Πανεπιστήμιο Δυτικής Μακεδονίας

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημιουργία συστήματος αποτροπής εισβολών σε περιβάλλον Δικτύων Ορισμένων από Λογισμικό με τη χρήση Παγίδων Εισβολών

Λιατίφης Αθανάσιος – Αριθμός Μητρώου 1027

Επιβλέπων Καθηγητής:

Επίκουρος Καθηγητής, Παναγιώτης Σαρηγιαννίδης

Ιούλιος 2020, Κοζάνη

University of Western Macedonia

Faculty of Engineering

Department of Informatics and Telecommunications

An Intrusion prevention system in SDN environments using Honeypots

Liatifis Thanasis – Student ID: 1027

Supervisor:

Assistant Professor, Panagiotis Sarigiannidis

July 2020, Kozani

Τέλος, μετά από πέντε χρόνια μια συναρπαστική διαδρομή έφθασε στο τέλος της. Όχι ακριβώς όπως την περίμενα αλλά αν όλα πήγαιναν όπως τα θέλαμε η ζωή θα ήταν βαρετή! Το επόμενο κεφάλαιο της ζωής μου σύντομα θα ξεκινήσει!

Θα ήθελα να ευχαριστήσω πρώτα από όλους την οικογένειά μου για την υποστήριξη που μου παρείχε όλα αυτά τα χρόνια. Χωρίς τους ανθρώπους αυτούς δεν θα ήταν δυνατό το ταξίδι αυτό. Στην συνέχεια θα ήθελα να ευχαριστήσω του φίλους μου, η κατανόηση και η υποστήριξη που μου παρείχαν ήταν σημαντική σε όλον αυτόν τον αγώνα διάρκειας πέντε ετών. Τέλος θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας, κύριο Παναγιώτη Σαρηγιαννίδη, για την καθοδήγηση και τις υποδείξεις που μου παρείχε. Ιδιαίτερα θα ήθελα να πω ένα μεγάλο ευχαριστώ στον Παναγιώτη Ράδογλου-Γραμματίκη για την βοήθεια που μου προσέφερε.

Περίληψη

Το θέμα της διπλωματικής εργασίας αφορά στην προστασία Κρίσιμων Υποδομών όπως εργοστάσια παραγωγής Ηλεκτρικής Ενέργειας και Βιομηχανικές εγκαταστάσεις. Οι κρίσιμες υποδομές είναι ένα χρήσιμο κομμάτι της κοινωνίας. Χωρίς τις υπηρεσίες που προσφέρουν, καθημερινές εργασίες γίνονται δύσκολες έως και ακατόρθωτες. Για το λόγο αυτό είναι σημαντική η προστασία των υποδομών αυτών και των λειτουργιών τους.

Σκοπός της διπλωματικής εργασίας ήταν η δημιουργία ενός πλαισίου Αρχιτεκτονικής το οποίο κάποιος μπορεί να ακολουθήσει και να επιτύχει προστασία των υποδομών δικτύων που υπάρχουν στις εγκαταστάσεις αυτές. Για την επίτευξη του στόχου αυτού γίνεται χρήση δύο σημαντικών εργαλείων. Πρώτον η τεχνολογία των δικτύων οριζόμενων από λογισμικό και δεύτερον η τεχνολογία των παγίδων εισβολών.

Αποτέλεσμα της διπλωματικής αυτής είναι το σύστημα GuardianNet. Χρησιμοποιώντας της τεχνολογία δικτύων οριζόμενων από λογισμικό είναι σε θέση δυναμικά και σε μηδενικό σχεδόν χρονικό διάστημα να αλλάξει τα μονοπάτια επικοινωνίας δίνοντας απεριόριστες δυνατότητες στον διαχειριστή του συστήματος. Από την άλλη οι παγίδες εισβολών χρησιμοποιούνται για την ανακατεύθυνση της κίνησης αυτής αφήνοντας την πραγματική υποδομή ασφαλή από επιθέσεις. Παράλληλα απασχολούν τον επιτιθέμενο δίνοντας την ψευδαίσθηση ότι πρόκειται για την ίδια συσκευή καταγράφοντας κάθε αλληλεπίδραση με τον επιτιθέμενο.

Η διπλωματική εργασία αρχικά επικεντρώνεται στην παρουσίαση των τεχνολογιών αυτών και των δυνατοτήτων τους ενώ στην συνέχεια παρουσιάζεται η δομή και κάποιες οδηγίες για την υλοποίηση του GuardianNet. Πιο συγκεκριμένα παρουσιάζονται τα πλεονεκτήματα και μειονεκτήματα των παγίδων εισβολών και γίνεται ταξινόμηση τους με βάση τα χαρακτηριστικά που φέρουν. Επίσης παρουσιάζεται η τεχνολογία των δικτύων οριζόμενων από λογισμικό, η διαφορές που έχει σε σχέση με τα παραδοσιακά δίκτυα και τα πλεονεκτήματα που φέρει αυτή η αρχιτεκτονική στην σύγχρονη εποχή. Τέλος παρουσιάζεται η Αρχιτεκτονική του προτεινόμενου συστήματος και γίνεται μια πρακτική υλοποίηση

Abstract

The topic of this diploma thesis regards the protection of Industrial Control Systems such as Electricity Generation Facilities and Industrial Facilities. Industrial Control Systems are considered an important part of today's society. Without their services, everyday tasks become difficult if not unfeasible one could say. Thus, it is important to protect these facilities and safeguard their operation.

The purpose of this diploma thesis is to create a Framework which one can use in order to establish a secure computer network in these facilities. This can be achieved with the combined usage of two technologies· software defined networks and honeypots.

The outcome of this thesis is called GuardianNet. By using software defined networks, GuardianNet is able to dynamically alter network paths in a near zero delay time offering the network operator omnipotence over the network. On the other hand, Honeypots are deployed and malicious traffic is redirected to them, leaving the operational network safe from malicious users. At the same time, they keep the malicious user occupied by giving the illusion that it is the real device they attack too.

The first part of this thesis in focused on the presentation of these two technologies and afterwards the GuardianNet framework is presented. More specifically the advantages and disadvantages are presented, followed by a classification based on some of their characteristics. Additionally, the software defined technology is presented, its key differences with the traditional networks and the benefits it brings in today’s networks. Finally, the Architecture of the proposed system is described and a proof of concept implementation follows.

Περιεχόμενα

Λίστα Σχημάτων

Λίστα Πινάκων

Πίνακας Ελληνικών Όρων

Ακρωνύμιο	Πλήρες Όνομα
ΔΟΛ	Δίκτυο Οριζόμενο από Λογισμικό
ΔτΠ	Διαδίκτυο των Πραγμάτων
ΕΔΗΕ	Έξυπνο Δίκτυο Ηλεκτρικής Ενέργειας

Πίνακας Αγγλικών Ακρωνυμίων

Ακρωνύμιο	Πλήρες όνομα
AMI	Advanced Metering Infrastructure
API	Application Programming Interface
ARP	Address Resolution Protocol
ASIC	Application Specific Integrated Circuit
DoS	Denial of Service
HIH	High Interaction Honeypot
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
LIH	Low Interaction Honeypot
MIH	Medium Interaction Honeypot
MTU	Master Terminal Unit
NOS	Network Operating System
REST	Representational State Transfer
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SDN	Software Defined Network
SG	Smart Grid
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer

TCP	Transmission Control Protocol
TLS	Transport Layer Secure
UDP	User Datagram Protocol
XML	Extensible Markup Language

1 Εισαγωγή

1.1 Κίνητρο και Στόχοι

Η επικοινωνία των συστημάτων αυτοματισμού και συστημάτων που βρίσκονται σε κρίσιμες υποδομές ξεκίνησε με χρήση σειριακών συνδέσεων. Πλέον, για λόγους οικονομίας και καλύτερης διαχείρισης, η επικοινωνία αυτή έχει μεταβεί σε δίκτυα Ethernet και Transmission Control Protocol / Internet Protocol - TCP/IP. Επίσης τα συστήματα αυτά έχουν χαμηλές υπολογιστικές ικανότητες ενώ μέθοδοι ασφάλειας, όπως η κρυπτογράφηση, δεν ήταν στις απαιτήσεις κατασκευής τους. Δεδομένων των παραπάνω τα συστήματα αυτά έχουν γίνει ευάλωτοι στόχοι.

Ένα τυπικό βιομηχανικό δίκτυο αποτελείται ένα μεγάλο πλήθος αισθητήρων υπεύθυνων για την εποπτεία και έλεγχο φυσικών υποδομών. Ο μη εξουσιοδοτημένος έλεγχος ενός τέτοιου αισθητήρα μπορεί να έχει πολλές επιπτώσεις στην οικονομία και την καθημερινότητα χιλιάδων ανθρώπων. Για το λόγο αυτό είναι σημαντική η προστασία των συστημάτων αυτών από ανεπιθύμητες ενέργειες.

Στόχος της διπλωματικής αυτής είναι η υλοποίηση ενός πλαισίου το οποίο είναι σε θέση να αποτρέψει επιθέσεις ενώ παράλληλα συγκεντρώνει όσο τον δυνατόν περισσότερες πληροφορίες για τις ενέργειες του επιτιθέμενου.

1.2 Δίκτυα Κρίσιμων Υποδομών

Τα σύγχρονα ηλεκτρικά δίκτυα ακολουθώντας την τάση του Διαδικτύου των Πραγμάτων – ΔτΠ (Internet of Things – IoT) έχουν πλέον συνδεθεί με το διαδίκτυο εισάγοντας μια σειρά από πλεονεκτήματα. Η νέα αυτή αρχιτεκτονική ονομαζόμενη Έξυπνο Δίκτυο Ηλεκτρικής Ενέργειας - ΕΔΗΕ (Smart Grid -SG) [1] είναι σε θέση να προσφέρει καλύτερη διαχείριση της ενέργειας,

μεγαλύτερη αξιοπιστία, εξισορρόπηση φόρτου εργασίας και μειωμένες εκπομπές ρύπων. Το ΕΔΗΕ αποτελεί ένα συνδυασμό διάφορων τεχνολογιών όπως συστήματα ελέγχου και αυτοματισμού (Supervisory control and Data Acquisition - SCADA) και την υποδομή προηγμένων μετρήσεων (Advanced Metering Infrastructure - AMI). Τα συστήματα αυτά επιτρέπουν τον έλεγχο μηχανισμών και τη λήψη μετρήσεων από αισθητήρες χρησιμοποιώντας βιομηχανικά πρωτόκολλα όπως Modbus [2], και DNP3 [3] πάνω από τις υπάρχουσες υποδομές δικτύων IP. Τα συστήματα SCADA είναι διαδεδομένα στον βιομηχανικό τομέα και αποτελούνται από ένα σύνολο στοιχείων:

- Ο διαχειριστής: ο διαχειριστής είναι υπεύθυνος για την επίβλεψη και ομαλή λειτουργία ολόκληρού του συστήματος SCADA.
- Συσκευές αλληλεπίδρασης ανθρώπου συσκευών (Human Machine Interfaces - HMIs): οι συσκευές αυτές είναι υπεύθυνες για την μετάφραση των δεδομένων σε μια μορφή πιο κατανοητή στους διαχειριστές.
- Κεντρικές Τερματικές Συσκευές (Master Terminal Units - MTUs) και Απομακρυσμένες Τερματικές Συσκευές (Remote Terminal Units - RTUs): οι συσκευές αυτές δουλεύουν συνεργατικά μεταξύ τους για τον έλεγχο και λήψη πληροφοριών από τους αισθητήρες.

Λόγω της μεγάλης έκτασης των συστημάτων αυτών ο έλεγχος γίνεται απομακρυσμένα. Το ΕΔΗΕ ενώ έχει εισαγάγει πολλά πλεονεκτήματα έχει εισαγάγει και αρκετές προκλήσεις επίσης [4].

1.3 Σύνοψη

Η παρούσα διπλωματική εργασία δομείται σε έξι κεφάλαια. Τα κεφάλαια αυτά παρουσιάζουν τις τεχνολογίες, τα εργαλεία και τις ρυθμίσεις που έγιναν. Το παρόν κεφάλαιο

δίνει μια γενική εικόνα του αντικειμένου της διπλωματικής εργασίας και παρουσιάζει την δομή της διπλωματικής εργασίας.

Στο δεύτερο κεφάλαιο γίνεται παρουσίαση των παγίδων εισβολών οι οποίες ακολουθούν μια διαφορετική τακτική άμυνας. Παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα των παγίδων εισβολών και επίσης η ταξινόμησή τους με βάση κάποια κριτήρια.

Το τρίτο κεφάλαιο επικεντρώνεται στα δίκτυα οριζόμενα από το λογισμικό. Το πεδίο αυτό είναι αρκετά μεγάλο και αν γινόταν πλήρης ανάλυση σε αυτό θα επισκίαζε τα υπόλοιπα κεφάλαια της συγκεκριμένης διπλωματικής εργασίας. Για τον λόγο αυτό έχει γίνει σημαντική προσπάθεια στην παρουσίαση των βασικών και απαραίτητων εννοιών. Συγκεκριμένα γίνεται παρουσίαση των διαφορών που έχουν σε σχέση με τα παραδοσιακά δίκτυα υπολογιστών και των δυνατοτήτων που προσφέρουν. Επίσης στο τέλος γίνεται μια βιβλιογραφική ανασκόπηση της έρευνας που έχει γίνει τα τελευταία χρόνια στον συνδυασμό της τεχνολογίας αυτής και των παγίδων εισβολών.

Στο τέταρτο κεφάλαιο ακολουθεί η παρουσίαση του προτεινόμενου συστήματος. Συγκεκριμένα θα γίνει μια λεπτομερής παρουσίαση του συνδυασμού των δύο τεχνολογιών και έπειτα γίνεται μια παρουσίαση των ενεργειών που εκτελούνται για την ανακατεύθυνση δικτυακής κίνησης από τον προστατευόμενο πόρο προς μια παγίδα εισβολών.

Το πέμπτο κεφάλαιο περιγράφει μια τοπολογία σε περιβάλλον προσομοίωσης. Επίσης γίνεται παρουσίαση των εργαλείων που χρησιμοποιήθηκαν στην διπλωματική εργασία. Για κάθε εργαλείο παρουσιάζονται οι δυνατότητές του, παραμετροποιήσεις που έγιναν.

Στο τελευταίο κεφάλαιο παρουσιάζονται τα αποτελέσματα, τα συμπεράσματα και οι μελλοντικές επεκτάσεις της παρούσας διπλωματικής εργασίας.

2 Παγίδες Εισβολών

Οι παγίδες εισβολών είναι συστήματα λογισμικού ή ακόμα και φυσικές συσκευές που σκοπό έχουν να εκτρέψουν ή να αποτρέψουν μια επίθεση σε ένα πληροφοριακό σύστημα. Μια παγίδα εισβολών, σε αντίθεση με τις παραδοσιακές μεθόδους, δεν στοχεύει στην αποτροπή πρόσβασης του επιτιθέμενου αλλά στην παραπλάνηση του επιτιθέμενου [5]. Οι παγίδες εισβολών τοποθετούνται σε ένα δίκτυο μαζί με πραγματικές συσκευές και δίνουν την ψευδαίσθηση ενός μεγάλου δικτύου υπολογιστών. Στην πραγματικότητα όμως ένα μέρος αυτού του δικτύου αφορά σε εικονικούς χρήστες και οποιαδήποτε αλληλεπίδραση μαζί τους δεν πρόκειται να επιφέρει κάποια ουσιαστική γνώση στον επιτιθέμενο.

Πρώτη βιβλιογραφική αναφορά σε παγίδες εισβολών έγινε το 1990, ενώ μια δεκαετία αργότερα υπήρξε η πρώτη υλοποίηση [6]. Πέρα από τον κλασσικό ορισμό της προσέλκυσης επιτιθέμενων οι παγίδες εισβολών έχουν αποδειχθεί ως χρήσιμο εργαλείο καταπολέμησης επιθέσεων εξάπλωσης. Στον τομέα των κρίσιμών υποδομών οι παγίδες εισβολών έχουν σημαντική παρουσία [7].

Το κεφάλαιο αυτό ξεκινάει με την παρουσίαση των πλεονεκτημάτων και των περιορισμών που φέρουν οι παγίδες εισβολών και στην συνέχεια παρουσιάζεται η ταξινόμηση των παγίδων εισβολών.

2.1 Πλεονεκτήματα Παγίδων Εισβολών

Μια παγίδα εισβολών φέρει μερικά πλεονεκτήματα σε σχέση με άλλες μεθόδους αποτροπής ή ανίχνευσης εισβολών. Τα πλεονεκτήματα απορρέουν από τον τρόπο με τον οποίο λειτουργεί μια παγίδα εισβολών και το σκοπό που έχει. Μια παγίδα εισβολών είναι σε θέση να παράγει μικρότερο όγκο δεδομένων σε σχέση με αυτό που παράγει ένα σύστημα ανίχνευσης εισβολών ενώ τα δεδομένα αυτά είναι πιο συγκεντρωμένα και αφορούν μόνο την συγκεκριμένη

παγίδα εισβολών. Οποιαδήποτε αλληλεπίδραση με μια παγίδα εισβολών εξ ορισμού κατηγοριοποιείται ως ύποπτη οπότε κάθε καταγραφή που έχει μια παγίδα εισβολών μπορεί να θεωρηθεί ως ύποπτη. Από το γεγονός αυτό μπορεί κάποιος να εξαγάγει δύο σημαντικά συμπεράσματα [8]: 1. μια παγίδα εισβολών είναι σε θέση να παράγει μικρότερο αριθμό ψευδών θετικών εγγραφών και 2. μια παγίδα εισβολών είναι ικανή να εντοπίζει ψευδώς αρνητικές καταγραφές. Η σωστή κατηγοριοποίησης της δικτυακής κίνησης από συστήματα ανίχνευσης εισβολών είναι πολύ σημαντική. Όσον αφορά απαιτήσεις σε πόρους μια παγίδα εισβολών είναι σε θέση να εκτελείται χωρίς πρόβλημα και σε σύστημα με λιγοστούς πόρους. Αντίθετα τείχη προστασίας και συστήματα ανίχνευσης εισβολών απαιτούν αρκετούς πόρους λόγω του μεγάλου όγκου κίνησης που πρέπει να επεξεργάζονται σε πραγματικό χρόνο.

2.2 Μειονεκτήματα Παγίδων Εισβολών

Όπως κάθε εργαλείο έτσι και οι παγίδες εισβολών εμφανίζουν και αυτές μειονεκτήματα και περιορισμούς. Πρώτο και κυριότερο μειονέκτημα είναι ότι μια παγίδα εισβολών είναι στην ουσία λογισμικό και όπως κάθε λογισμικό μπορεί να έχει σφάλματα και κενά ασφάλειας. Σε περίπτωση που κάποιος επιτιθέμενος καταφέρει να αποκτήσει πρόσβαση σε μια παγίδα εισβολών τότε η ίδια η παγίδα γίνεται ένα εργαλείο επιθέσεων. Οι διαχειριστές συστημάτων πρέπει να λαμβάνουν υπόψιν τους αυτό το ενδεχόμενο. Το πεδίο ‘όρασης’ μιας παγίδα εισβολών περιορίζεται μόνο στην αλληλεπίδραση που έχει με τον οποιοδήποτε. Αν κάποιος επιτιθέμενος γνωρίζει ότι το συγκεκριμένο σύστημα είναι παγίδα εισβολών μπορεί πολύ εύκολα να το αποφύγει. Μια παγίδα εισβολών είναι χρήσιμη μόνο όταν ο επιτιθέμενος δεν έχει καμία επίγνωση της ύπαρξής της. Σημαντικό ζήτημα είναι η τοποθέτηση και το πλήθος των παγίδων εισβολών σε ένα δίκτυο[9].

2.3 Ταξινόμηση με βάση το επίπεδο αλληλεπίδρασης

Ανάλογα με το επίπεδο αλληλεπίδρασης που μπορεί να προσομοιώσει μια παγίδα εισβολών μπορεί να καταταχθεί σε μία από τις ακόλουθες κατηγορίες. Είναι σημαντικό να σημειωθεί ότι όσο ανεβαίνει το επίπεδο αλληλεπίδρασης ανεβαίνουν η πολυπλοκότητα, το επίπεδο αλληλεπίδρασης που μπορεί να υποστηρίξει η παγίδα εισβολών αλλά και το κόστος συντήρησης της.

2.3.1 Παγίδες Εισβολών Χαμηλής Αλληλεπίδρασης

Οι παγίδες εισβολών χαμηλής αλληλεπίδρασης (Low Interaction Honeypots - LIHs) συχνά υλοποιούν ένα μικρό κομμάτι των υποστηριζόμενων λειτουργιών των υπηρεσιών που υποστηρίζουν, ενώ η πρόσβαση στο Λειτουργικό Σύστημα δεν υπάρχει [10]. Στόχο έχουν τον εντοπισμό αυτοματοποιημένων εργαλείων επιθέσεων και την εξαγωγή απλών στατιστικών στοιχείων. Οι παγίδες εισβολών αυτής της κατηγορίας έχουν χαμηλή πολυπλοκότητα και η ρύθμισή τους είναι εύκολη. Παρόλα αυτά λόγω του μικρού αριθμού των λειτουργιών ή ακόμα και της απουσίας μερικών από αυτές είναι εύκολο κάποιος επιτιθέμενος να καταλάβει ότι πρόκειται για κάποια παγίδα εισβολών και να την αποφύγει σε σύντομο χρονικό διάστημα. Τέλος οι παγίδες εισβολών αυτής της κατηγορίας δεν είναι σε θέση να καταγράψουν ιδιαίτερα πολύτιμες πληροφορίες ενώ η χρήση τους είναι κυρίως εμπορική και όχι τόσο ερευνητική.

2.3.2 Παγίδες Εισβολών Μεσαίας Αλληλεπίδρασης

Η κατηγορία των Παγίδων Εισβολών Μεσαίας Αλληλεπίδρασης (Medium Interaction Honeypots - MIHs) έρχεται να επεκτείνει τις παγίδες εισβολών χαμηλής αλληλεπίδρασης προσφέροντας περισσότερες δυνατότητες και η παραμετροποίησή τους είναι πιο δύσκολη σε σχέση με την προηγούμενη κατηγορία. Οι παγίδες εισβολών αυτής της κατηγορίας υλοποιούν

περισσότερες λειτουργίες σε σχέση με την προηγούμενη κατηγορία, όμως και εδώ δεν υπάρχει πρόσβαση στο Λειτουργικό Σύστημα. Αυτό έχει σαν συνέπεια να υπάρχει μεγαλύτερη αλληλεπίδραση με τον επιτιθέμενο αλλά και πάλι κάποιος έμπειρος χρήστης είναι σε θέση να καταλάβει ότι πρόκειται για παγίδα εισβολών. Τέλος τα αρχεία καταγραφών έχουν περισσότερες πληροφορίες σε σχέση με αυτά την προηγούμενης κατηγορίας.

2.3.3 Παγίδες Εισβολών Υψηλής Αλληλεπίδρασης

Τελευταία κατηγορία είναι αυτή των παγίδων εισβολών Υψηλής Αλληλεπίδρασης (High Interaction Honeypots - HIH). Σε αυτή την κατηγορία μια παγίδα εισβολών πρέπει να είναι σε θέση να προσομοιώνει εξολοκλήρου μια υπηρεσία ή ακόμα και ένα ολόκληρο Λειτουργικό Σύστημα. Σαν επιτιθέμενος κάποιος θα πρέπει να μην μπορεί εύκολα να ξεχωρίσει μια παγίδα εισβολών της κατηγορίας αυτής από ένα πραγματικό σύστημα. Οι παγίδες εισβολών της κατηγορίας αυτής είναι οι πιο δύσκολες στην παραμετροποίηση λόγω των περίπλοκών υπηρεσιών που προσφέρουν. Όσον αφορά τα αρχεία καταγραφών που παράγουν είναι αυτά που έχουν την μεγαλύτερη αξία για ανάλυση. Παγίδες εισβολών αυτής της κατηγορίας συχνά χρησιμοποιούνται για την εύρεση zero-day ευπαθειών.

2.4 Ταξινόμηση με βάση το πεδίο λειτουργίας

Ανάλογα με το πεδίο εφαρμογής τους οι παγίδες εισβολών κατατάσονται σε δύο κατηγορίες: 1. Παγίδες Εισβολών Ερευνητικού Σκοπού (Research Honeypots) και 2. Παγίδες Εισβολών Εμπορικού Σκοπού (Production Honeypots). Κάθε κατηγορία φέρει διαφορετικά χαρακτηριστικά ανάλογα με το πεδίο εφαρμογής τους και το κύριο σκοπό που έχουν.

2.4.1 Παγίδες Εισβολών Ερευνητικού σκοπού

Οι Παγίδες Εισβολών Ερευνητικού Σκοπού έχουν ως στόχο την συλλογή όσο το δυνατόν περισσότερων δεδομένων ώστε μετά να γίνει ανάλυση αυτών. Συχνά χρησιμοποιούνται από ερευνητές και ειδικούς Ασφάλειας Δικτύων για την μελέτη κινήσεων των επιτιθέμενων και υλοποίηση αντίμετρων αυτών. Σε αυτή την κατηγορία υπάρχουν μόνο Παγίδες Υψηλού Επιπέδου Αλληλεπίδρασης ενώ η παραμετροποίησή τους είναι δύσκολη.

2.4.2 Παγίδες Εισβολών Εμπορικού σκοπού

Οι Παγίδες Εισβολών Εμπορικού Σκοπού έχουν ως κύριο στόχο την αποτροπή μιας επίθεσης προς μια πραγματική συσκευή στο δίκτυο παρουσιάζοντας τον εαυτό τους ως ένα καλύτερο στόχο για τον επιτιθέμενο. Η παραμετροποίησή τους είναι εύκολη και συνήθως είναι Χαμηλής ή Μέτριας Αλληλεπίδρασης. Αρνητικά των παγίδων εισβολών της κατηγορίας αυτής είναι ότι τα αρχεία καταγραφών που παράγονται δεν έχουν σημαντική πληροφορίες σε αντίθεση με τις Παγίδες Εισβολών Ερευνητικού Σκοπού.

2.5 Διαπιστευτήρια Παγίδες Εισβολών

Ως παγίδα εισβολών μπορεί να θεωρηθεί και μια ψηφιακή καταγραφή. Τα honeytokens, αποτελούν στοιχεία αυθεντικοποίησης τα οποία χρησιμοποιούνται για πρόσβαση σε άλλα συστήματα. Υπό κανονικές συνθήκες κάποιος χρήστης δεν πρέπει να ανακτήσει ή να χρησιμοποιήσει τις πληροφορίες που περιέχουν για οποιοδήποτε σκοπό, όπως για παράδειγμα να συνδεθεί σε ένα μηχάνημα απομακρυσμένα. Η χρήσης τους θεωρείται επίσης ύποπτη.

3. Δίκτυα Οριζόμενα από Λογισμικό

Τα Δίκτυα Οριζόμενα από Λογισμικό (ΔΟΛ) (Software Defined Networks – SDNs) αποτελούν ένα νέο πρότυπο διαχείρισης δικτύων το οποίο ακολουθεί μια διαφορετική αρχιτεκτονική σε σχέση με τα παραδοσιακά δίκτυα υπολογιστών. Προτάθηκε πρώτη φορά το 2008 από ερευνητές των Πανεπιστημίων του Stanford και Berkeley [11] και ως στόχο έχει τον κεντρικοποιημένο έλεγχο του δικτύου ενώ παράλληλα δημιουργεί ένα επίπεδο αφαίρεσης του υλικού που χρησιμοποιείται.

Στο κεφάλαιο αυτό γίνεται παρουσίαση των επιπέδων ενός δικτύου υπολογιστών και των λειτουργιών τους. Στην συνέχεια παρουσιάζεται η δομή ενός παραδοσιακού δικτύου και παραθέτονται τα πλεονεκτήματα και οι περιορισμοί της προσέγγισης αυτής. Επίσης θα γίνει ανάλυση της αρχιτεκτονικής ενός ΔΟΛ και τέλος ακολουθεί η βιβλιογραφική ανασκόπηση της έρευνας που έχει γίνει τα τελευταία χρόνια στον τομέα της ασφάλειας δικτύων υπολογιστών η οποία συνδυάζει ΔΟΛ και παγίδες εισβολών.

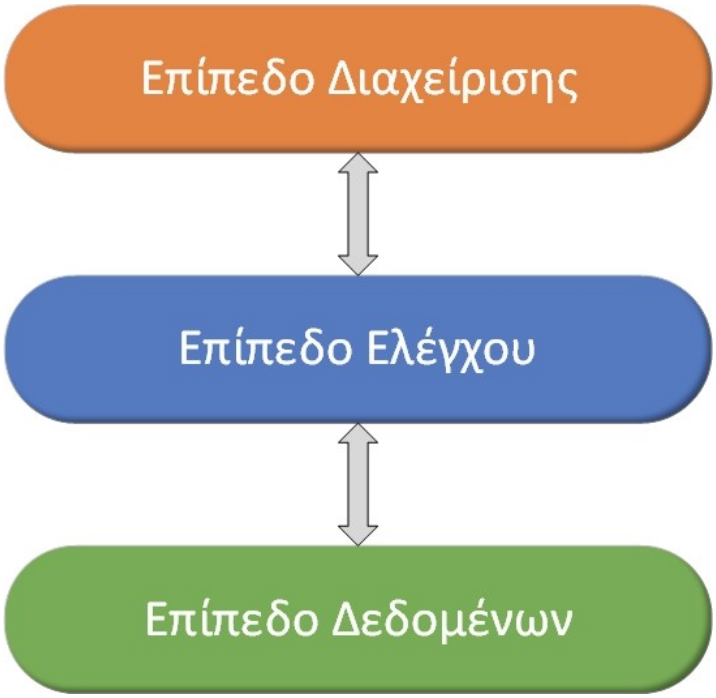
3.1 Τα Επίπεδα ενός Δικτύου Υπολογιστών

Κάθε δίκτυο Υπολογιστών μπορεί να χωριστεί σε τρία λογικά επίπεδα. Τα επίπεδα αυτά εκτελούν διαφορετικές λειτουργίες και επικοινωνούν με το γειτονικό τους επίπεδο. Το *Σχήμα 1* παρουσιάζει τα τρία αυτά επίπεδα. Όσο ανεβαίνει κάποιος επίπεδο τόσο απομακρύνεται από το φυσικό επίπεδο και πλησιάζει προς το διαχειριστικό μέρος του δικτύου. Οι λειτουργίες των επιπέδων αυτών εξηγούνται πιο αναλυτικά παρακάτω:

- Το πρώτο επίπεδο είναι το **επίπεδο προώθησης δεδομένων (Forwarding Plane)** ή αλλιώς **επίπεδο δεδομένων (Data Plane)** και είναι υπεύθυνο για την προώθηση των πλαισίων από την πηγή μέχρι και τον προορισμό τους. Οι συσκευές σε αυτό το επίπεδο

μπορεί να είναι φυσικές συσκευές ή και πακέτα λογισμικού που εκτελούν λειτουργίες δρομολόγησης όπως εικονικοί μεταγωγείς (virtual switches) και εικονικοί δρομολογητές (virtual routers). Για κάθε εισερχόμενο πλαίσιο γίνεται αναζήτηση σε εσωτερικές δομές και λαμβάνεται μια απόφαση για την θύρα εξόδου από την οποία θα φύγει το πλαίσιο. Ειδικές περιπτώσεις πλαισίων μπορεί να χρειαστεί να προωθηθούν προς ανώτερα επίπεδα.

- Δεύτερο είναι το **επίπεδο ελέγχου (Control Plane),** που είναι υπεύθυνο για την δημιουργία των κανόνων δρομολόγησης. Αρμοδιότητα του επιπέδου αυτού είναι να καθοδηγεί το επίπεδο δεδομένων συμπληρώνοντας σωστά πίνακες δρομολόγησης ή παρόμοιες δομές τις οποίες συμβουλεύεται το επίπεδο δεδομένων. Πρωτόκολλα όπως το Open Shortest Path First - OSPF ή Border Gateway Protocol - BGP ανήκουν σε αυτό το επίπεδο.
- Τελευταίο επίπεδο είναι το **επίπεδο διαχείρισης (Management Plane)**. Το επίπεδο αυτό χρησιμοποιείται από διαχειριστές για την ρύθμιση ή τροποποίηση των δικτυακών συσκευών. Επίσης στο επίπεδο αυτό γίνεται και ο συνολικός έλεγχος του δικτύου. Για παράδειγμα ο διαχειριστής μπορεί να αποφασίσει να αποτρέψει την επικοινωνία μεταξύ δύο υποδικτύων. Στο επίπεδο αυτό συναντώνται πρωτόκολλα όπως Secure Shell Protocol - SSH, Simple Network Management Protocol - SNMP) ή και γραφικές διεπαφές.

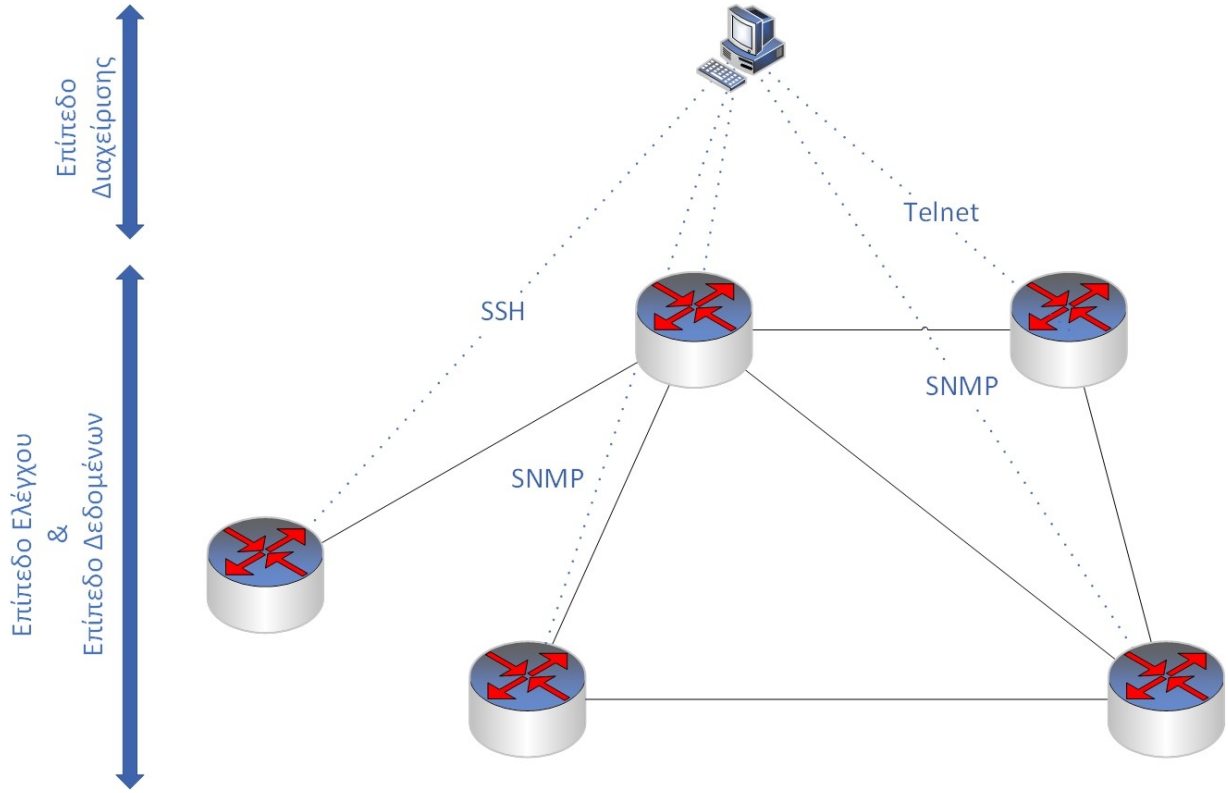


Σχήμα 1 Τα επίπεδα ενός δικτύου Υπολογιστών

3.2 Αρχιτεκτονική Παραδοσιακών Δικτύων Υπολογιστών

Σε ένα παραδοσιακό δίκτυο υπολογιστών τα επίπεδα προώθησης δεδομένων και ελέγχου συνυπάρχουν μαζί σε κάθε δικτυακή συσκευή. Το *Σχήμα 2* παρουσιάζει με γραφικό τρόπο την δομή των πεδίων σε ένα παραδοσιακό δίκτυο. Ο σχεδιασμός αυτός οδηγεί σε ένα δίκτυο πιο ανθεκτικό σε σφάλματα κάτι που ήταν σημαντικό παλαιότερα. Μειονέκτημα όμως αυτής της προσέγγισης είναι ότι οδηγεί σε πολύπλοκα δίκτυα τα οποία είναι στατικά. Η ύπαρξη επιπρόσθετων δικτυακών οντοτήτων όπως συστήματα ανίχνευσης εισβολών και τειχών προστασίας επίσης συνεισφέρει αρνητικά σε αυτό το πρόβλημα. Ειδικά στον τομέα της Ασφάλειας δικτύων οι αλλαγές της πολιτικής και των κανόνων είναι ένα λεπτό ζήτημα που πολλοί θέλουν να αποφύγουν. Μια μικρή αλλαγή μπορεί να επιφέρει προβλήματα σε πολλά μέρη του δικτύου, ενώ η εκ των προτέρων γνώση των επιπτώσεων αυτών είναι δύσκολη

διαδικασία [11]. Ως παράδειγμα μπορεί κάποιος να θεωρήσει την προσθήκη ενός νέου χρήστη σε ένα μεγάλο δίκτυο. Για την σωστή επικοινωνία ο διαχειριστής πρέπει να τροποποιήσει κανόνες σε τείχη προστασίας και να αλλάξει πιθανόν κανόνες στο σύστημα ανίχνευσης εισβολών ώστε να μην θεωρεί τον χρήστη αυτόν ως ύποπτο. Τέλος σε ένα σύγχρονο δίκτυο υπάρχει σημαντικά αυξανόμενο πλήθος συσκευών. Από τα παραπάνω μπορεί κάποιος να συμπεράνει πως τα παραδοσιακά δίκτυα φτάνουν στα όρια των επιδόσεών τους. Σύντομα θα υπάρξει ένα σημείο όπου το κόστος συντήρησης θα ξεπεράσει το κέρδος που αποφέρουν.



Σχήμα 2 Αρχιτεκτονική ενός παραδοσιακού Δικτύου Υπολογιστών

3.3 Αρχιτεκτονική ενός Δικτύου Οριζόμενου από Λογισμικό

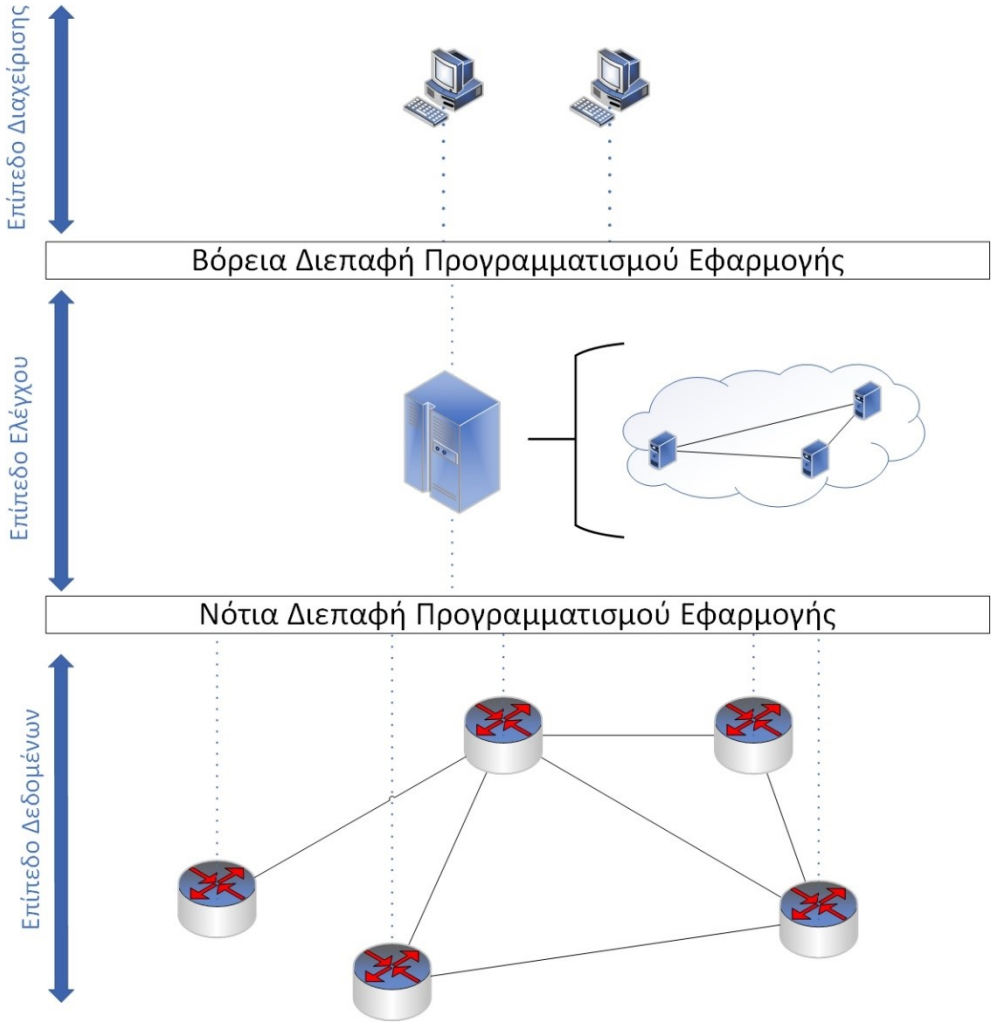
Ένα δίκτυο οριζόμενο από το λογισμικό χωρίζει τα επίπεδα προώθησης και δρομολόγησης που προηγουμένως ήταν αλληλένδετα μεταξύ τους. Το *Σχήμα 3* παρουσιάζει τον διαχωρισμό

αυτό ο οποίος εκ πρώτης όψεως φαίνεται να μην προσφέρει κάτι σημαντικό. Παρόλα αυτά στην συνέχεια θα διαπιστωθεί ότι με αυτό τον διαχωρισμό δημιουργούνται πολλές δυνατότητες [12]. Σαν αρχιτεκτονική τα δίκτυα οριζόμενα από λογισμικό έχουν τέσσερεις βασικούς πυλώνες:

- Διαχωρισμός των επιπέδων προώθησης δεδομένων και ελέγχου. Ο διαχωρισμός αυτός επιτρέπει σε ένα τέτοιο δίκτυο να είναι πιο ευέλικτο.
- Ύπαρξη ενός κεντρικού σημείου ελέγχου του δικτύου που ονομάζεται ελεγκτής. Το σημείο αυτό επιτρέπει τον συνολικό έλεγχο του δικτύου. Ο διαχειριστής αντιμετωπίζει τον ελεγκτή ενώ στην πραγματικότητα μπορεί να είναι ένα σύνολο από συστήματα υπολογιστών που δρουν συνεργατικά μεταξύ τους.
- Υποστήριξη προγραμματισμού του δικτύου. Ο προγραμματισμός του δικτύου επιτρέπει στους διαχειριστές να αυτοματοποιούν διαδικασίες και συμβάντα με χρήση λογισμικού.
- 'Υπαρξη ανοιχτών προτύπων επικοινωνίας. Η χρήση ανοιχτών προτύπων έρχεται να αντιμετωπίσει το κλασικό πρόβλημα των παραδοσιακών δικτύων όπου κάθε προμηθευτής παρέχει διαφορετικές δυνατότητες για τον εξοπλισμός που προσφέρει.

Στην πιο καθαρή του μορφή ένα δίκτυο οριζόμενο από λογισμικό αποτελείται από δύο μόνο οντότητες: 1. τις δικτυακές συσκευές, ονόματι μεταγωγέας ΔΟΛ (SDN switches) και 2. τον ελεγκτή (Controller) ή αλλιώς λειτουργικό σύστημα δικτύου (Network Operating System – NOS). Οι δύο αυτές οντότητες επικοινωνούν μεταξύ τους με χρήση προκαθορισμένων πρωτοκόλλων. Οι δικτυακές συσκευές έχουν μόνο λειτουργίες της προώθησης των δεδομένων. Είναι σημαντικό να τονιστεί ότι σε ένα καθαρό (ΔΟΛ) δίκτυο δεν υπάρχουν οι έννοιες δρομολογητής, μεταγωγέας ή τείχος προστασίας. Αντί αυτού υπάρχει μόνο μια οντότητα: ο μεταγωγέας δικτύου οριζόμενου από λογισμικό (SDN switch). Οι μεταγωγείς αυτοί δεν είναι σε θέση να πάρουν αυθαίρετα απόφαση για κάποιο πλαίσιο. έχουν μια συλλογή από κανόνες, ενώ σε περίπτωση που δεν γνωρίζουν πώς να χειριστούν ένα πλαίσιο πρέπει να ενημερώσουν τον ελεγκτή με χρήση προκαθορισμένων μεθόδων. Η επικοινωνία με τον ελεγκτή γίνεται χρησιμοποιώντας

συγκεκριμένα πρωτόκολλα και μεθόδους. Ο ελεγκτής αποτελεί τον ‘εγκέφαλο’ ενός δικτύου οριζόμενου από το λογισμικό. Σκοπός του είναι να συντονίζει την λειτουργία των συσκευών δικτύου και επιπρόσθετα να προσφέρει διεπαφή προγραμματισμού εφαρμογής- ΔΠΕ (Application Programming Interface – API) προς τους διαχειριστές.



Σχήμα 3 Αρχιτεκτονική ενός Δικτύου Οριζόμενου από Λογισμικό

3.3.1 Ο Ελεγκτής

Ο ελεγκτής αποτελεί κομβικό σημείο ενός δικτύου οριζόμενο από λογισμικό. Πολύπλοκες πολιτικές που επιθυμεί να εφαρμόσει ο διαχειριστής από το επίπεδο διαχείρισης στέλνονται προς τον ελεγκτή ο οποίος με την σειρά του πρέπει να είναι σε θέση να μεταφράσει τις εντολές αυτές σε εντολές κατανοητές από το επίπεδο δεδομένων. Ο ελεγκτής μπορεί να αποτελείται από ένα μοναδικό κόμβο, η αρχιτεκτονική του οποίου λέγεται ‘κεντρικοποιημένη’ ενώ μπορεί

να αποτελείτε και από ένα σύνολο κόμβων, η αρχιτεκτονική της οποίας ονομάζεται ‘μη κεντρικοποιημένη’. Και στις δύο περιπτώσεις ο ελεγκτής πρέπει να παρουσιάζεται ως μια οντότητα στο δίκτυο. Στην περίπτωση του ‘κεντρικοποιημένου’ ελεγκτή οι διαδικασίες της διαχείρισης και της εκκίνησης του ελεγκτή είναι σχετικά απλές όμως σε περίπτωση που τεθεί εκτός λειτουργίας ο ελεγκτής το ΔΟΣ βγει εκτός λειτουργίας. Από την άλλη μια ‘μη κεντρικοποιημένη’ αρχιτεκτονική παρέχει ένα πιο ανθεκτικό δίκτυο ενώ παράλληλα ο φόρτος εργασίας μπορεί να εξισορροπείται στους κόμβους του ελεγκτή. Η αρχιτεκτονική αυτή όμως εισάγει αυξημένη πολυπλοκότητα στο δίκτυο.

Η επικοινωνία με τον ελεγκτή ορίζεται κάθετα, μεταξύ των επιπέδων διαχείρισης και δεδομένων, και οριζόντια σε περίπτωση που ο ελεγκτής είναι ‘μη κεντρικοποιημένος’. Η επικοινωνία με το επίπεδο δεδομένων ορίζει μια διεπαφή, που ονομάζεται νότια διεπαφή (Southbound Interface). Το επίπεδο δεδομένων επικοινωνεί με το ελεγκτή χρησιμοποιώντας καλώς ορισμένα πρωτόκολλα όπως NETCONF και OpenFlow. Η διεπαφή επικοινωνίας με το επίπεδο διαχείρισης, που ονομάζεται βόρεια διεπαφή (Northbound Interface) γίνεται χρήση γνωστών μεθόδων επικοινωνίας. Η βασική ιδέα του ΔΟΛ είναι το λογισμικό να καθοδηγεί και να αλλάζει την δομή του δικτύου σύμφωνα με περιορισμούς που ορίζει ο διαχειριστής. Υπηρεσίες Representational state transfer - REST επικρατούν στην πλειοψηφία των ελεγκτών. Το REST αποτελεί ένα σύνολο από αρχές σχεδίασης μιας δικτυακής υπηρεσίας που επικεντρώνει στους πόρους ενός συστήματος. Μια υπηρεσία REST στηρίζεται στο πρωτόκολλο Hypertext Transfer Protocol - HTTP και των μεθόδων του για την επικοινωνία μεταξύ των πελατών και του διακομιστή που υλοποιεί την υπηρεσία αυτή. Έχει γίνει ευρέως αποδεκτό από την κοινότητα των προγραμματιστών και είναι ένας από τους κύριους λόγους όπου υπάρχει και υποστήριξη από όλους τους σύγχρονους ελεγκτές. Τέλος η ορίζονται επικοινωνία, που αλλιώς ονομάζεται και ανατολική-δυτική διεπαφή (East-West Interface) ορίζεται σε μη κεντρικοποιημένες αρχιτεκτονικές ελεγκτών και αφορά την επικοινωνία μεταξύ κόμβων του ελεγκτή.

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά γνωστοί ελεγκτές ΔΟΛ. Εύκολα μπορεί κάποιος να δει ότι η πλειοψηφία αυτών είναι 'μη κεντρικοποιημένοι' ελεγκτές. Οι ανάγκες των σύγχρονων δικτύων απαιτούν να υπάρχει ανθεκτικότητα και μεγάλη περίοδος λειτουργία χωρίς προβλήματα. Επίσης όλες οι διεπαφές με το επίπεδο διαχείρισης είναι REST. Τέλος οι γλώσσες προγραμματισμού που επικρατούν είναι γλώσσες υψηλού επιπέδου.

Πίνακας 1 Γνωστοί Ελεγκτές Δικτύων Οριζόμενων από Λογισμικό

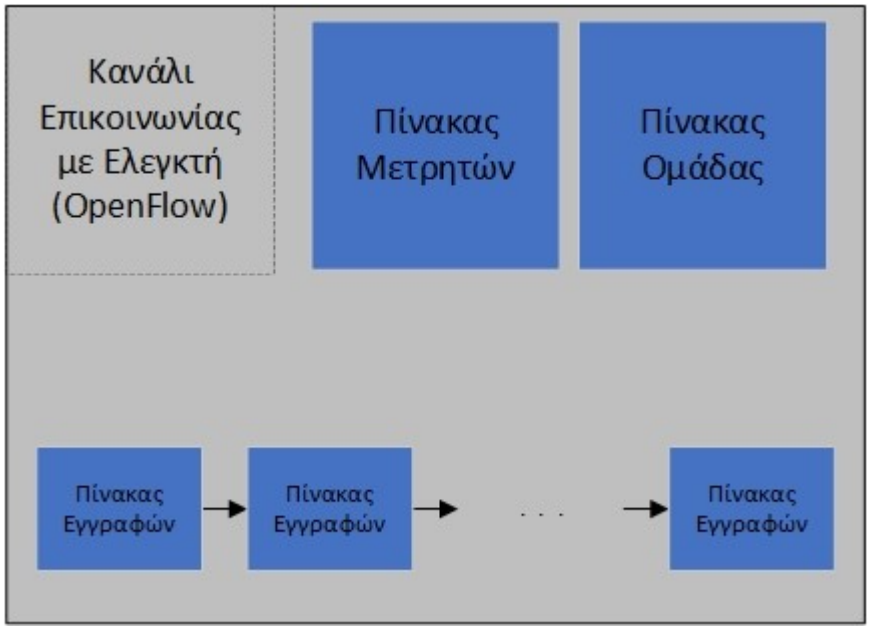
Όνομα	Αρχιτεκτονική	N-API	OpenFlow version	Γλώσσα Προγραμματισμού
Ryu **[13]**	Κεντρικοποιημένη	REST	1.0 - 1.5	Python3
ONOS **[14]**	Μη Κεντρικοποιημένη	REST	1.0, 1.3	Java
OpenDaylight **[15]**	Μη Κεντρικοποιημένη	REST	1.0, 1.3	Java
Floodlight **[16]**	Μη Κεντρικοποιημένη	REST	1.0, 1.3	Java
OpenMul **[17]**	Μη Κεντρικοποιημένη	REST	1.0	C
NOX/POX **[18]**	Κεντρικοποιημένη	ad-hoc	1.0	C++ / Python

3.3.2 Μεταγωγέας Οριζόμενος από Λογισμικό

Δεύτερη οντότητα και εξίσου σημαντική με τον ελεγκτή είναι ο μεταγωγέας οριζόμενος από το λογισμικό. Ένας μεταγωγέας οριζόμενος από το λογισμικό, όπως και ένας κλασικό μεταγωγέας δεδομένων, έχει πίνακες δρομολόγησης όπως εικονίζονται στο *Σχήμα 4* τους οποίους συμβουλεύεται για την επιλογή διεπαφής εξόδου ενός εισερχόμενου πλαισίου. Όπως και οι παραδοσιακές δικτυακές συσκευές, έτσι και οι μεταγωγείς αυτοί έχουν ολοκληρωμένα

κυκλώματα ειδικού σκοπού (Application Specific Integrated Circuits – ASICs) για την ταχύτερη επεξεργασία των δεδομένων [19].

Η διαφορά με τον παραδοσιακό δικτυακό εξοπλισμό είναι ότι ένας μεταγωγέας οριζόμενος από το λογισμικό δεν συμπληρώνει εγγραφές στους εσωτερικούς πίνακες δρομολόγησης με τον ίδιο τρόπο. Αντί αυτού για κάθε πλαίσιο το οποίο δεν μπορεί να ταυτιστεί με καμία εγγραφή στέλνεται προς τον ελεγκτή. Ο ελεγκτής έπειτα πρέπει να πάρει μια απόφαση για την θύρα εξόδου του πλαισίου αυτού ή ακόμα και αν το πλαίσιο αυτό θα απορριφθεί από τον μεταγωγέα.



Σχήμα 4 Δομή ενός Μεταγωγέα OpenFlow

3.4 Το πρωτόκολλο OpenFlow

Το OpenFlow είναι ένα πρωτόκολλο επικοινωνίας μεταξύ του ελεγκτή και μεταγωγέα ΔΟΛ και είναι το πιο ευρέως χρησιμοποιούμενο πρωτόκολλο Νότιας Διεπαφής Επικοινωνίας (Southbound Interface). Ως πρωτόκολλο επιπέδου εφαρμογής, στηρίζεται στο TCP για την μεταφορά της πληροφορίας ενώ ταυτόχρονα υποστηρίζει κρυπτογραφημένη επικοινωνία μέσω

Secure Socket Layer - SSL ή Transport Layer Secure - TLS. Επιθέσεις κατά του OpenFlow υπάρχουν και θα αυξηθούν με την όλο και αυξανόμενη χρήση του [20]. Από την αρχική του έκδοση 1.0, που δημοσιεύτηκε το 2010, μέχρι και την πιο πρόσφατη 1.5, που δημοσιεύτηκε το 2015, έχει γνωρίσει αρκετές τροποποιήσεις και βελτιώσεις, προσφέροντας περισσότερη ευελιξία στον διαχειριστή [21]. Η έκδοση 1.6 του πρωτοκόλλου αν και έχει γραφεί είναι διαθέσιμη σε κλειστές ομάδες προς το παρόν. Σαν πρωτόκολλο το OpenFlow πρέπει να υποστηρίζεται τόσο από τον ελεγκτή όσο και από τον μεταγωγέα ΔΟΛ. Ορίζει τους τύπους, το πλήθος και την δομή των μηνυμάτων που πρέπει να ανταλλάξουν ένας μεταγωγέας ΔΟΛ και ο ελεγκτής. Επίσης, προσδιορίζει τις δομές πινάκων που πρέπει να περιέχει ένας μεταγωγέας ΔΟΛ και περιγράφει τον τρόπο με τον οποίο ο ελεγκτής μπορεί να τροποποιήσει τις εγγραφές αυτές με χρήση κατάλληλων μηνυμάτων.

3.4.1 Εγγραφές OpenFlow

Ένας μεταγωγέας ΔΟΛ που είναι συμβατός με το πρωτόκολλο OpenFlow, γνωστός επίσης στην βιβλιογραφία και ως μεταγωγέας OpenFlow (OpenFlow Switch), έχει ένα σύνολο από πίνακες οι οποίοι περιέχουν εγγραφές OpenFlow (OpenFlow entries). Κάθε πίνακας από τους παραπάνω περιέχει εγγραφές οι οποίες συνδυαζόμενες μεταξύ τους μπορούν να εκφράσουν πολύπλοκες πολιτικές. Το OpenFlow ορίζει τρεις κατηγορίες πινάκων. Κάθε κατηγορία έχει διαφορετικούς σκοπούς, οι οποίοι περιγράφονται παρακάτω:

- **Πίνακες εγγραφών OpenFlow (OpenFlow Tables)**: πρόκειται για μια σειρά από πίνακες που δέχονται εγγραφές OpenFlow. Είναι αριθμημένοι ξεκινώντας από το 0 και ένας μεταγωγέας ξεκινάει πάντα από τον μικρότερο πίνακα προς τον μεγαλύτερο αναζητώντας μια εγγραφή που να ταιριάζει με τα στοιχεία του πλαισίου δεδομένων. Οι πίνακες αυτοί είναι γενικού σκοπού σε αντίθεση με τις υπόλοιπες δύο κατηγορίες.
- **Πίνακας Μετρητών (Meter Table)**: ένας πίνακας μετρητή χρησιμοποιείται για την εφαρμογή πολιτικής ποιότητας υπηρεσιών. Όπως και στο προηγούμενο υποκεφάλαιο,

ένα πλαίσιο δεδομένων μιας δικτυακής ροής που ταυτίζεται με μια εγγραφή του πίνακα θα ακολουθήσει την πολιτική που ορίζει η εγγραφή.

- **Πίνακας Ομάδων (Group Table)**: ο πίνακας αυτός επιτρέπει τη χρήση πιο σύνθετων μεθόδων προώθησης δεδομένων, όπως μηνύματα πολυδιανομής (multicast messages), μηνύματα που αφορούν περισσότερα του ενός συστήματα στο δίκτυο.

Μια εγγραφή OpenFlow ενός πίνακα εγγραφών έχει δομή με αυτή του *Σχήματος 5*. Τα πεδία αυτά επεξηγούνται παρακάτω:

- **Προτεραιότητα (priority)**: καθορίζει την σειρά με την οποία ο μεταγωγέας ΔΟΛ θα ψάξει τους κανόνες. Κανόνες με μεγαλύτερη προτεραιότητα θα ελεγχθούν πρώτοι. Οι έγκυρες τιμές είναι [0-65535].
- **Κανόνας αντιστοίχισης ή απλώς Κανόνας (Flow Rule)**: είναι το μέρος μιας εγγραφής το οποίο ορίζει τα πεδία ενός πλαισίου που πρέπει να ταιριάζουν ώστε να ενεργοποιηθεί ο κανόνας. Εάν ένα πλαίσιο ενεργοποιεί πολλούς κανόνες τότε ο μεταγωγέας ΔΟΛ θα σταματήσει στον πρώτο κανόνα. Ως πεδία ταιριάσματος το OpenFlow ορίζονται τα πεδία κεφαλίδας πολλών πρωτοκόλλων όπως Internet Control Message Protocol - ICMP, IP, TCP και User Datagram Protocol - UDP.
- **Ενέργειες (Actions):** ορίζει τις ενέργειες που πρέπει να εκτελέσει ο μεταγωγέας ΔΟΛ όταν ενεργοποιηθεί ο κανόνας. Οι ενέργειες αυτές μπορεί να περιλαμβάνουν απλές εντολές όπως αποστολή προς μια διεπαφή εξόδου, αλλαγές στις κεφαλίδες πρωτοκόλλων όπως για παράδειγμα αλλαγή της διεύθυνσης IP προέλευσης και επίσης ανακατεύθυνση σε άλλο πίνακα. Μια εγγραφή OpenFlow που έχει κενό το πεδίο ενεργειών θα απορρίπτει κάθε πλαίσιο που την ενεργοποιεί.
- **Χρόνοι Ακύρωσης εγγραφής (timeout counters)**: το OpenFlow ορίζει δύο τιμές αναμονής για κάθε εγγραφή. Εάν οποιαδήποτε από τις δύο φτάσει την οριακή τιμή τότε ο μεταγωγέας ΔΟΛ αφαιρεί από τον πίνακα την εγγραφή αυτή. Η πρώτη τιμή αναφέρεται

στο μέγιστο χρονικό διάστημα, μετά το πέρας του οποίου η εγγραφή αφαιρείται αυτόματα. Η τιμή αυτή ονομάζεται ως hard-timeout. Η δεύτερη τιμή ορίζει το μέγιστο χρονικό διάστημα αναμονής για ένα πλαίσιο να ενεργοποιήσει τον κανόνα αυτό. Μετά το πέρας του διαστήματος αυτού ο μεταγωγέας ΔΟΛ αφαιρεί την εγγραφή. Και στις τιμές αν δοθεί η τιμή 0 τότε θεωρείται ότι ο κανόνας έχει άπειρη χρονική διάρκεια.

- **Μετρητές (Counters)**: τελευταίο μέρος μιας εγγραφής είναι οι μετρητές. Οι μετρητές αυτοί περιλαμβάνουν πλήθος των πλαισίων που ενεργοποίησαν την εγγραφή αυτή, πλήθος bytes που έχουν επεξεργασθεί από την εγγραφή αυτή. Οι μετρητές αυτές είναι χρήσιμοι σε περιπτώσεις επιλογής διαγραφής μιας εγγραφής επειδή οι πίνακες είναι πλήρεις ή για περεταίρω ανάλυση των τιμών τους [22].
- **Cookie**: η τιμή αυτή επιλέγεται από τον ελεγκτή και χρησιμοποιείται για την εύκολη αναγνώριση και ταυτοποίηση εγγραφών. Για παράδειγμα εγγραφές που δημιουργήθηκαν για μια συγκεκριμένη λειτουργία και υπάρχουν σε πολλούς πίνακες μπορούν να έχουν την ίδια τιμή cookie.

Σχήμα 5 Δομής μίας εγγραφής OpenFlow

3.4.2 Διαγραφή εγγραφών OpenFlow

Η διαγραφή μιας εγγραφής OpenFlow γίνεται με δύο τρόπους. Αρχικά αν έχουν τεθεί χρόνοι ακύρωσης εγγραφής τότε η εγγραφή θα αφαιρεθεί αυτόματα μόλις ξεπεραστεί η χρονική διάρκεια που ορίζουν. Δεύτερη μέθοδος είναι μέσω του ελεγκτή. Για την διαγραφή της εγγραφής απαιτούνται μεταβλητές του πεδίου *ενέργειες.* Η διαγραφή γίνεται με σύγκριση των

δοθέντων τιμών. Αυτό σημαίνει ότι σε περίπτωση που δοθούν λιγότερες μεταβλητές υπάρχει περίπτωση να διαγραφούν περισσότερες της μιας εγγραφής.

3.4.3 Μηνύματα HELLO

Κάθε νέα σύνδεση ενός μεταγωγέα ΔΟΛ με έναν ελεγκτή ξεκινά με μηνύματα HELLO. Με τα μηνύματα αυτά μεταγωγέας και ελεγκτής καταλήγουν στην πιο υψηλή κοινώς υποστηριζόμενη έκδοση OpenFlow. Επίσης σε αυτή την φάση ο μεταγωγέας γνωστοποιεί στον ελεγκτή τις δυνατότητες που έχει και άλλες πληροφορίες . Από το σημείο αυτό και πέρα το κανάλι επικοινωνίας OpenFlow έχει εγκαθιδρυθεί. Ο ελεγκτής μπορεί αυθαίρετα να εισάγει νέες εγγραφές, να τροποποιήσει ή να διαγράψει υπάρχουσες εγγραφές χρησιμοποιώντας το πεδίο *Ενέργειες* ή το *Cookie* για την αναγνώρισή τους. Στο κανάλι επικοινωνίας αποστέλλονται σε τακτά χρονικά διαστήματα μηνύματα KEEP-ALIVE. Τα μηνύματα αυτά κρατούν το κανάλι επικοινωνίας ενεργό και παράλληλα σηματοδοτούν την διακοπή σύνδεσης σε περίπτωση που σταλεί απάντηση εντός κάποιων χρονικών ορίων.

3.4.4 Μηνύματα Packet In και Packet Out

Ερχόμενο ένα πλαίσιο αποθηκεύεται σε μια προσωρινή μνήμη μέχρι να παρθεί απόφαση δρομολόγησης. Ο μεταγωγέας ΔΟΛ αρχικά θα ψάξει τους πίνακες εγγραφών ξεκινώντας από τον πίνακα με τον μικρότερο αριθμό και ψάχνοντας τις εγγραφές με φθίνουσα σειρά ξεκινώντας από την εγγραφή με την υψηλότερη προτεραιότητα. Αν βρεθεί έγγραφή που ενεργοποιείται από το πλαίσιο αυτό τότε ο μεταγωγέας εκτελεί τις ενέργειες όπως αυτές περιγράφονται στο αντίστοιχο πεδίο. Σε περίπτωση που έχουν εξαντληθεί όλοι οι πίνακες χωρίς να έχει ενεργοποιηθεί κάποιος κανόνας, ένα μήνυμα packet in παράγεται και αποστέλλεται προς τον ελεγκτή.

Το μήνυμα packet in μπορεί να περιέχει ολόκληρο το πλαίσιο, μέρος αυτού ή κάποιο αναγνωριστικό το οποίο παράγει ο μεταγωγέας όταν πρόκειται να το αποθηκεύσει στην προσωρινή μνήμη του. Επιπρόσθετα στοιχεία που έχει αυτό το μήνυμα είναι μεταξύ άλλων το αναγνωριστικό του μεταγωγέα ΔΟΛ και την διεπαφή εισόδου. Ο ελεγκτής πρέπει σε προκαθορισμένο χρονικό πλαίσιο να αποφανθεί πως πρέπει να δρομολογηθεί το πλαίσιο αυτό. Το αποτέλεσμα είναι ένα μήνυμα packet out που περιέχει μια νέα εγγραφή OpenFlow και το αναγνωριστικό του πλαισίου ή το πλαίσιο ολόκληρο.

3.5 Χρήση ΔΟΛ με Παγίδες Εισβολών

Η ερευνητική κοινότητα προσπαθεί δραστήρια να συνδυάσει την ελαστικότητα που προσφέρουν τα ΔΟΛ με άλλες Τεχνολογίες. Στο πλαίσιο της ασφάλειας δικτύων και υπολογιστών έχουν γίνει αρκετές απόπειρες συνδυασμού της τεχνολογίας αυτής με παγίδες εισβολών. Πριν τη προτυποποίηση των ΔΟΛ η ερευνητική κοινότητα είχε ήδη συλλάβει την ιδέα της ανακατεύθυνσης κίνησης με σκοπό την ομαλή λειτουργία των συστημάτων [23] κάνοντας όμως κάποιους συμβιβασμούς.

Οι ερευνητές στο [24] προτείνουν ένα νέο σχήμα αρχιτεκτονικής που συνδυάζει την δυνατότητα καταγραφής των δεδομένων παγίδων εισβολών με την ελαστικότητα που προσφέρει ένα ΔΟΛ. Πιο συγκεκριμένα υπάρχουν δύο ομάδες παγίδων εισβολών, μία αποτελούμενη από παγίδες εισβολών χαμηλής αλληλεπίδρασης και μια ομάδας αποτελούμενη από παγίδες εισβολών υψηλής προτεραιότητας και η υποδομή ΔΟΛ αποτελούμενη από τον ελεγκτή και τον μεταγωγέα ΔΟΛ. Ανάλογα με τον τύπο επίθεσης η κίνηση κατευθύνεται προς την ανάλογη ομάδα. Αν η επίθεση είναι χαμηλού επιπέδου, όπως ανίχνευση θυρών, τότε προωθείται προς την πρώτη, ενώ πιο περίπλοκες επιθέσεις αφήνονται για την δεύτερη ομάδα. Το προτεινόμενο σύστημα είναι σε θέση να εκτελέσει την διαδικασία της ανακατεύθυνσης σε πολύ σύντομο χρονικό διάστημα χωρίς να γίνει αντιληπτό από το επιτιθέμενο.

Το HoneyMix [25] αποτελεί ένα έξυπνο δίκτυο παγίδων εισβολών αποτελούμενο από πέντε συστατικά στοιχεία. Είναι σε θέση να αποκρύπτει ευαίσθητες πληροφορίες που αλλιώς θα φανέρωναν ότι πρόκειται για παγίδα εισβολών, να αλλάξει δυναμικά δικτυακή κίνηση χωρίς να γίνει αντιληπτό από τον επιτιθέμενο, να επιλέγει την πιο ιδανική παγίδα εισβολών ανάλογα με τις κινήσεις του επιτιθέμενου χρησιμοποιώντας υποδομή ΔΟΛ και επίσης είναι σε θέση να αξιολογεί το επίπεδο αλληλεπίδρασης του επιτιθέμενου παράγοντας ένα τελικό σκορ. Το HoneyMix συνδυάζοντας τα παραπάνω καταφέρνει να εξουδετερώσει τεχνικές αναγνώρισης.

Στο [26] οι συγγραφείς περιγράφουν και παρουσιάζουν το HoneyDOC μια αρχιτεκτονική παγίδων εισβολών που είναι σε θέση να συλλέξει δεδομένα υψηλής ποιότητας. Η αρχιτεκτονική του HoneyDOC χωρίζει την λειτουργία του αντιπερισπασμού (decoy) του επιτιθέμενου από την λειτουργία της καταγραφής δεδομένων (captor) και εισάγει ένα σύστημα το οποίο αναλαμβάνει να συντονίσει τα υπόλοιπα δύο συστήματα (orchestrator). Το σύστημα αντιπερισπασμού είναι υπεύθυνο για την εκκίνηση των παγίδων εισβολών στο δίκτυο, το σύστημα καταγραφής είναι υπεύθυνο για την λήψη και καταγραφή των δεδομένων των παγίδων εισβολών και τέλος της ερμηνείας αυτών με τρόπο που να μην γίνει αισθητός στον επιτιθέμενο. Τέλος το σύστημα συντονισμού αναλαμβάνει να εναρμονίσει τα υπόλοιπα δύο συστήματα. Το HoneyDOC στηρίζεται στις λειτουργίες του ΔΟΛ για να πετύχει τους παραπάνω στόχους. Πιο συγκεκριμένα χρησιμοποιεί τις λειτουργίες που προσφέρει ένα ΔΟΛ για την αποτελεσματική και απαρατήρητη μεταφορά πληροφορίας. Επίσης η δρομολόγηση της κίνησης προς μια παγίδα εισβολών γίνεται μέσο του ΔΟΛ. Το HoneyDOC επαληθεύτηκε μέσω ενός πειραματικού σενάριο και σε ένα πραγματικό δίκτυο.

Η έρευνα στο [27] περιγράφουν μια στρατηγική που μπορεί να ακολουθήσει ο επιτιθέμενος ώστε να παρακάμψει την προστασία που προσφέρουν οι παγίδες σε περιβάλλοντα ΔτΠ που χρησιμοποιούν ΔΟΛ. Ως αντίμετρο αυτής οι συγγραφείς επίσης παρουσιάζουν μια στρατηγική που μπορεί να ακολουθήσει ο αμυνόμενος ώστε να αποτρέψει τον επιτιθέμενο από το στόχο του. Το προτεινόμενο μοντέλο καταλήγει σε ένα παίγνιο μεταξύ δύο παικτών και επικυρώνεται

με ένα πειραματικό δίκτυο αποτελούμενο από ένα ελεγκτή και ένα μεταγωγέα ΔΟΛ. Τα αποτελέσματα που καταλήγουν δείχνουν μειωμένη συνολική κατανάλωση ισχύος και μεγαλύτερη συνολική ασφάλεια για το ΔτΠ.

Χρησιμοποιώντας τις δυνατότητες που προσφέρει ένα ΔΟΛ οι συγγραφείς στο [28] προτείνουν ένα σύστημα πέντε συστατικών στοιχείων που είναι σε θέση να παράγει δυναμικά μια διαφορετική τοπολογία δικτύου καθιστώντας ανέφικτη την εύρεση της πραγματικής τοπολογίας του δικτύου. Ο controller διαχειρίζεται την υποδομή ΔΟΛ, ο deception server προσομοιώνει εικονικά συστήματα, ο virtual network view generator που παράγει μια διαφορετική τοπολογία για κάθε συνδεδεμένο σύστημα και τέλος ο honeypot server που είναι υπεύθυνος για την εκκίνηση και διαχείριση των παγίδων εισβολών μέσα στις εικονικές τοπολογίες που δημιουργούνται. Μηνύματα πρωτοκόλλων που ανιχνεύουν συστήματα υπολογιστών εισέρχονται προς το deception server και πλαστές απαντήσεις επιστρέφονται. Τέλος για να υπάρχει πιο ρεαλιστική συμπεριφορά εισάγεται τεχνητή καθυστέρηση σε πλαίσια ώστε να μην μπορεί να γίνει αντιληπτός ένα εικονικό σύστημα από ένα πραγματικό.

Ο Martins και λοιποί στο [29] χρησιμοποιούν το γνωστό σύστημα ανίχνευσης εισβολών snort [30] υλοποίησαν ένα σύστημα που είναι σε θέση να εντοπίσει και να ανακατευθύνει κακόβουλη κίνηση προς παγίδες εισβολών. Πιο αναλυτικά έχει γίνει κατηγοριοποίηση της κακόβουλης κίνησης σε τρεις κατηγορίες μια εκ των οποίων ανακατευθύνεται προς παγίδες εισβολών για περεταίρω ανάλυση των κινήσεων του επιτιθέμενου. Επιπρόσθετα υπάρχουν δύο αρχεία τα οποία κρατούν αποθηκευμένες διευθύνσεις Medium Access Control - MAC. Το πρώτο περιέχει μια λίστα με εκείνες τις διευθύνσεις που έχον εξουσιοδοτηθεί για επικοινωνία ενώ το δεύτερο περιέχει εκείνες τις διευθύνσεις που έχουν διεξάγει κάποια ύποπτη κίνηση.

Συνδυάζοντας διάφορες τεχνικές μηχανικής μάθησης και την τεχνολογία ΔΟΛ στο [31] οι συγγραφείς παρουσιάζουν πιθανά σενάρια εφαρμογών της μηχανικής μάθησης. Μεταξύ αυτών είναι και η ανακατεύθυνση κίνησης σε παγίδες εισβολών. Το προτεινόμενο σύστημα καταναλώνει λιγότερους πόρους σε σχέση με τις συμβατικές μεθόδους ανίχνευσης εισβολών.

Ασχολούμενοι με Κρίσιμες υποδομές οι ερευνητές στο [32] παρουσιάζουν μια εικονική παγίδα εισβολής υψηλής αλληλεπίδρασης. Χρησιμοποιώντας το MiniCPS, μια επέκταση του Mininet προορισμένη για κρίσιμες υποδομές και βιομηχανικά δίκτυα. Το εικονικό δίκτυο του MiniCPS είναι πλήρως ελεγχόμενο από τον ελεγκτή. Αφήνοντας ένα τρωτό σημείο στο δίκτυο οδηγούν τον επιτιθέμενο στην εικονική παγίδα εισβολών όπου οι κινήσεις του καταγράφονται διαρκώς. Η παγίδα αυτή είναι σχεδιασμένη ως ένα πιστό αντίγραφο του πραγματικού δικτύου εξομοιώνοντας πραγματικές συσκευές όπως PLCs και HMIs και εισάγοντας ρεαλιστικές καθυστερήσεις και διακυμάνσεις τιμών.

Οι συγγραφείς στο [33] το σύστημα GT-HWDS. Αποτελούμενο από τρία υποσυστήματα εφαρμόζει το μοντέλο Holt-Winters Digital Signature θεωρίας παιγνίων για την εύρεση επιθέσεων DoS και DDoS. Επτά χαρακτηριστικά παράγονται και τροφοδοτούνται στο μοντέλο αυτό για την κατηγοριοποίηση των επιθέσεων. Στην συνέχεια οι κακόβουλες δικτυακές ροές κατευθύνονται προς παγίδες εισβολών ή απορρίπτονται από το τείχος προστασίας. Οι συγγραφείς συγκρίνουν το προτεινόμενο σύστημα με ένα άλλο δημοφιλές μοντέλο θεωρίας παιγνίων και σύμφωνα με τα αποτελέσματα το GT-HWDS έχει καλύτερα αποτελέσματα.

Στο [34] υλοποιείται ένα σύστημα ανακατεύθυνσης δικτυακής κίνησης που καθιστά εφικτή την ανακατεύθυνση δικτυακών ροών TCP χωρίς να γίνει αισθητό από τον επιτιθέμενο. Η διαδικασία μεταφοράς της δικτυακής ροής προς την παγίδα εισβολών περιγράφεται σε τρία βήματα τα οποία περιλαμβάνουν πρώτα την εγκαθίδρυση σύνδεσης με το πραγματικό σύστημα, έπειτα την μεταφορά της δικτυακής κίνησης προς της παγίδα εισβολών στέλνοντας εκ νέου τα πλαίσια της τριμερής χειραψίας και αλλάζοντας τους αριθμούς ακολουθίας καταλλήλως. Τέλος γίνεται καταχώρηση των εγγραφών OpenFlow για την άμεση επικοινωνία μεταξύ των δυο συστημάτων. Τα αποτελέσματα δείχνουν ότι το σύστημα αυτό είναι σε θέση να εκτελέσει την λειτουργία της ανακατεύθυνσης χωρίς να γίνει αντιληπτό από τον επιτιθέμενο.

4 Το προτεινόμενο πλαίσιο GuardianNet

Στο παρόν κεφάλαιο παρουσιάζεται το προτεινόμενο σύστημα ανακατεύθυνσης εισβολών με όνομα GuardianNet που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας. Παρουσιάζεται η αρχιτεκτονική του συστήματος και ο ρόλος κάθε εργαλείου. Στην συνέχεια παρουσιάζεται σε βήματα την διαδικασία ανακατεύθυνσης δικτυακής κίνησης. Το κεφάλαιο αυτό αποσκοπεί στην παρουσίαση του πλαισίου στο οποίο λειτουργεί το GuardianNet σαν εργαλείο και τις δυνατότητες που προσφέρει.

4.1 Αρχιτεκτονική του Συστήματος GuardianNet

Το GuardianNet είναι ένα πλαίσιο ανακατεύθυνσης δικτυακής κίνησης. Στηριζόμενο στην τεχνολογία ΔΟΛ και στις παγίδες εισβολών στοχεύει στην παραγωγή χρήσιμων δεδομένων για τον διαχειριστή. Αντί να μπλοκάρει εντελώς τον επιτιθέμενο από μια συσκευή τον ανακατευθύνει σε μια παγίδα εισβολών και καταγράφει την μεταξύ του επικοινωνία. Τα αρχεία που συλλέγονται μπορούν αργότερα να αναλυθούν και να εξαχθούν συμπεράσματα όπως κίνητρα που πιθανόν να είχε. Σχεδιαστικά το GuardianNet αποτελείται από τα εξής στοιχεία:

- **Ο διαχειριστής**: ως διαχειριστής μπορεί να είναι κάποιο φυσικό πρόσωπο το οποίο χρησιμοποιεί μια γραφική διεπαφή ή και μια εφαρμογή όπως ένα παραδοσιακό σύστημα ανίχνευσης εισβολών. Ο διαχειριστής βρίσκεται στο επίπεδο διαχείρισης του δικτύου και έχει πρόσβαση σε μια γενική κατάσταση του. Επικοινωνεί με τον ελεγκτή ΔΟΛ μέσω της βόρειας διεπαφής. Στο πλαίσιο της διπλωματικής δεν εξετάζεται πως ο διαχειριστής αποφασίζει ότι μια δικτυακή ροή είναι ύποπτη.

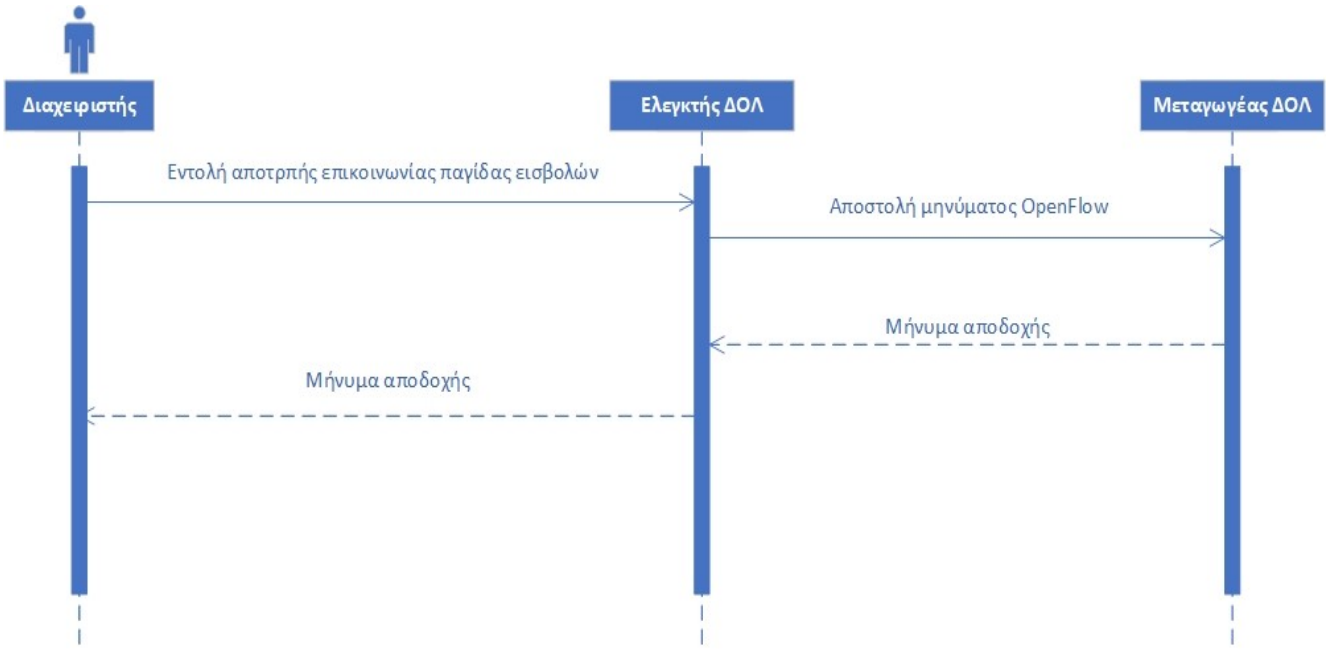
- **Παγίδες εισβολών**: υπάρχουν δύο κατηγορίες παγίδων εισβολών που τοποθετούνται στο δίκτυο.
    - Πρώτη κατηγορία είναι παγίδες εισβολών που εισάγονται στο δίκτυο και σκοπός τους είναι να φαίνονται ως ευάλωτοι στόχοι. Οι παγίδες της κατηγορίας αυτής φαίνονται στο δίκτυο σαν κανονικοί χρήστες. Η παρουσία τους δεν είναι υποχρεωτική για την ομαλή λειτουργία του GuardianNet αλλά συνιστάται να υπάρχουν στο δίκτυο.
    - Δεύτερη κατηγορία είναι παγίδες εισβολών που έχουν ρυθμιστεί καταλλήλως ώστε να προσομοιώνουν επαρκώς την υπηρεσία που προστατεύουν. Οι παγίδες αυτές μένουν κρυμμένες από όλους τους χρήστες και όταν το γίνει αντιληπτή μια ύποπτη κίνηση προς την πραγματική υπηρεσία γίνεται ανακατεύθυνση της κίνησης της προς την αντίστοιχη παγίδα εισβολών. Οι παγίδες αυτής της κατηγορίας έχουν ίδια διεύθυνση MAC και IP με τις αντίστοιχες υπηρεσίες που εξομοιώνουν. Για αποφυγή σύγχυσης οι παγίδες αυτής της κατηγορίας θα αναφέρονται ως πιστό αντίγραφο παγίδα εισβολών (honeypot replica).
- **Υποδομή ΔΟΛ**: η υποδομή ΔΟΛ απαρτίζεται από τον ελεγκτή και από μεταγωγούς ΔΟΛ. Η υποδομή είναι υπεύθυνη για την ανακατεύθυνση της κίνησης από μια πραγματική υπηρεσία προς την αντίστοιχη παγίδα εισβολών. Εντολές που στέλνονται από τον διαχειριστή προωθούνται στην φυσική υποδομή με χρήση της βόρειας διεπαφής.

Σημαντικό στοιχείο είναι η φυσική διασύνδεση των παγίδων εισβολών πάνω στον μεταγωγέα ΔΟΛ. Ο διαχειριστής πρέπει αν έχει επίγνωση της φυσικής διασύνδεσης των υπολογιστών στο δίκτυο. Οι εγγραφές OpenFlow περιέχουν στο πεδίο του κανόνα αντιστοίχισης της διεπαφή που είναι συνδεδεμένο το σύστημα.

4.1 Αρχικοποίηση Εγγραφών Μεταγωγέα ΔΟΛ

Η διαδικασία αρχικοποίησης των εγγραφών γίνεται μετά την εγκαθίδρυση του καναλιού επικοινωνίας μεταξύ μεταγωγέα ΔΟΛ και ελεγκτή. Για κάθε πιστό αντίγραφο παγίδα εισβολών προωθούνται κατάλληλες εγγραφές OpenFlow, όπως φαίνεται στο διάγραμμα ακολουθίας του *Σχήματος 6*, με προτεραιότητα μεγαλύτερη των υπόλοιπων εγγραφών όπου απορρίπτουν μηνύματα προερχόμενα από την παγίδα εισβολών. Εξαίρεση αποτελούν μηνύματα Adress Resolution Protocol - ARP και πολυδύναμής τα οποία ενώ λαμβάνονται από την φυσική διεπαφή τυχόν απαντήσεις αυτών αυτομάτως απορρίπτονται την στιγμή που ληφθούν από τον μεταγωγέα. Με τον τρόπο αυτό οι παγίδες εισβολών παραμένουν κρυφές από το πραγματικό δίκτυο. Η εντολή που περιγράφει την παραπάνω λειτουργικά είναι η εξής:

in_port: honeypot _port → -

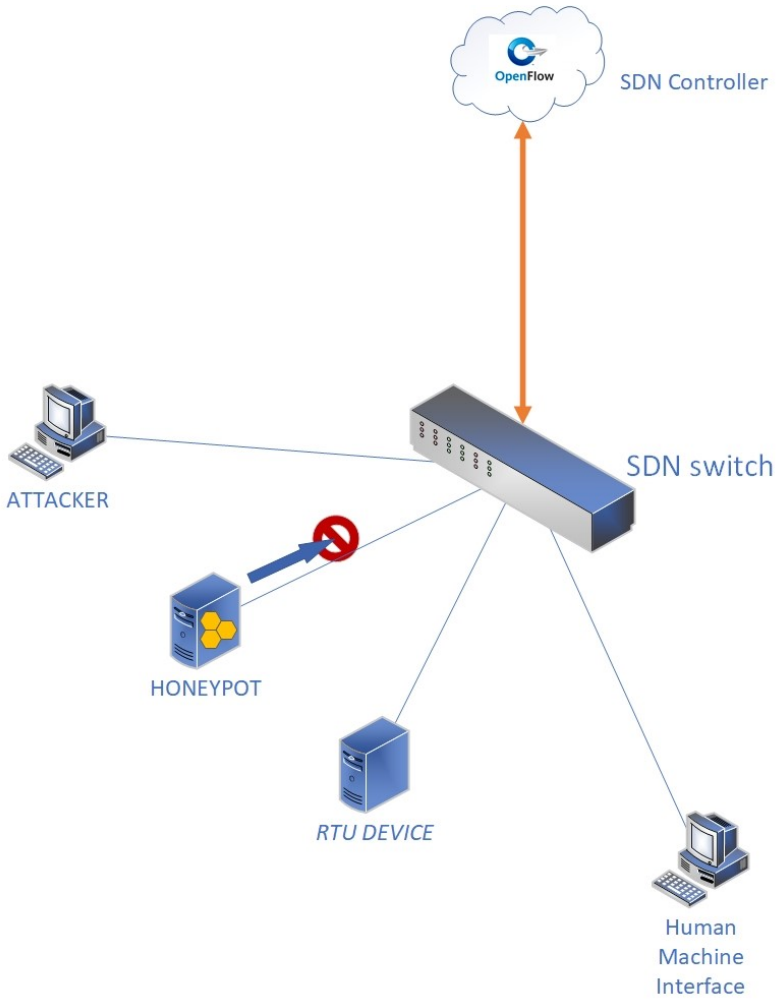


Σχήμα 6 Διάγραμμα ακολουθίας απόρριψης επικοινωνίας Παγίδας Εισβολών

4.2 Φυσιολογική Λειτουργία Δικτύου

Στα παραδοσιακά δίκτυα υπολογιστών η επικοινωνία σε ένα τοπικό δίκτυο γίνεται με την βοήθεια ενός μεταγωγέα επιπέδου δύο. Εσωτερικά ο μεταγωγέας αυτός περιέχει ένα πίνακα που αντιστοιχεί τις διευθύνσεις MAC με τις φυσικές θύρες (ή και εικονικές στην περίπτωση εικονικού μεταγωγέα). Το GuardianNet χρησιμοποιεί ακριβώς την ίδια λογική. Ο ελεγκτής εκτελεί μια διεργασία που αναλαμβάνει να κρατά την πληροφορία αυτή για κάθε μεταγωγέα που είναι συνδεδεμένος. Με τον τρόπο αυτό δεν είναι αναγκαίο να γίνουν αλλαγές στις ρυθμίσεις δικτύου κάθε υπολογιστή που συνδέεται στο ΔΟΛ.

Τα πιστά αντίγραφα παγίδες εισβολών έχοντας τις ίδιες διευθύνσεις MAC και IP με τις υπηρεσίες που μιμούνται θα αποτελούσαν πρόβλημα για τον μεταγωγέα. Δύο υπολογιστές ή περισσότεροι υπολογιστές με ίδια δικτυακά χαρακτηριστικά δεν πρέπει να υπάρχουν στο ίδιο δίκτυο. Ο περιορισμός αυτός πηγάζει από το τρόπο λειτουργίας του επιπέδου δύο. Κάθε πλαίσιο που εισέρχεται στον μεταγωγέα μπορεί να οδηγήσει σε ενημέρωση του πίνακα ARP ώστε μελλοντικά πλαίσια να εξέρθουν από την σωστή θύρα. Για παράδειγμα η αλλαγή της θύρας σύνδεσης ενός υπολογιστή επειδή άλλαξε θέση οδηγεί σε μια τροποποίηση εγγραφής του πίνακα ARP του μεταγωγέα. Το πρόβλημα αυτό λύνεται επίσης με την απαγόρευση επικοινωνίας της παγίδας εισβολών η οποία έχει μεγαλύτερη προτεραιότητα. Από την στιγμή που το πλαίσιο θα απορριφθεί δεν θα γίνει ποτέ η διαδικασία ενημέρωσης του πίνακα. Οπότε συνοψίζοντας η αποτροπή επικοινωνίας πέρα από την απόκρυψη συνεισφέρει και στην σωστή λειτουργία του δικτύου.



Σχήμα 7 Η εξερχόμενη κίνηση απορρίπτεται

4.3 Διαδικασία Ανακατεύθυνσης Κίνησης

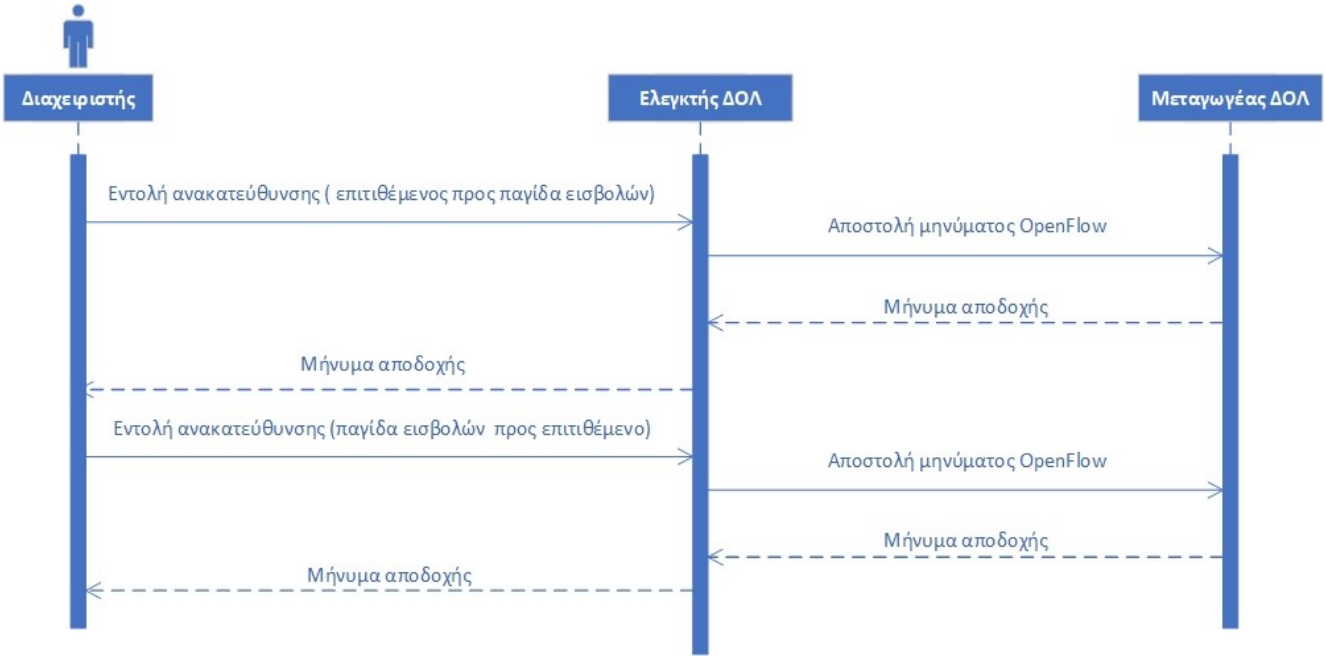
Η διαδικασία ανακατεύθυνσης κίνησης ξεκινά με τον διαχειριστή. Είτε κάποιο φυσικό πρόσωπο ή λογισμικό ο διαχειριστής χρησιμοποιώντας την βόρεια διεπαφή στέλνει τις κατάλληλες εντολές στον ελεγκτή. Πιο συγκεκριμένα ακολουθείτε η λογική του απλού

μεταγωγέα επιπέδου δύο. Απαιτείται γνώση των διευθύνσεων MAC και των φυσικών θυρών που είναι συνδεδεμένοι οι δύο υπολογιστές.

Δύο εγγραφές OpenFlow πρέπει να σταλούν προς τον μεταγωγέα ΔΟΛ ώστε να γίνει εφικτή η επικοινωνία μεταξύ επιτιθέμενου και παγίδας εισβολών όπως αυτές περιγράφονται και σχηματικά από το διάγραμμα ακολουθίας στο *Σχήμα 8*. Οι παρακάτω δύο εντολές σε ψευδοκώδικα περιγράφουν το πεδίο αντιστοίχιση. Η πρώτη αφορά την μονόδρομη δικτυακή ροή από τον επιτιθέμενο προς την παγίδα εισβολών και η δεύτερη την μονόδρομη δικτυακή ροή από την παγίδα εισβολών προς τον επιτιθέμενο. Οι δύο αυτές εντολές μαζί δημιουργούν την αμφίδρομη δικτυακή ροή μεταξύ παραβλέποντας τους κανόνες απόρριψης της παγίδας εισβολών.

$$in\_port: attacker\_port \text{ ^ } source\_mac: mac_{attacker} \text{ ^ } destination\_mac: mac_{service} \rightarrow out\_port: honeypot\_port$$

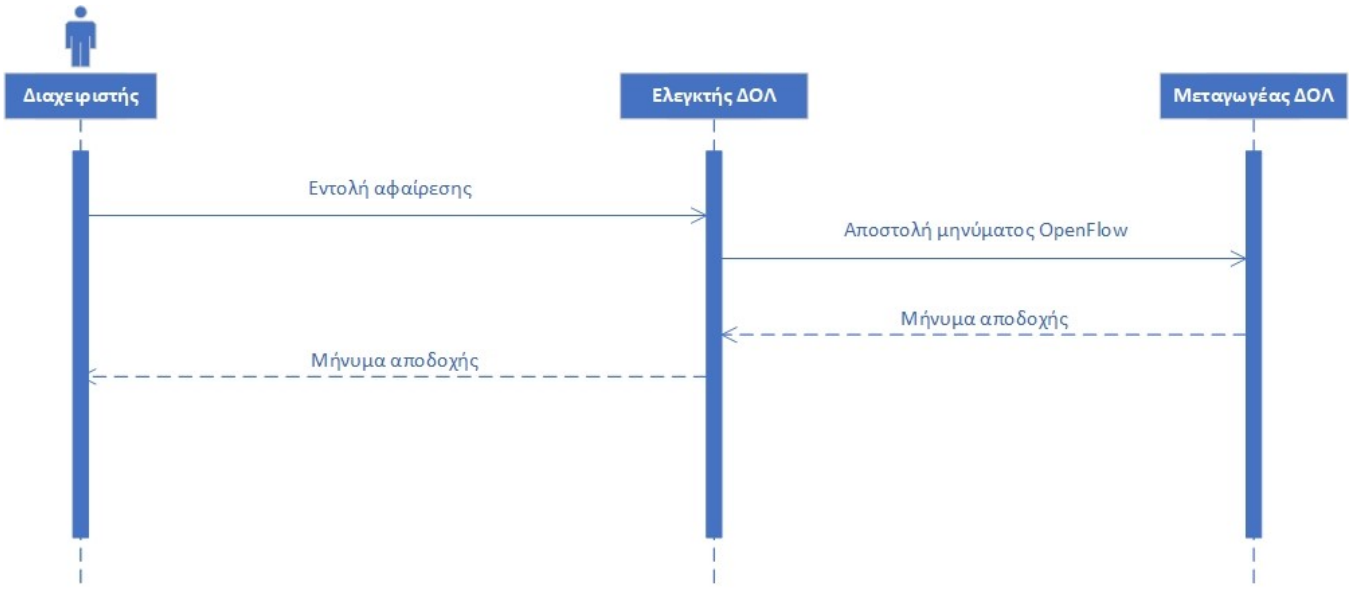
$$in\_port: honeypot\_port \text{ ^ } source\_mac: mac_{service} \text{ ^ } destination\_mac: mac_{attacker} \rightarrow out\_port: attacker\_port$$



Σχήμα 8 Διαδικασία ανακατεύθυνσης κίνησης προς μια παγίδα εισβολών

4.3 Διαδικασία επαναφοράς

Η επαναφορά της επικοινωνίας είναι μια απλή διαδικασία διαγραφής των δύο εντολών που προηγουμένως στάλθηκαν. Για την άμεση διαγραφή των κανόνων πρέπει να σταλούν κατάλληλες εντολές στον ελεγκτή μέσω της βόρειας διεπαφής επικοινωνίας. Η διαγραφή ενός κανόνα OpenFlow γίνεται με βάση τις τιμές που ορίζονται στο μήνυμα διαγραφής και του πεδίου *ενέργειες* της εγγραφής. Ο διαχειριστής πρέπει να έχει αποθηκευμένα τις εντολές που έχει στείλει ώστε να είναι εφικτή η άμεση και ορθή διαγραφή έγγραφής και ακολουθείται η διαδικασία του *Σχήματος 9*.

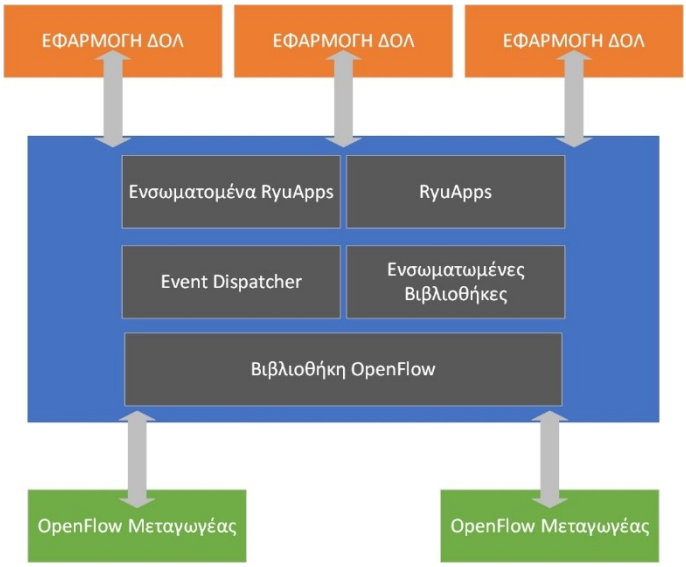


Σχήμα 9 Διαδικασία επαναφοράς

5 Υλοποίηση και Αποτίμηση Απόδοσης

Στο κεφάλαιο αυτό γίνεται μια υλοποίηση του προτεινόμενου συστήματος GuardianNet. Το κεφάλαιο ξεκινά με μια εισαγωγή και παρουσίαση των εργαλείων. Συγκεκριμένα παρουσιάζονται οι δυνατότητες του κάθε εργαλείου και στο τέλος παρουσιάζεται η συνολική υλοποίηση του συστήματος.

5.1 Ryu



Σχήμα 10 Αρχιτεκτονικη του ελεγκτή Ryu

Ο Ryu είναι ένα προγραμματιστικό πλαίσιο για την συγγραφή εφαρμογών ελέγχου ΔΟΛ βασισμένο σε συστατικά στοιχεία. Πρόκειται για ένα κεντρικοποιημένο ελεγκτή υλοποιημένος αποκλειστικά σε γλώσσα προγραμματισμού Python3 ενώ παρέχει υποστήριξη για μεγάλο

πλήθος Southbound πρωτοκόλλων συμπεριλαμβανομένου και του OpenFlow. Το διάγραμμα αρχιτεκτονικής του απεικονίζεται στο *Σχήμα 10.*

Ο έλεγχος του επιπέδου δεδομένων γίνεται με την χρήση RyuApps. Ένα RyuApp στην πράξη είναι ένα αρχείο κώδικα Python το οποίο χρησιμοποιεί κλάσεις και βιβλιοθήκες που παρέχονται από τον Ryu. Ο Ryu είναι σε θέση να εκτελεί πολλαπλά RyuApps και η επικοινωνία μεταξύ είναι γεγονοδηγούμενη. Κάθε RyuApp μπορεί ακούει σε πολλά γεγονότα και τα γεγονότα μπορούν να μεταδίδονται σε πολλά RyuApps. Τελειώνοντας ο Ryu έρχεται μια συλλογή από έτοιμα RyuApps ώστε ο προγραμματιστής να μην χρειάζεται να ξεκινά από το μηδέν κάθε φορά.

Στην παρούσα διπλωματική χρησιμοποιήθηκε η έκδοση 4.33 του ryu και χρησιμοποιήθηκαν τόσο RyuApps που παρέχονται από το Ryu όσο και RyuApps από τρίτους. Παρακάτω ακολουθεί μια λίστα τα RyuApps:

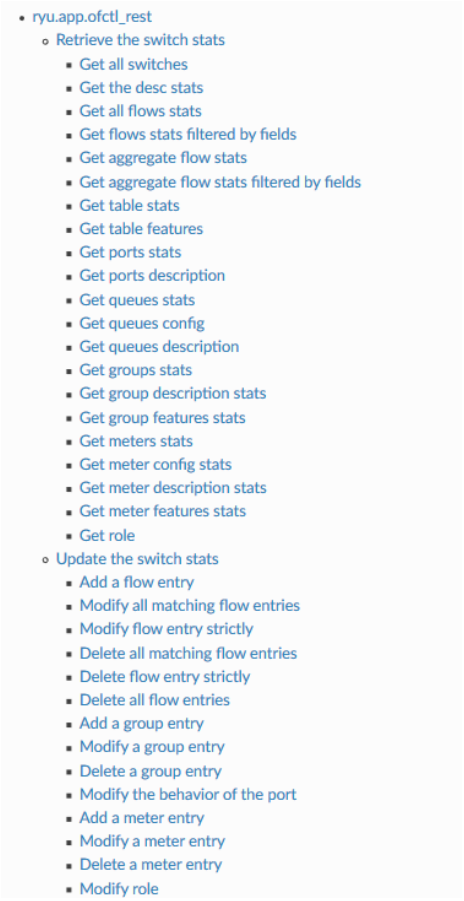
5.1.1 RyuApp: SimpleSwitch13

Το RyuApp αυτό έρχεται μαζί με τον Ryu και υλοποιεί ένα απλό μεταγωγέα επιπέδου δύο. Η επικοινωνία στην νότια διεπαφή χρησιμοποιεί την έκδοση OpenFlow 1.3. Χρησιμοποιώντας τα δεδομένα που περιέχονται από ένα packet in μηνύματα, το RyuApp αυτό είναι σε θέση να κατασκευάσει τον πίνακα ARP. Στην παρακάτω εικόνα φαίνονται οι εγγραφές που παρήγαγε το simple switch σε μια τοπολογία αστέρα με τρεις υπολογιστές.

5.1.2 RyuApp: ofctl_rest

Το RyuApp αυτό συμπεριλαμβάνεται στον Ryu και υλοποιεί μια υπηρεσία REST μέσω της οποίας μπορεί κάποιος να στείλει εντολές προς τον ελεγκτή [35]. Το RyuApp έχει μια εκτενή τεκμηρίωση των εντολών και παραδείγματα αυτό Στο πλαίσιο της διπλωματική χρησιμοποιήθηκαν οι εντολές προσθήκης και αφαίρεσης εγγραφών OpenFlow. Στο *Σχήμα 11*

παρέχεται η λίστα τον υποστηριζόμενων εντολών REST όπως αυτές είναι στον ιστότοπο του Ryu.

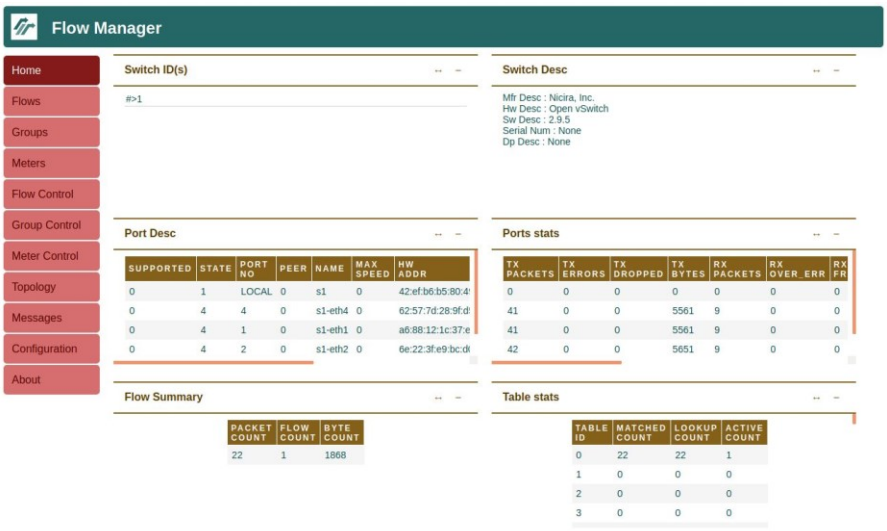


Σχήμα 11 Υποστηριζόμενες λειτουργίες του OFCTL_REST

5.1.3 RyuApp: FlowManager

Το Flowmanager [36] είναι ένα RyuApp που δεν παρέχεται από τον Ryu. Προσφέρει μια γραφική διεπαφή φυλλομετρητή μέσω της οποίας μπορεί κάποιος να παρακολουθεί στατιστικά των μεταγωγέων ΔΟΛ, να βλέπει την τοπολογία του δικτύου και προσθαφαιρεί

εγγραφές OpenFlow και να τροποποιεί υπάρχουσες. Το Flowmanager χρησιμοποιεί και αυτό το ofctl_rest μέχρι ένα σημείο αλλά παρέχει μια δικιά του υπηρεσία REST. Στο πλαίσιο της διπλωματικής όμως δεν χρησιμοποιήθηκε.



Σχήμα 12 Κεντρική σελίδα του Flowmanager

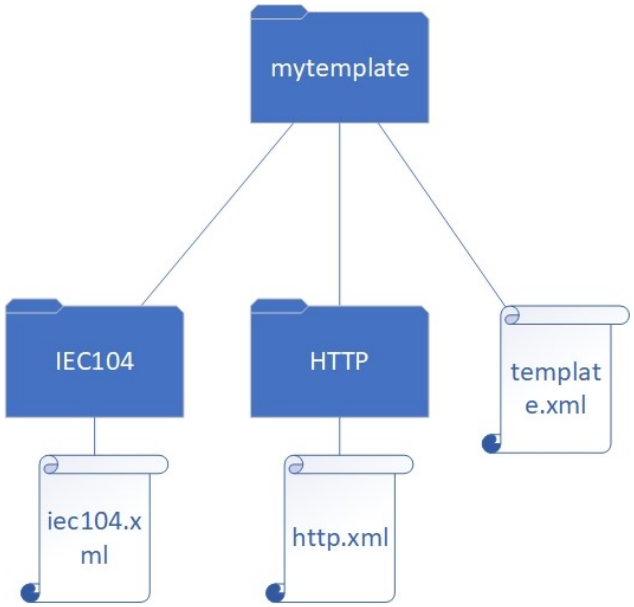
5.2 Conpot

Το Conpot είναι μια παγίδα εισβολών χαμηλής αλληλεπίδρασης υλοποιημένη σε Python3 σχεδιασμένο για Βιομηχανικά δίκτυα [37]. Παρέχει υποστήριξη για ένα μεγάλο πλήθος βιομηχανικών πρωτοκόλλων ενώ έχει σχεδιαστεί με τέτοιο τρόπο ώστε λειτουργίες παραμετροποίησης και επέκτασης είναι εύκολες [38]. Η κατηγοριοποίησή του ως προς τον σκοπό δεν είναι αναφέρεται ξεκάθαρα.

Η παραμετροποίηση του Conpot γίνεται σε δύο στάδια. Ο προγραμματιστής πρώτα ρυθμίζει το Conpot ως πακέτο λογισμικού και έπειτα να τροποποιεί τα πρωτόκολλα που επιθυμεί να εκκινεί το Conpot. Στην πρώτη περίπτωση το αρχείο ρύθμισης είναι ένα απλό αρχείο

κειμένου. Στο αρχείο αυτό προσδιορίζονται μεταβλητές όπως η διαδρομή που θα χρησιμοποιηθεί για την αποθήκευση δεδομένων, χρόνοι αναμονής και ακόμα μέθοδοι αποθήκευσης των αρχείων εγγραφών.

Η παραμετροποίηση των πρωτοκόλλων χρησιμοποιεί το πρότυπο Extensible Markup Language - XML. Πιο συγκεκριμένα ο προγραμματιστής πρέπει να τροποποιήσει μια σειρά από αρχεία XML ενώ τα αρχεία αυτά πρέπει να έχουν συγκεκριμένα ονόματα. Πρώτο αρχείο είναι το *template.xml* το οποίο χρησιμοποιείται από το Conpot για την δημιουργία μερικών καθολικών μεταβλητών, μεταβλητών που χρησιμοποιούνται από πολλά υποσυστήματα του Conpot. Επίσης στο αρχείο αυτό περιγράφονται δομές δεδομένων που χρησιμοποιεί το κάθε πρωτόκολλο. Στην συνέχεια ακολουθούν τα αρχεία XML που είναι ειδικά για το κάθε πρωτόκολλο. Τα αρχεία αυτά έχουν διαφορετική δομή. Για την καλύτερη οργάνωση τα αρχεία παραμετροποίησης των πρωτοκόλλων είναι σε φακέλους προφίλ (templates).Στο Σχήμα 13 παρουσιάζεται η δομή ενός προφίλ παραμετροποίησης. Κάθε φάκελος προφίλ περιέχει ένα αρχείο template.xml και τα αρχεία των πρωτοκόλλων που επιθυμεί ο προγραμματιστής.



Σχήμα 13 Δομή αρχείων ενός προφίλ του Conpot

5.2.1 Εκκίνηση Conpot

Η εκκίνηση του Conpot γίνεται από την γραμμή εντολών. Ο προγραμματιστής πρέπει να προσδιορίσει το προφίλ που θέλει να χρησιμοποιήσει, το αρχείο cfg.

```
(base) thanasis@UbuntuDev:~$ conpot --template default -f --temp_dir /tmp
WARNING:scapy.runtime:No route found for IPv6 destination :: (no default route?)

                     _
  ___ ___ ___ ___ ___| |_
 |  _| . |   | . | . |  _|
 |___|___|_|_|  _|___|_|
             |_|

 Version 0.6.0
 MushMush Foundation

WARNING:root:--force option specified. Using testing configuration
2020-06-30 23:13:08,148 --force option specified. Using testing configuration
2020-06-30 23:13:08,149 Starting Conpot using template: /home/thanasis/anaconda3/lib/python3.7/site-packages/conpot/templates/default
2020-06-30 23:13:08,149 Starting Conpot using configuration found in: /home/thanasis/anaconda3/lib/python3.7/site-packages/conpot/testing.cfg
WARNING:conpot.core.virtual_fs:Using default FS path. tar:///home/thanasis/anaconda3/lib/python3.7/site-packages/conpot/data.tar
2020-06-30 23:13:08,160 Using default FS path. tar:///home/thanasis/anaconda3/lib/python3.7/site-packages/conpot/data.tar
2020-06-30 23:13:08,160 Initializing Virtual File System at /tmp/__conpot__tdlpmi_u. Source specified : tar:///home/thanasis/anaconda3/lib/python3.7/site-packages/conpot/data.tar
 Please wait while the system copies all specified files
```

Σχήμα 14 Εκκίνηση του Conpot

Από την έκδοση 0.6.0 και έπειτα δεν είναι δυνατόν κάποιος να εκκινήσει το Conpot με δικαιώματα υπερχρήστη. Αυτό θέτει ένα πρόβλημα δεδομένου ότι πρωτόκολλα όπως το Modbus και το HTTP χρησιμοποιούν θύρες μικρότερες της 1024 οι οποίες απαιτούν δικαιώματα υπερχρήστη για την δέσμευσή τους. Λύση στο πρόβλημα αυτό είναι οι χρήση κανόνων iptables. Το λογισμικό πακέτο iptables επιτρέπει την συγγραφή κανόνων δικτυακών ροών.

Οι κανόνες στην περίπτωση του Conpot τροποποιούν καταλλήλως πεδία TCP ώστε ένας εξωτερικό χρήστης να έχει την ψευδαίσθηση ότι επικοινωνεί χρησιμοποιώντας την κανονική

θύρα του πρωτοκόλλου ενώ εσωτερικά το Conpot έχει δεσμεύσει θύρα με αριθμό μεγαλύτερο από 1024.

5.3 Open vSwitch

Το Open vSwitch είναι ένας μεταγωγέας λογισμικού πολλαπλών επιπέδων. Στην πράξη σημαίνει ότι είναι σε θέση να επεξεργαστεί κίνηση σε επίπεδο ζεύξης δεδομένων και σε επίπεδο δικτύου εκτελώντας δηλαδή και λειτουργίες ενός δρομολογητή. Έχει υλοποιηθεί σε γλώσσα προγραμματισμού C και το πεδίο εφαρμογών του είναι κυρίως σε περιβάλλοντα εικονικών μηχανών στην λειτουργία δρομολόγησης κίνησης. Υποστηρίζει όλες τις λειτουργίες ενός πραγματικού μεταγωγέα χωρίς την ανάγκη για ειδικό εξοπλισμό. Επιπρόσθετα παρέχει υποστήριξη για το πρωτόκολλο OpenFlow.

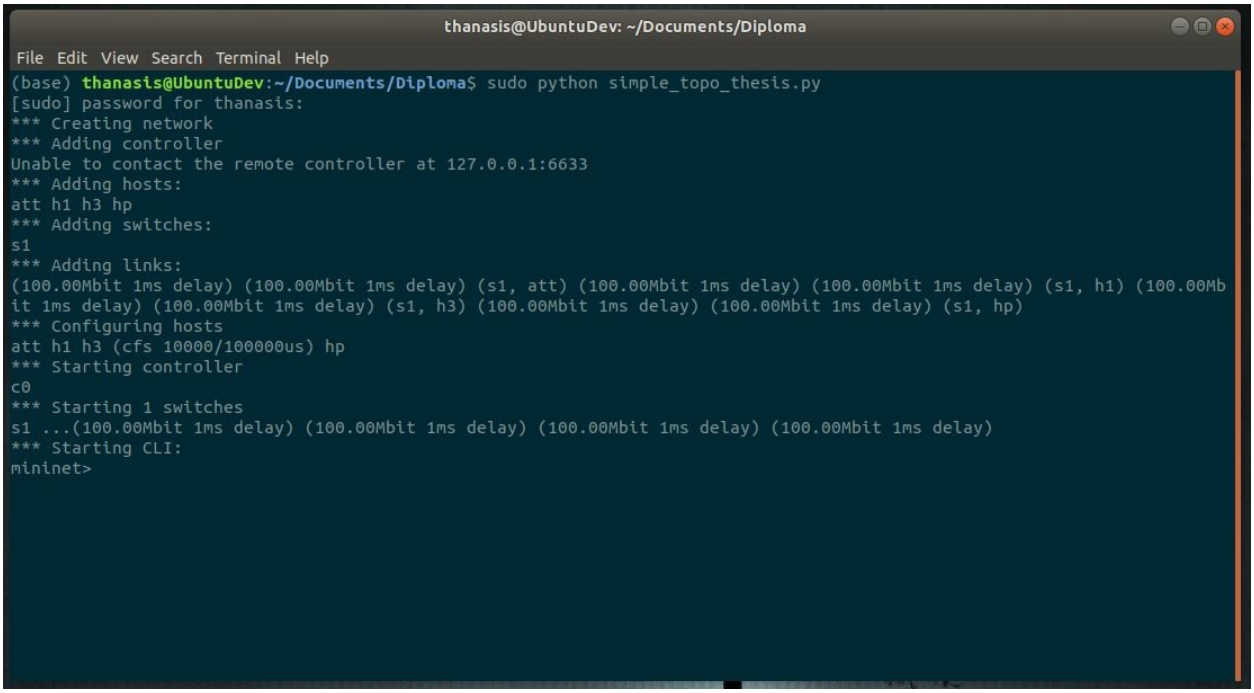
Εγκαθιστώντας κάποιος το Open vSwitch εγκαθιστά μερικές διεργασίες που συνεργατικά μεταξύ του προσφέρουν όλη την λειτουργικότητα. Οι πιο κύριες διεργασίες είναι οι εξής:

- **ovs-vswitchd**: είναι η κύρια διεργασία που υλοποιεί τον εικονικό μεταγωγέα. Οι λειτουργίες της δρομολόγησης πλαισίων προς τον σωστό παραλήπτη εκτελούνται από αυτή την διεργασία.
- **ovsdb-server**: είναι η βάση δεδομένων που αποθηκεύονται δεδομένα.
- **ovs-vsctl**: εργαλείο τερματικού που παρέχει την δυνατότητα επικοινωνίας με το ovs-vswitchd.
- **ovs-ofctl**: εργαλείο τερματικού που παρέχει την υποστήριξη για τον πρωτόκολλο OpenFlow.

5.4 Mininet

Το Mininet [39] είναι ένας εξομοιωτής δικτύων που δημιουργεί εικονικές τοπολογίες ΔΟΛ. Έχει υλοποιηθεί στο μεγαλύτερο μέρος του σε γλώσσα προγραμματισμού Python2 και προσφέρει μια απλοϊκή και εύκολα επεκτάσιμη [40] διεπαφή προγραμματισμού επίσης σε γλώσσα Python2. Στο παρασκήνιο χρησιμοποιεί το Open vSwitch για την δημιουργία των εικονικών μεταγωγών. Ο προγραμματιστής δεν εισάγει εντολές μέσο του ovs-vsctl αλλά γράφοντας σενάρια σε python και χρησιμοποιώντας τις βιβλιοθήκες που παρέχει το Mininet στήνει την τοπολογία που επιθυμεί. Καθώς το Mininet στηρίζεται στο Open vSwitch για την δημιουργία του δικτύου συνιστάται η χρήση λειτουργικών συστημάτων της οικογένειας Unix/Linux.

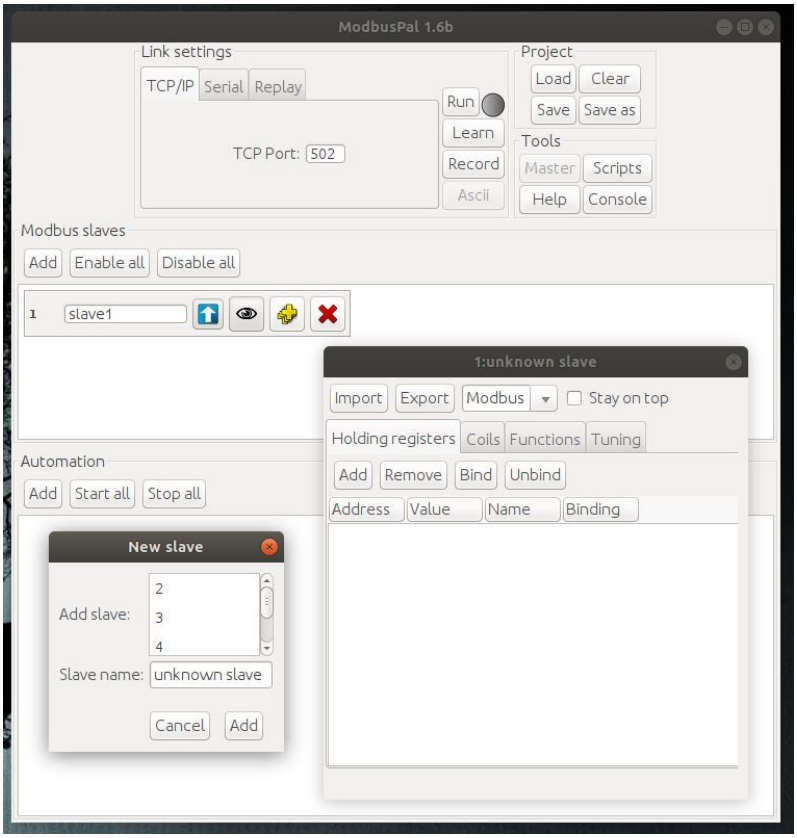
Ένα μεγάλο πλεονέκτημα του Mininet είναι ότι χρησιμοποιεί τεχνικές εικονοποίησης που παρέχονται από το λειτουργικό σύστημα. Οι εικονικοί χρήστες στην πραγματικότητα είναι διεργασίες που εκτελούνται από το υποκείμενο λειτουργικό σύστημα και είναι σε θέση να εκτελέσουν οποιοδήποτε κομμάτι κώδικα. Ο προγραμματιστεί μπορεί να ορίσει και επιπρόσθετα χαρακτηριστικά όπως ρυθμαπόδοση της διασύνδεσης ακόμα και ποσοστό επεξεργαστικής δύναμης που μπορεί να έχει ο κάθε εικονικός χρήστης.



Σχήμα 15 Εκκίνηση τοπολογίας Mininet

5.5 ModbusPal

Το ModbusPal [41] είναι ένας προσομοιωτής για το πρωτόκολλο Modbus υλοποιημένος σε γλώσσα προγραμματισμού Java.Είναι σε θέση να προσομοιώσει την λειτουργία που προσφέρει ένα RTU. Επιτρέπει στον προγραμματιστή να ορίσει Modbus slaves μέσω του γραφικού περιβάλλοντος και να προσθέσει καταχωρητές σύμφωνους με το πρωτόκολλο Modbus. Παρέχει επίσης την δυνατότητα στον προγραμματιστή να ορίσει σε ποια function codes μπορεί να απαντά, επιτρέπει την εισαγωγή χρονική καθυστέρησης και τέλος υποστηρίζει την εισαγωγή ποσοστού λαθών. Η πιο πρόσφατη έκδοση είναι η 1.6b η οποία και χρησιμοποιήθηκε στα



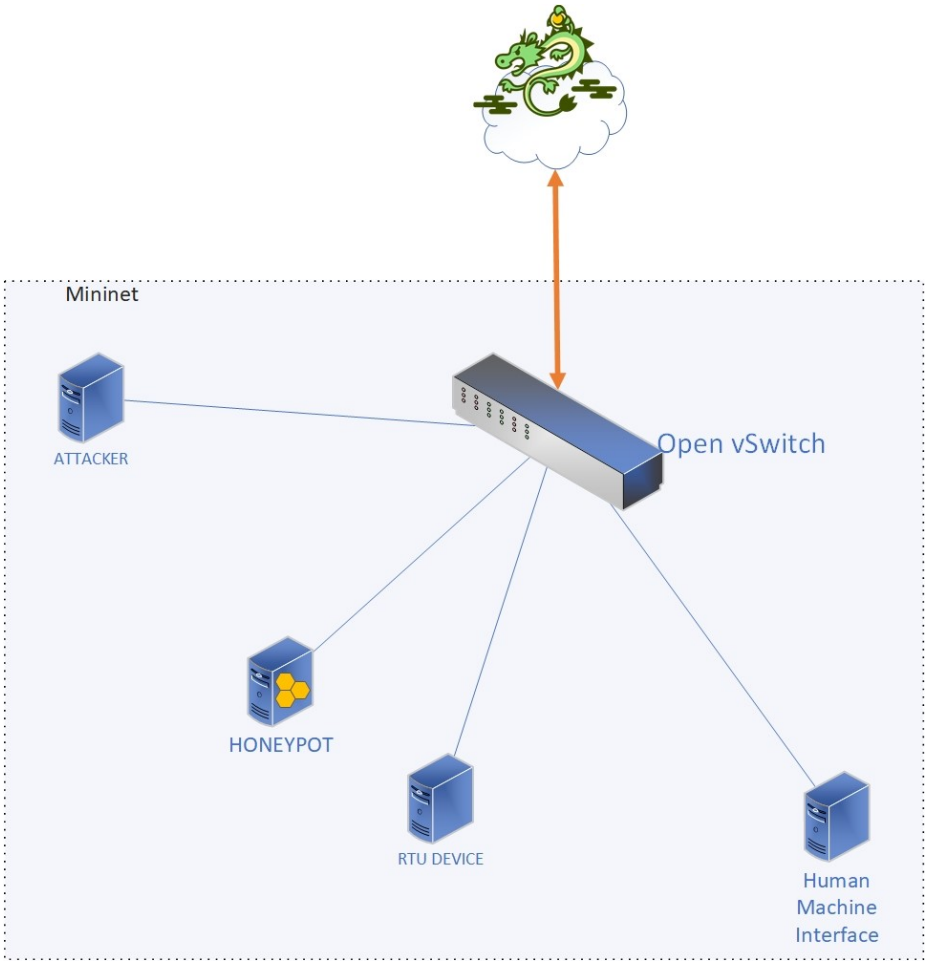
Σχήμα 16 Κεντρικό παράθυρο του ModbusPal

πλαίσιο της διπλωματικής ενώ στο *Σχήμα 16 εικονίζεται το κεντρικό μενού ρυθμίσεων*.

5.6 Υλοποίηση σε περιβάλλον Mininet

Η λειτουργικότητα του GuardianNet επικυρώθηκε μέσω μιας προσομοίωσης. Η υλοποίηση του συστήματος έγινε σε Λειτουργικό Σύστημα Ubuntu 18.04 με έκδοση kernel *5.3.0-59-generic*. Έγινε εγκατάσταση όλων των προαναφερθέντων πακέτων λογισμικού.

Πιο συγκεκριμένα χρησιμοποιήθηκε ο ελεγκτής ryu με τα RyuApps SimpleSwitch13, ofctl_rest και Flowmanager. Το πακέτο λογισμικού ModbusPal χρησιμοποιήθηκε για το ρόλο του RTU και το Conpot ως πιστό αντίγραφο παγίδα εισβολής. Ειδικά για το Conpot ενεργοποιήθηκε μόνο το πρωτόκολλο Modbus ενώ και στα δύο πακέτα λογισμικού οι ρυθμίσεις έγιναν έτσι ώστε να φαίνονται ως η ίδια συσκευή σε ένα τρίτο. Τέλος χρησιμοποιήθηκε ο εξομοιωτής Mininet σε συνδυασμό με τον εικονικό μεταγωγέα Open vSwitch για τη δημιουργία των δικτυακών συσκευών. Η τοπολογία περιγράφεται στο *Σχήμα 17* ενώ στον *Πίνακα 2* περιέχονται οι διευθύνσεις MAC, IP των συστημάτων και του αριθμού εικονικής θύρας που είναι συνδεδεμένο το κάθε σύστημα.

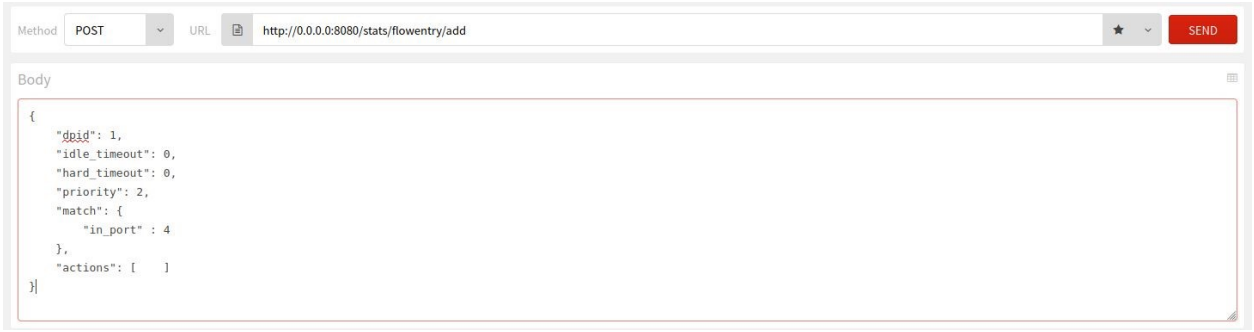


Σχήμα 17 Διάγραμμα δικτύου υλοποίησης προσομοίωσης

Πίνακας 2 Δικτυακά χαρακτηριστικά των συστημάτων

Όνομα	Διεύθυνση MAC	Διεύθυνση IP	Αριθμός θύρας
Human Machine Interface	00:00:00:00:00:01	192.168.10.1	1
Attacker	00:00:00:00:00:02	192.168.10.2	2
RTU Device	00:00:00:00:00:03	192.168.10.3	3
Honeypot	00:00:00:00:00:03	192.168.10.3	4

Αρχικά γίνεται αποστολή της εντολής αποτροπής επικοινωνίας του υπόλοιπου δικτύου με την παγίδα εισβολών από τον διαχειριστή χρησιμοποιώντας την υπηρεσία REST του ofctl_rest RyuApp. Η εντολή εικονίζεται στο *Σχήμα* 18. Για να επιτευχθεί αυτό απαιτείται η θύρα που είναι συνδεδεμένη η παγίδα εισβολών. Η εικόνα παρακάτω είναι από το λογισμικό POSTMAN [42] και περιέχει την εντολή σε μορφή JSON.



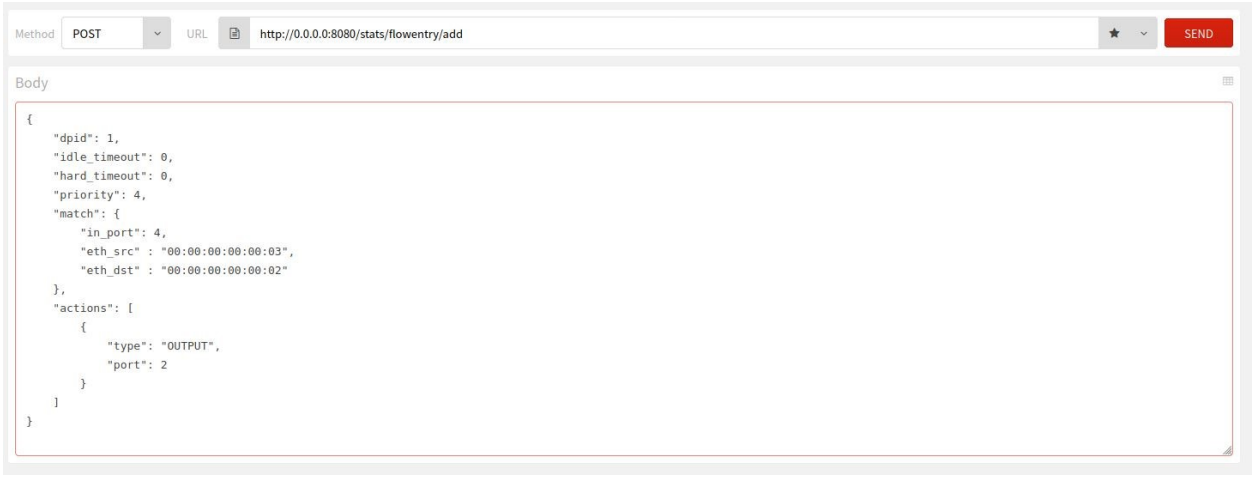
Σχήμα 18 Εντολή REST απόκρυψης παγίδας εισβολής

Από το σημείο αυτό και έπειτα είναι δυνατή η επικοινωνία μεταξύ των συστημάτων. Τα συστήματα επικοινωνούν μεταξύ τους ανταλλάσσοντας δεδομένα χωρίς να έχουν κάποια επίγνωση της ύπαρξης της παγίδας εισβολών. Ο *πίνακας 3* παρουσιάζει τις εφικτές αλληλεπιδράσεις που μπορούν να έχουν τα συστήματα μεταξύ τους. Όπως είναι εμφανές το Honeypot είναι αποκομμένο από το υπόλοιπο δίκτυο.

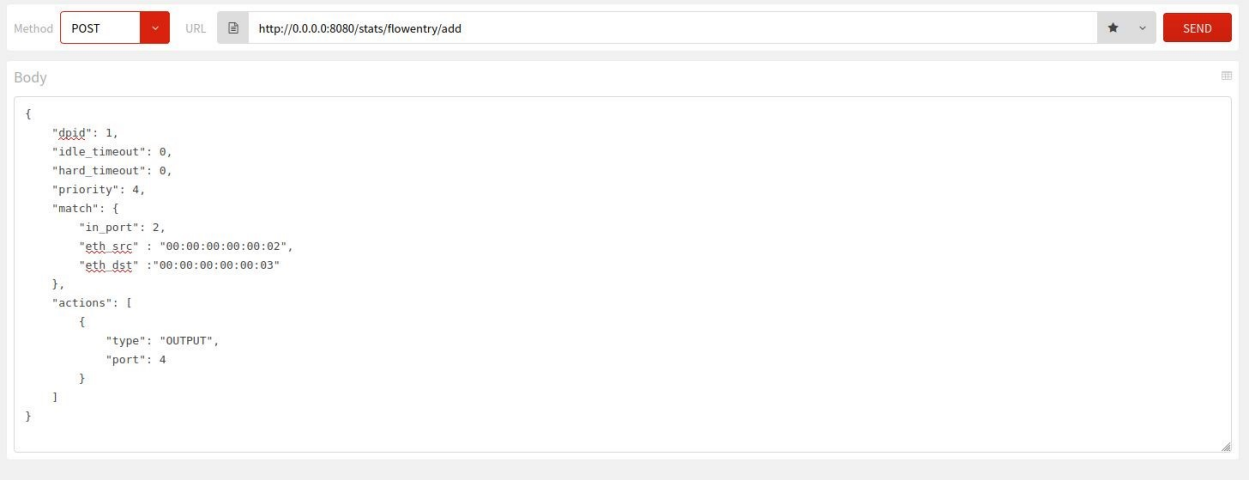
Πίνακας 3 Εφικτές αλληλεπιδράσεις μεταξύ των συστημάτων

	HMI	Attacker	RTU	Honeypot
HMI	x	x	x	
Attacker	x	x	x	
RTU	x	x	x	
Honeypot				x

Μετά από ένα χρονικό διάστημα ο επιτιθέμενος εκτελεί μια κακόβουλη ενέργεια, όπως μια επίθεση άρνησης υπηρεσιών (Denial of Service – DoS), προς το RTU device και ο διαχειριστής μετά από σύντομο χρονικό διάστημα αποφασίζει να εκκινήσει την διαδικασία ανακατεύθυνσης. Γνωρίζοντας τις διευθύνσεις MAC των δύο εμπλεκόμενων χρηστών (παγίδας εισβολών και επιτιθέμενου) στέλνονται οι δύο εντολές ώστε να γίνει δυνατή η επικοινωνία μεταξύ των δύο συστημάτων όπως αυτές απεικονίζονται στις *Σχήματα 19 και 20* Από το σημείο αυτό και έπειτα η επικοινωνία μεταξύ επιτιθέμενου και παγίδας εισβολών έχει γίνει με επιτυχία ενώ η επικοινωνία με το φυσικό RTU δεν είναι δυνατή. Οι εφικτές αλληλεπιδράσεις απεικονίζονται στον *πίνακα 5*. Επίσης στο *Σχήμα 21* φαίνονται οι εντολές OpenFlow έχει ο πίνακας εγγραφών μηδέν.



Σχήμα 19 Εντολή ανακατεύθυνσης κίνησης παγίδας εισβολών προς επιτιθέμενο

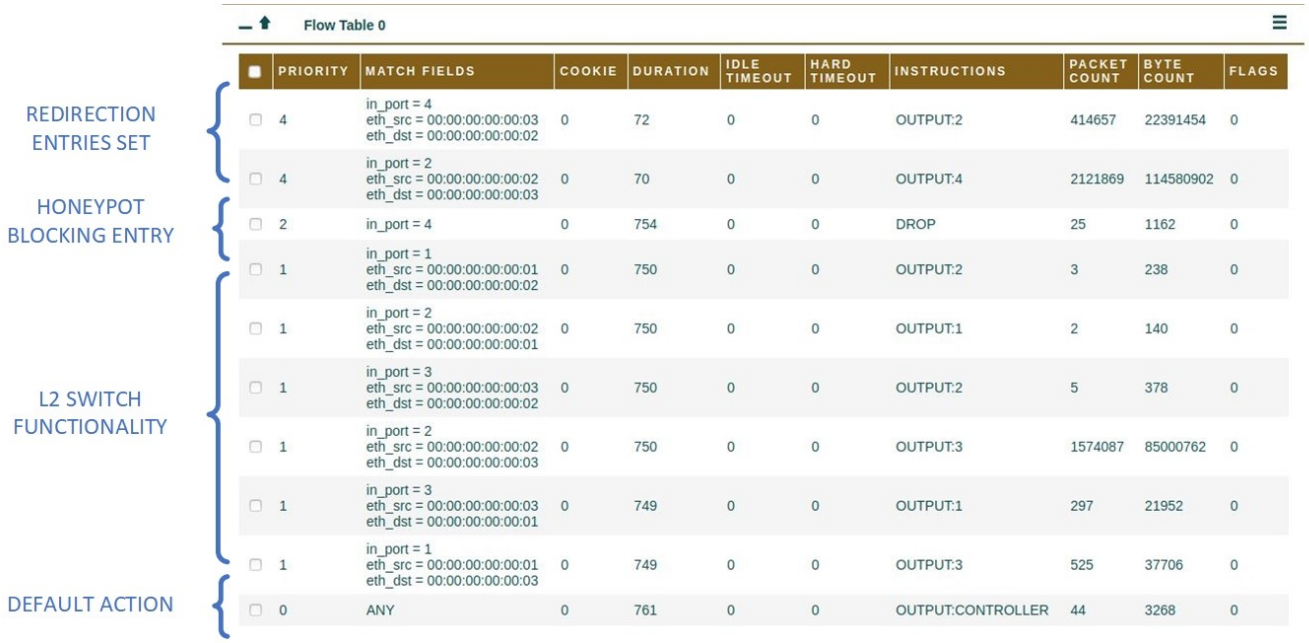


Σχήμα 20 Εντολή ανακατεύθυνσης επιτιθέμενου προς παγίδα εισβολών

Πίνακας 4 Εφικτές αλληλεπιδράσεις μεταξύ των συστημάτων μετά την ανακατεύθυνσης

	HMI	Attacker	RTU	Honeypot
HMI	x	x	x	
Attacker	x	x		x
RTU	x		x	
Honeypot		x		x

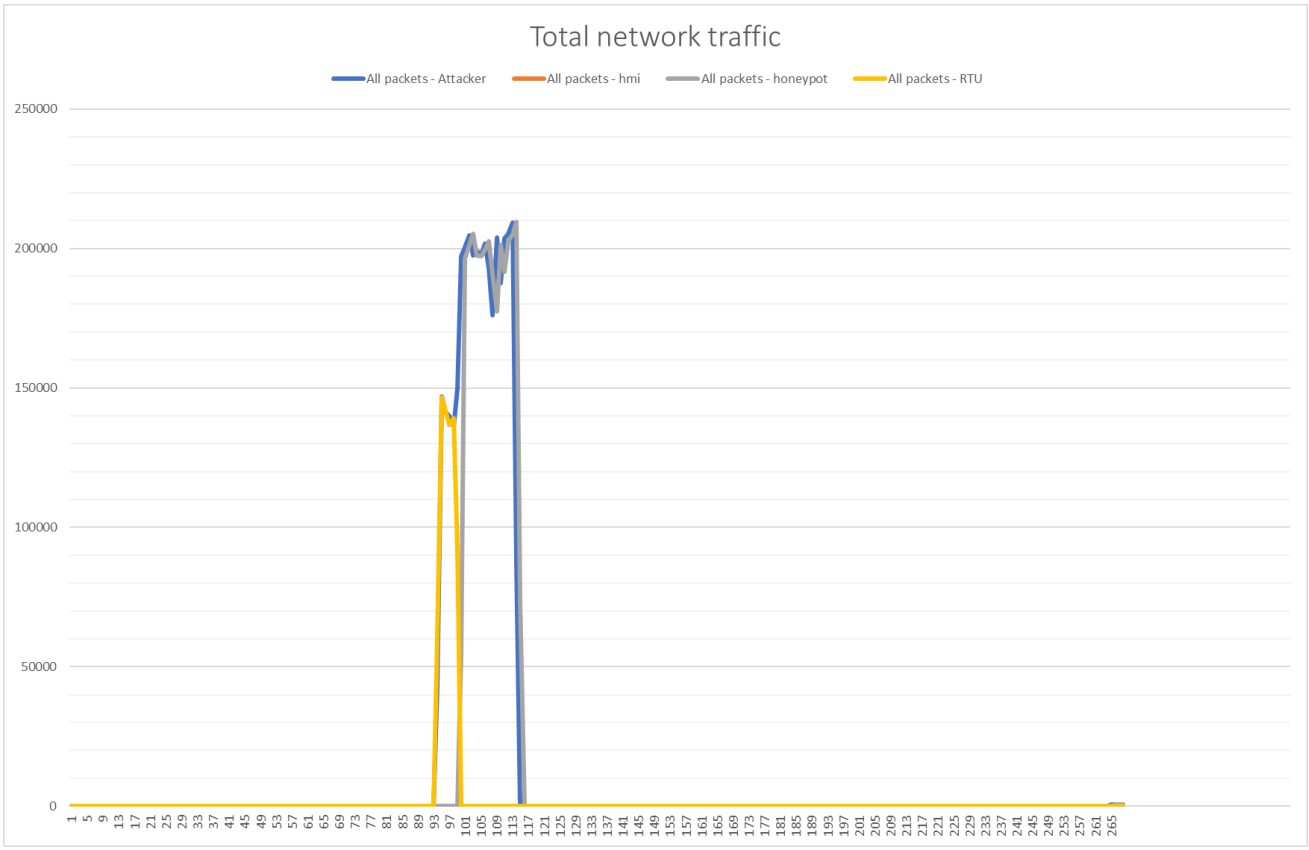
Κατά την διάρκεια της προσομοίωσης γινόταν καταγραφή της δικτυακής κίνησης σε κάθε σύστημα. Το *Σχήμα 22* προβάλει μια συνολική εικόνας της κίνησης στο δίκτυο. Ένα μικρό πλήθος πλαισίων αρχικά καταφέρνουν να φθάσουν στο πραγματικό RTU. Την χρονική στιγμή 100 όμως στην οποία και γίνεται η ανακατεύθυνση, η κίνηση αυτή μεταφέρεται προς την παγίδα εισβολών.



Flow Table 0

PRIORITY	MATCH FIELDS	COOKIE	DURATION	IDLE TIMEOUT	HARD TIMEOUT	INSTRUCTIONS	PACKET COUNT	BYTE COUNT	FLAGS
4	in_port = 4 eth_src = 00:00:00:00:00:03 eth_dst = 00:00:00:00:00:02	0	72	0	0	OUTPUT:2	414657	22391454	0
4	in_port = 2 eth_src = 00:00:00:00:00:02 eth_dst = 00:00:00:00:00:03	0	70	0	0	OUTPUT:4	2121869	114580902	0
2	in_port = 4	0	754	0	0	DROP	25	1162	0
1	in_port = 1 eth_src = 00:00:00:00:00:01 eth_dst = 00:00:00:00:00:02	0	750	0	0	OUTPUT:2	3	238	0
1	in_port = 2 eth_src = 00:00:00:00:00:02 eth_dst = 00:00:00:00:00:01	0	750	0	0	OUTPUT:1	2	140	0
1	in_port = 3 eth_src = 00:00:00:00:00:03 eth_dst = 00:00:00:00:00:02	0	750	0	0	OUTPUT:2	5	378	0
1	in_port = 2 eth_src = 00:00:00:00:00:02 eth_dst = 00:00:00:00:00:03	0	750	0	0	OUTPUT:3	1574087	85000762	0
1	in_port = 3 eth_src = 00:00:00:00:00:03 eth_dst = 00:00:00:00:00:01	0	749	0	0	OUTPUT:1	297	21952	0
1	in_port = 1 eth_src = 00:00:00:00:00:01 eth_dst = 00:00:00:00:00:03	0	749	0	0	OUTPUT:3	525	37706	0
0	ANY	0	761	0	0	OUTPUT:CONTROLLER	44	3268	0

Σχήμα 21Εγγραφές OpenFlow



Σχήμα 22 Πλήθος πλαισίων στο δίκτυο

6 Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Η τελευταία δεκαετία μπορεί να χαρακτηριστεί από ένα νέο κύμα αλλαγών στον τομέα των δικτύων. Τα δίκτυα οριζόμενα από το λογισμικό είναι μια νέα αρχιτεκτονική πολλά υποσχόμενη. Προσφέρει ευελιξία και υπόσχεται να λύσει πολλά από προβλήματα και τους περιορισμούς των παραδοσιακών δικτύων που πρωτοεμφανίστηκαν το 1980. Από την άλλη οι κυβερνοεπιθέσεις έχουν γίνει πιο σύνθετες καθιστώντας πολλές συμβατικές μεθόδους προστασίας

αναποτελεσματικές. Στον τομέα των βιομηχανικών συστημάτων ειδικά αποτελεί ένα μεγάλο ρίσκο. Τέλος οι παγίδες εισβολών έχουν αποδειχθεί ως μια χρήσιμη πηγή γνώσης για τις προθέσεις και τις κινήσεις χρηστών που αλληλοεπιδρούν μαζί τους. Καταγράφοντας οποιαδήποτε αλληλεπίδραση που έχουν με κάποιο τρίτο προσφέρουν στους ειδικούς τα απαραίτητα δεδομένα ώστε να παραχθούν τα μελλοντικά συστήματα και μέθοδοι καταπολέμησης επιθέσεων ενάντια σε υπολογιστικές υποδομές.

Στην διάρκεια της διπλωματικής εργασίας μελετήθηκαν προτεινόμενες αρχιτεκτονικές όπου συνδύαζαν την τεχνολογία των δικτύων οριζόμενων από λογισμικό με τις δυνατότητες που προσφέρουν οι παγίδες εισβολών. Το σύστημα που αναπτύχθηκε είναι σε θέση να ανακατευθύνει ένα επιτιθέμενο σε μια παγίδα εισβολών ειδικά ρυθμισμένη ώστε να μιμείται την πραγματική συσκευή ή υπηρεσία που στόχευσε ο επιτιθέμενος κάνοντας όσο το δυνατόν λιγότερο αισθητό ότι έγινε αυτή η μεταπήδηση. Το προτεινόμενο σύστημα ονόματι GuardianNet καταφέρνει να προστατεύσει μια πραγματική συσκευή, όπως ένα RTU ή ένα κεντρικό διακομιστή ενώ παράλληλα δίνει την εντύπωση στον επιτιθέμενο ότι ακόμα μιλά με τον αρχικό στόχο του.

Ο ελεγκτής αν και ως ένα απλό μέσο επικοινωνίας με την υφιστάμενη υποδομή μπορεί να αποτελέσει ένα σημαντικό εργαλείο εάν συνδυαστεί με κατάλληλα εργαλεία. Το GuardianNet είναι σε θέση να δεχθεί οποιοδήποτε έξυπνο σύστημα αποφάσεων ή και περισσότερα του ενός. Η διεπαφή REST που προσφέρει αν και απλή δίνει ευελιξία στον διαχειριστή όσον αφορά τις πολιτικές που μπορεί να εφαρμόσει.

6.2 Μελλοντικές Επεκτάσεις

Η διπλωματική εργασία αυτή μπορεί να χρησιμοποιηθεί ως εναρκτήριο σημείο για περεταίρω μελέτη των δυνατοτήτων που μπορεί να προσφέρει ο συνδυασμός των δύο αυτών

τεχνολογιών. Ενδιαφερόμενοι θα μπορούσαν να επεκταθούν, χωρίς να περιορίζονται μόνο σε αυτές τις κατευθύνσεις ή να περιορίζονται σε κρίσιμες υποδομές, ως εξής:

- υποστήριξη δυναμικής ανακατεύθυνσης υπαρχόντων συνεδριών TCP από την πραγματική συσκευή προς την παγίδα εισβολής. Η υπάρχουσα στοίβα πρωτοκόλλων δεν υποστηρίζει μια τέτοια λειτουργία δυναμικής μεταφοράς. Η παγίδα εισβολών πρέπει να είναι σε θέση να δέχεται και να απαντάει σωστά σε οποιοδήποτε πλαίσιο και αν λάβει. Οι υπάρχουσες υλοποιήσεις αν και λύνουν μερικώς το πρόβλημα δεν επαρκούν καθώς λαμβάνουν υπόψιν τους μόνο το κομμάτι της ενημέρωση κεφαλίδων και όχι το περιεχόμενο.
- Απομόνωση σημείων του δικτύου που έχουν προσβληθεί από κάποιο κακόβουλο λογισμικό με σκοπό τον περιορισμό της εξάπλωσης του. Κακόβουλα λογισμικά όπως αυτοαναπαραγόμενα κακόβουλα λογισμικά (worms) εξαπλώνονται μέσω του δικτύου και μολύνουν περισσότερα συστήματα. Οι παγίδες εισβολών σε συνδυασμό με την τεχνολογία ΔΟΛ μπορούν να απομονώσουν και συγχρόνως να μελετήσουν την συμπεριφορά των λογισμικών αυτών. Επίσης από τις εγγραφές των παγίδων εισβολών είναι εφικτό να βρεθεί το αρχικό σύστημα από το οποίο μολύνθηκε το δίκτυο.
- Συνδυασμός των στατιστικών δεδομένων που προσφέρει το υλικό με τα αρχεία καταγραφής των παγίδων εισβολών για εντόπιση ανωμαλιών σε ένα δίκτυο. Οι μεταγωγείς ΔΟΛ σύμφωνα με το πρότυπο OpenFlow πρέπει να είναι σε θέση να κρατούν μια σειρά από μετρητές. Οι μετρητές αυτοί σε συνδυασμό με την δραστηριότητα που καταγράφουν οι παγίδες εισβολών μπορούν να συμβάλουν στην γρήγορη και αποτελεσματική εύρεση ύποπτης δραστηριότητας στο δίκτυο.

Βιβλιογραφικές Αναφορές

[1] G. Dileep, "A survey on smart grid technologies and applications," *Renew. Energy*, vol. 146, pp. 2589–2625, Feb. 2020, doi: 10.1016/j.renene.2019.08.092.

[2] "Modbus," *The Modbus Organization*. http://www.modbus.org/.

[3] "DNP3," *Overview of DNP3 Protocol*. https://www.dnp.org/About/Overview-of-DNP3-Protocol.

[4] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 1, pp. 5–20, 2013, doi: 10.1109/SURV.2012.021312.00034.

[5] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *Proceedings of the 45th annual southeast regional conference on - ACM-SE 45*, Winston-Salem, North Carolina, 2007, p. 321, doi: 10.1145/1233341.1233399.

[6] B. Cheswick, "An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied," p. 11.

[7] R. S. Ramachandruni and P. Poornachandran, "Detecting the network attack vectors on SCADA systems," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, Aug. 2015, pp. 707–712, doi: 10.1109/ICACCI.2015.7275694.

[8] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis, "Detecting Targeted Attacks Using Shadow Honeypots," p. 15.

[9] O. Hayatle, A. Youssef, and H. Otrok, "Dempster-Shafer Evidence Combining for (Anti)-Honeypot Technologies," *Inf. Secur. J. Glob. Perspect.*, vol. 21, no. 6, pp. 306–316, Jan. 2012, doi: 10.1080/19393555.2012.738375.

[10] D. K. Rahmatullah, S. M. Nasution, and F. Azmi, "Implementation of low interaction web server honeypot using cubieboard," in *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, Bandung, Indonesia, Sep. 2016, pp. 127–131, doi: 10.1109/ICCEREC.2016.7814970.

[11] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking Control of the Enterprise," p. 12.

[12] H. Kim, T. Benson, A. Akella, and N. Feamster, "The evolution of network configuration: a tale of two campuses," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*, Berlin, Germany, 2011, p. 499, doi: 10.1145/2068816.2068863.

[13] "Software-Defined Networking: The New Norm for Networks." ONF, Winter 2013.

[14] "Ryu SDN Framework." https://ryu-sdn.org/ (accessed Jun. 29, 2020).

[15] "Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions." https://www.opennetworking.org/onos/ (accessed Jun. 29, 2020).

[16] "Home - OpenDaylight." https://www.opendaylight.org/ (accessed Jun. 29, 2020).

[17] "Project Floodlight." https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview (accessed Jun. 29, 2020).

[18] "Open MUL Foundation Home - HOME." http://www.openmul.org/ (accessed Jun. 29, 2020).

[19] "POX Manual Current documentation." https://noxrepo.github.io/pox-doc/html/ (accessed Jun. 29, 2020).

[20] S. Wijeratne, A. Ekanayake, S. Jayaweera, D. Ravishan, and A. Pasqual, "Scalable High Performance SDN Switch Architecture on FPGA for Core Networks," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Seaside CA USA, Feb. 2019, pp. 117–117, doi: 10.1145/3289602.3293933.

[21] W. Li, W. Meng, and L. F. Kwok, "A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures," *J. Netw. Comput. Appl.*, vol. 68, pp. 126–139, Jun. 2016, doi: 10.1016/j.jnca.2016.04.011.

[22] W. Braun and M. Menth, "Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices," *Future Internet*, vol. 6, no. 2, pp. 302–336, May 2014, doi: 10.3390/fi6020302.

[23] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014, pp. 1–8, doi: 10.1109/NOMS.2014.6838228.

[24] R. K. Singh, "Intrusion Detection System Using Advanced Honeypots," vol. 2, no. 1, p. 9, 2009.

[25] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, Mar. 2019, pp. 1602–1606, doi: 10.1109/ITNEC.2019.8729425.

[26] W. Han, Z. Zhao, A. Doupé, and G.-J. Ahn, "HoneyMix: Toward SDN-based Intelligent Honeynet," in *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization - SDN-NFV Security '16*, New Orleans, Louisiana, USA, 2016, pp. 1–6, doi: 10.1145/2876019.2876022.

[27] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernandez, "HoneyDOC: An Efficient Honeypot Architecture Enabling All-Round Design," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 683–697, Mar. 2019, doi: 10.1109/JSAC.2019.2894307.

[28] M. Du and K. Wang, "An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 648–657, Jan. 2020, doi: 10.1109/TII.2019.2917912.

[29] S. Achleitner, T. F. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Deceiving Network Reconnaissance Using SDN-Based Virtual Topologies," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 4, pp. 1098–1112, Dec. 2017, doi: 10.1109/TNSM.2017.2724239.

[30] J. S. B. Martins and M. B. Campos, "A SDN-based Flexible System for On-the-Fly Monitoring and Treatment of Security Events," p. 5.

[31] "Snort - Network Intrusion Detection & Prevention System." https://www.snort.org/ (accessed Jun. 30, 2020).

[32] D. Comaneci and C. Dobre, "Securing Networks Using SDN and Machine Learning," in *2018 IEEE International Conference on Computational Science and Engineering (CSE)*, Bucharest, Oct. 2018, pp. 194–200, doi: 10.1109/CSE.2018.00034.

[33] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards High-Interaction Virtual ICS Honeypots-in-a-Box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy - CPS-SPC '16*, Vienna, Austria, 2016, pp. 13–22, doi: 10.1145/2994487.2994493.

[34] M. V. O. De Assis, A. H. Hamamoto, T. Abrao, and M. L. Proenca, "A Game Theoretical Based System Using Holt-Winters and Genetic Algorithm With Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks," *IEEE Access*, vol. 5, pp. 9485–9496, 2017, doi: 10.1109/ACCESS.2017.2702341.

[35] W. Fan and D. Fernandez, "A novel SDN based stealthy TCP connection handover mechanism for hybrid honeypot systems," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, Bologna, Italy, Jul. 2017, pp. 1–9, doi: 10.1109/NETSOFT.2017.8004194.

[36] "ryu.app.ofctl_rest — Ryu 4.34 documentation." https://ryu.readthedocs.io/en/latest/app/ofctl_rest.html (accessed Jun. 29, 2020).

[37] "GitHub - martimy/flowmanager: An SDN application that gives a network administrator, or a student, the ability to control flows in an OpenFlow network without coding." https://github.com/martimy/flowmanager (accessed Jun. 29, 2020).

[38] "Conpot." http://conpot.org/ (accessed Jun. 29, 2020).

[39] D. Pliatsios, P. Sarigiannidis, T. Liatifis, K. Rompolos, and I. Siniosoglou, "A Novel and Interactive Industrial Control System Honeypot for Critical Smart Grid Infrastructure," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Limassol, Cyprus, Sep. 2019, pp. 1–6, doi: 10.1109/CAMAD.2019.8858431.

[40] "Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet." http://mininet.org/ (accessed Jun. 30, 2020).

[41] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A toolkit for security research on CPS Networks," *ArXiv150704860 Cs*, Jul. 2015, Accessed: Jun. 27, 2020. [Online]. Available: http://arxiv.org/abs/1507.04860.

[42] "ModbusPal - Java MODBUS simulator." http://modbuspal.sourceforge.net/ (accessed Jun. 29, 2020).

[43] "Postman | The Collaboration Platform for API Development." https://www.postman.com/ (accessed Jun. 30, 2020).