013, pp. 49-54

[28] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in Proceedings of Network and Distributed Security Symposium, 2013

[29]S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in 20th IEEE International Conference on Network Protocols (ICNP). IEEE, 2012, pp. 1-6.

[30]Wen, X., Chen, Y., Hu, C., Shi, C., & Wang, Y. (2013, August). Towards a secure controller platform for openflow applications. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 171-172). ACM.

[31]M. Mendonca, S. Seetharaman, and K. Obraczka, “A flexible in-network IP anonymization service,” in Proc. IEEE ICC, 2012,pp. 6651–6656

[32]Porras, Philip & Shin, Seungwon & Yegneswaran, Vinod & Fong, Martin & Tyson, Mabry & Gu, Guofei.., “A security enforcement kernel for OpenFlow networks,” in Proc. 1st Workshop Hot Topics Softw. Defined Netw., 2012, pp. 121–126.

[33]S. Shin, V. Yegneswaran, P. Porras, and G. Gu, “AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks,” in Proc. ACM SIGSAC Conf. Comput. Commun. Security, 2013, pp. 413–424.

[34]S. Shin , Y. Song , T. Lee , S. Lee , J. Chung , P. Porras , V. Yegneswaran , J. Noh , B. Kang., “Rosemary: A robust, secure, and high-performance network operating system,” in Proc. ACM SIGSAC Conf. Comput. Commun. Security, 2014, pp. 78–89.