ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
ΤΕΙ ΗΠΕΙΡΟΥ

“ΜΕΛΕΤΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΣΦΑΛΕΙΑΣ ΔΙΚΤΥΩΝ VPN ΚΑΙ ΑΝΑΠΤΥΞΗ ΜΕΘΟΔΟΛΟΓΙΩΝ ΓΙΑ ΤΗΝ ΑΝΕΥΡΕΣΗ ΚΑΙ ΑΝΤΙΜΕΤΩΠΙΣΗ ΕΠΙΘΕΣΕΩΝ ”

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΑΠΟ ΤΟΝ ΣΠΟΥΔΑΣΤΗ

ΚΑΒΒΑΔΑ ΒΑΣΙΛΕΙΟ

11662

Email vasilis.kavvadas1992@gmail.com

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ

ΛΙΑΓΚΟΥ ΒΑΣΙΛΙΚΗ

ΑΡΤΑ 2018

“STUDYING AND ANALYSING THE SECURITY OF A VPN NETWORK AND METHODOLOGY DEVELOPMENT FOR TRACKING AND CONFRONTATE ATTACKS”

Εγκρίθηκε απο τριμελή εξεταστική επιτροπή

1. Επιβλέπων καθηγητής

2. Μέλος επιτροπής

3. Μέλος επιτροπής

Ο/Η Προϊστάμενος/η του τμήματος

Ευχαριστίες

Η παρούσα πτυχιακή εργασία αποτελεί το τελευταίο κομμάτι μιας μακροχρόνιας πορείας στον χώρο της εκπαίδευσης. Σε όλα αυτά τα χρόνια γνώρισα πολλούς αξιόλογους συναδέλφους που με βοήθησαν σε όλη αυτή την πορεία . Θα ήθελα να τους ευχαριστήσω όλους για ότι μου προσέφεραν όλα αυτά τα χρόνια και τους εύχομαι και εγώ ότι καλύτερο σε ότι και αν κάνουν. Ειδικότερα, θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγητριά μου ΛΙΑΓΚΟΥ ΒΑΣΙΛΙΚΗ για την υπομονή της και την καθοδηγησή της όλο αυτό το διάστημα της συνεργασίας μας . Εύχομαι τα καλύτερα διότι είναι μια άξια εκπαιδευτικός με πολύ υπομονή και ζήλο πάνω στο αντικειμενό της. Τέλος, ένα μεγάλο ευχαριστώ στην οικογενειά μου που ήταν δίπλα μου όλα αυτά τα χρόνια και με στήριξαν και ειδικότερα στον πατέρα μου που δεν είναι πλέον στην ζωή και δυστυχώς δεν πρόλαβε να με δεί να ολοκληρώνω τις σπουδές μου και να ανοίγω το επόμενο κεφάλαιο στην ζωή μου .

Η παρουσα εργασία ειναι αφιερωμένη στον πατέρα μου

Καββαδά Θεόδωρο

Δήλωση Πνευματικής Ιδιοκτησίας

Η παρούσα πτυχιακή εργασία αποτελεί προϊόν αποκλειστικά δικής μου προσπάθειας. Όλες οι πηγές που χρησιμοποιήθηκαν περιλαμβάνονται στην βιβλιογραφία και γίνεται ρητή αναφορά σε αυτές μέσα στο κείμενο όπου έχουν χρησιμοποιηθεί

Περίληψη

Η πτυχιακή εργασία επιδιώκει να αναπτύξει ένα μοντέλο για την ανίχνευση επιθέσεων σε VPN δίκτυα, μελετώντας τα πρωτόκολλα και την ασφάλεια που παρέχουν. Τα VPN είναι ένας σύγχρονος τρόπος σύνδεσης για ασφαλέστερη σύνδεση στο διαδίκτυο και πρόσβαση σε περιεχόμενα που πρίν ίσως για κάποιο λόγο δεν ήταν προσβάσιμα .Χάρις στην ανάπτυξη του VPN και των τεχνολογιών του πολλοί χρήστες, ιδιώτες και επιχειρήσεις, επιλέγουν να κάνουν χρήση αυτής της τεχνολογίας με σκοπό να εξασφαλίσουν την πιο σημαντική λειτουργία που επιτελέι ένα τέτοιο δίκτυο που είναι η ιδιωτικότητα και η ασφάλεια μέσω του δημοσίου δικτύου. Έτσι ένας ιδιώτης μπορεί να έχει πρόσβαση σε περιεχόμενο που πριν δεν ήταν προσβάσιμο στην χώρα του αλλά και μια εταιρία να επεκταθεί και να έχει πρόσβαση στο δικτύό της όπου και να βρίσκεται. Η πτυχιακή χωρίζεται σε πέντε κεφάλαια.

Στο 1ο κεφάλαιο περιλαμβάνεται μια εισαγωγή στην έννοια των **V**irtual **P**rivate **N**etwork (VPN) μέσω του ορισμού θα εξετάσουμε την ασφάλεια ενός τέτοιου δικτύου και θα εξετάσουμε το πως μπορούμε να δημιουργήσουμε τέτοιου τύπου δικτύου, τους τύπους , τα τεχνικά τους χαρακτηριστικά καθώς και τις τεχνολογίες που διαθέτουν.

Στο 2ο κεφάλαιο εξετάσουμε μερικά προβλήματα ασφαλείας των VPN δικτύων . Θα εξετάσουμε προβλήματα κατα την αυθεντικοποίηση του χρήστη , απο ιούς και κακόβουλο λογισμικό , προβλήματα μη εξουσιοδοτημένης πρόσβασης , προβλήματα προσαρμοστικότητας των VPN δικτύων και γενικά ζητήματα ασφαλείας στα VPN δίτκυα.

Στο 3ο κεφάλαιο θα εξετάσουμε επιθέσεις σε VPN διακομιστές και τους τύπους επιθέσεων

Στο 4ο κεφάλαιο θα δούμε μεθοδολογίες για την ανίχνευση επιθέσεων

Τέλος, στο 5ο κεφάλαιο θα αναπτυχθεί ένα μοντέλο ανίχνευσης επιθέσεων στα VPN δίκτυα και το οποίο θα ανιχνεύει τις διάφορες επιθέσεις στο δίκτυο και θα μας ενημερώνει σχετικά με τον τύπο του και την ώρα που έγινε η επίθεση και εφόσον ελέγξει όλη την κυκλοφορία θα διατηρεί ένα log file με τις επιθέσεις και την ip της πηγής επίθεσης.

Λεξεις κλειδια : VPN, ασφάλεια, επιθέσεις δικτύου ,τούνελ, πρωτόκολλα

Abstract

The goal of this thesis is to develop a program in order to identify various network attacks in a VPN network, by studying their protocols (VPN) and the security that this type of network connection provides. Virtual Private Network (VPN) is a new way of connection for securely connect to the network and gain access in restricted content that previously was unavailable due to prohibition. Due to the vast development of VPN and its technologies most users and companies choose to use this kind of connection to make use of it's most important feature that's the privacy and security of the messages transmitted through a public network like internet. So, a user can have access to restricted content and a company can expand to different country, state and be like they are all together in the same network and have access to files and folders of the network. This thesis divided in five chapters.

The 1st chapter includes an introduction in the meaning of VPN and we will study it s security and how we can create this kind of VPN, different types of VPN, their technical specifications and their technologies that they consist of.

In 2nd chapter we will analyse some security issues of VPN networks during user authentication, due to virus and malware, unauthorized access, interoperability and VPN security issues in general.

In 3rd chapter we will analyse different kind of attacks in VPN and its compartments like server or client side like DoS attacks, flood attacks, DNS attacks.

In 4th chapter we will study some intrusion detection methodologies like mechanical learning.

In last, 5th chapter we will develop a program that will detect different kind of attacks and we will inform us about its type and time that this attack occurred and after it checked the whole traffic it will record a log file with the kind of attack and its source ip.

Key Words: VPN, security, tunnel, network attacks, protocols

Πίνακας Περιεχομένων

Συντομογραφίες

VPN ..... Virtual Private Network

OSI ..... Open Systems Interconnection

ESP ..... Encapsulating Security Payload

RFC…….....Request for Comments

AH ..... Authentication Header

IKE ..... Internet Key Exchange

ISAKMP ..... Internet Security Association and Key Management Protocol

DoS .....……. Denial of Service

PPP ….....……. Point to Point

AAA ….....……. Authentication, Authorization and Accounting

DNS.…….....……. Domain Name Server

UDP.…….....…. User Datagram Protocol

TCP.……Transmission Control Protocol

ICMP.……Internet Control Message Protocol


Κεφάλαιο 1
Εισαγωγή στα VPN δίκτυα

1. Εισαγωγή στα VPN δίκτυα

1.1 Τι είναι το VPN

Το VPN είναι μια τεχνολογία δικτύου η οποία επιτρέπει στους χρήστες της να δημιουργούν ασφαλείς συνδέσεις μέσω του μοντέλου server – client. Η τεχνολογία αυτή αναπτύχθηκε έτσι ώστε να ενισχύσει την ασφάλεια των δικτύων καθώς και την προσβασιμότητα σε ασφαλή δίκτυα , όπως για παράδειγμα δίκτυα εταιριών, δίκτυα συναλλαγών κ.α.

Το VPN μεταφέρει τα δεδομένα δημιουργώντας "τούνελ" και προωθώντας τα μέσα από αυτό. Για να γίνει αυτή η μεταφορά των δεδομένων το πακέτο του χρήστη ενσωματώνεται σε ένα άλλο πακέτο με μια νέα κεφαλίδα, η οποία περιέχει όλες τις πληροφορίες οι οποίες χρειάζονται για να "ταξιδέψει" το πακέτο μέσα στο μη ασφαλές δημόσιο δίκτυο. Από την άλλη πλευρά ο δέκτης του πακέτου το λαμβάνει, εξάγει το αρχικό πακέτο το οποίο είχε στείλει ο χρήστης και το προωθεί καταλλήλως.Το μονοπάτι το οποίο ακολουθεί το πακέτο στο δημόσιο δίκτυο ονομάζεται "τούνελ". Για να καταστεί δυνατή αυτή η μεταφορά και τα δύο άκρα του "τούνελ" θα πρέπει να υποστηρίζουν το ίδιο πρωτόκολλο .

Αυτό το "τούνελ" μπορεί να υπάρξει σε διαφορετικά επίπεδα του OSI Standard. Είτε στο επίπεδο 2 , που είναι το επίπεδο ζεύξης δεδομένων, είτε στο επίπεδο 3, το οποίο είναι το επίπεδο δικτύου.

Με αυτό το τρόπο ένας χρήστης που χρησιμοποιεί ένα ιδιωτικό δίκτυο μπορεί να στείλει πακέτα παγκοσμίως με ασφαλή τρόπο αυξάνοντας εικονικά το μέγεθος του ιδιωτικού του δικτύου.

1.2 Ιστορία του VPN

Η ασφάλεια των δικτύων από την αρχή της ανάπτυξης του Διαδικτύου ήταν κάτι το οποίο ήταν απαραίτητο, καθώς η αρχική του χρήση ήταν για σκοπούς του Αμερικάνικου Στρατού. Έτσι στις αρχές της δεκαετίας του 90 άρχισαν οι πρώτες εμφανίσεις των πρώιμων τεχνικών ασφάλειας για τα δίκτυα.

Μία από τις πρώτες και πρόδρομος του VPN ήταν η δουλειά των John Ioannides και Matt Blaze με τίτλο "The SwIPe IP Security Protocol" το οποίο αναπτύχθηκε σε δεξαμενές ιδεών του πανεπιστημίου Columbia και των AT&T Bell Laboratories. Αργότερα ο Wei Xu ξεκινώντας το 1994 δημιούργησε το πρωτόκολλο IPSec, ενώ ταυτόχρονα εμφανίστηκε και ένα άλλο πρωτόκολλο το ESP, το οποίο είχε αναπτυχθεί από την υπηρεσία έρευνας του Αμερικάνικου Ναυτικού (NAVAL Reseach) υπό την καθοδήγηση της Υπηρεσίας Προηγμένων ερευνών του Υπουργείου Εθνικής Αμύνης των ΗΠΑ (DARPA). Με όλη αυτή την πρόοδο στον τομέα της ασφάλειας των δικτύων δημιουργήθηκε το 1995 η IETF (Internet Engineering Task Force) με σκοπό της την εξέλιξη του διαδικτύου και την ομαλή του λειτουργία. Το 1996 ο Gurdeep Singh-Pall υπάλληλος στην Microsoft ανακαλύπτει το πρωτόκολλο PPTP το οποίο δημιουργεί μια πιο ασφαλή και ιδιωτική σύνδεση μεταξύ ενός υπολογιστή και του διαδικτύου.

Στην αρχή την τεχνολογία VPN με όποιο πρωτόκολλο και να υλοποιούταν την χρησιμοποιούσαν οι εταιρίες για την προστασία των δεδομένων του εντός της εταιρίας καθώς και για την επικοινωνία μεταξύ γραφείων σε διαφορετικές τοποθεσίες. Μέχρι εκείνη την ώρα για αυτό το λόγο χρησιμοποιούσαν ιδιωτικές γραμμές τις οποίες νοίκιαζαν από τον παροχέα τους, οι οποίες όμως είχαν πολύ μεγάλο κόστος αλλά και το μειονέκτημα ότι μόνο άτομα τα οποία είχαν φυσική παρουσία στον χώρο της εταιρίας μπορούσαν να έχουν πρόσβαση στα δεδομένα τα οποία ήταν αποθηκευμένα στο εσωτερικό της δίκτυο (Intranet).

Με την εξέλιξη όμως των μεθόδων ασφαλείας οι εταιρίες στράφηκαν στις λιγότερο ακριβές και περισσότερο χρήσιμες μεθόδους του VPN. Έτσι από τα καθαρά ιδιωτικά δίκτυα οδηγηθήκαμε στα εικονικά ιδιωτικά δίκτυα τα οποία χρησιμοποιούν γραμμές του διαδικτύου για να παρέχουν το ίδιο αποτέλεσμα με τις πρότερες μεθόδους.

Πλέον όλο και περισσότεροι χρήστες χρησιμοποιούν VPN για την πρόσβαση τους στο διαδίκτυο, όχι μόνο για επαγγελματικούς λόγους αλλά και για ιδιωτικούς. Με αυτό το τρόπο

μπορούν να βλέπουν περιεχόμενο το οποίο είναι μη προσβάσιμο από την χώρα τους , για περιεχόμενο το οποίο έχει για κάποιο λόγο απαγορεύσει ο πάροχός τους ή απλά για να ενισχύσουν την ασφάλεια κατά την περιήγηση τους στο διαδίκτυο.

1.3 Ασφάλεια VPN (Γιατί μας ενδιαφέρει)

Το κύριο χαρακτηριστικό των δικτύων VPN και ο λόγος που μας ενδιαφέρουν τόσο και γίνεται ολοένα και ευρύτερη η χρήση τους στο διαδίκτυο είναι η ασφάλεια και η προσβασιμότητα που προσφέρουν σε περιεχόμενα(για ιδιώτες) και υπηρεσίες(για επαγγλεματιες).

Όπως έχει αναφερθεί αυτό που κάνει το VPN να ξεχωρίζει απο τις υπόλοιπες τεχνολογίες δικτύου είναι η μετατροπή μιας δημόσιας σύνδεσης (ιντερνετ) σε ιδιωτική μέσω μίας εικονικής σύνδεσης. Αυτό σημαίνει το ότι περνάει μέσα απ αυτή την ιδιωτική σύνδεση θα πρέπει να είμαστε σίγουροι οτι για οποιοδήποτε λόγο δεν έχει κλαπεί ή παραβιαστεί το περιεχόμενο του μυνήματος/πακέτου που μεταφέρεται μέσω αυτής της σύνδεσης.Γενικά, για να θεωρείται ένα VPN ασφαλές θα πρέπει να παρακολουθείται και να ενημερώνεται τακτικά μόνο έτσι μπορούμε να γνωρίζουμε αν το VPN μας είναι ασφαλές ή όχι.

Πρόσφατα ανακοινώθηκε ότι βρέθηκε ένας τρόπος να παραβιάζουν ακόμα και το πιο ασφαλές πρωτόκολλο ευρείας χρήσης , το WPA 2,το οποίο χρησιμοποιούσε κλειδί 256 bit με την χρήση εξαναγκασμένης επίθεσης. Αυτό επηρεάζει κάθε χρήστη που χρησιμοποιεί ασύρματο δίκτυο.

Παρόλα αυτά το VPN και τα πρωτόκολλά του ακόμα θεωρούνται τα πιο ασφαλή τα οποία υπάρχουν.

Τα χαρακτηριστικά που κάνουν το VPN τόσο ασφαλές είναι η κρυπτογράφηση,η σωστή δημιουργία των τούνελ, η εξακρίβωση χρήστη και ο έλεγχος ακεραιότητας των δεδομένων.

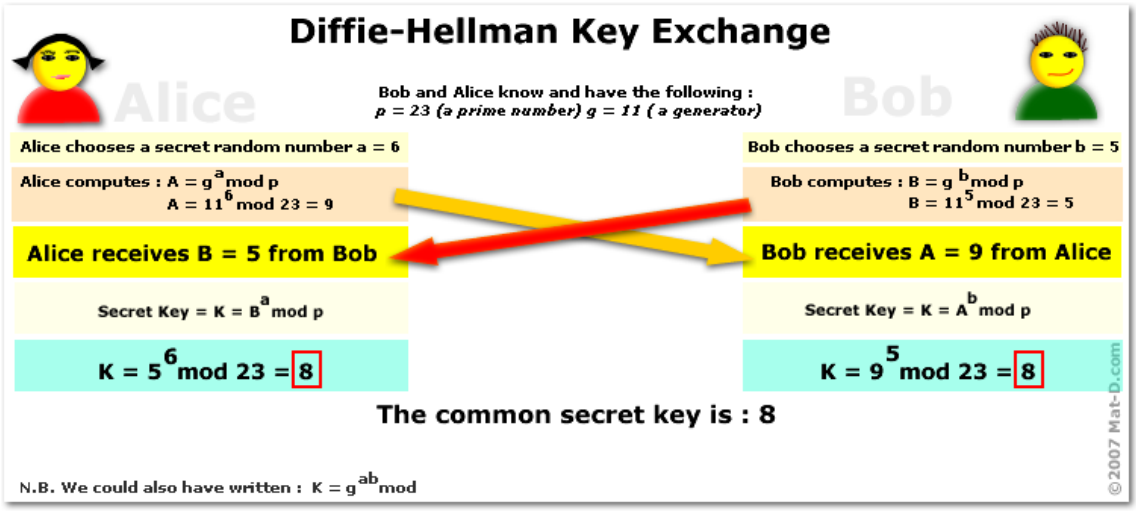
1.3.1 Κρυπτογράφηση

Το VPN χρησιμοποιεί μηχανισμούς κρυπτογράφησης για να προστατέψει τα δεδομένα που περνάν μέσα από το “τούνελ” του. Το πακέτο το οποίο μεταφέρεται από τον χρήστη στον διακομιστή της VPN υπηρεσίας κρυπτογραφείται για να μεταφερθεί μέσα από το διαδίκτυο. Έτσι το πακέτο με την καινούρια κεφαλίδα που προσθέτει ο διακομιστής μπορεί με ασφάλεια να μεταφερθεί μέσα από το δίκτυο. Η κρυπτογράφηση των δεδομένων χρησιμοποιεί μια αλγοριθμική διαδικασία χρησιμοποιώντας ένα κρυφό κλειδί για να μεταφέρει ενα απλο κείμενο σε κωδικοποιημένη μορφή. Αυτό γίνεται με σκοπό να αποτρέψει οτι κανένας άλλος εκτός απο τον συγκεκριμένο παραλήπτη θα έχει πρόσβαση στην πληροφορία. Χάρις σε αυτο του το χαρακτηριστικό ένας χρήστης που δεν έχει το δικαίωμα να δεί αυτη την πληροφορία χωρίς να διαθέτει το κατάλληλο μέσω για την αποκρυπτογράφηση της πληροφορίας κάνει την πληροφορία αυτη που μεταφέρεται ακατανόητη. Επιπλέον, η τεχνική αυτή παρέχει εμπιστευτικότητα και ιδιωτικότητα στην πληροφορία που ανταλλάσεται. Εφόσον δημιουργηθεί το τούνελ διάφοροι αλγόριθμοι όπως ο AES ο οποίος λειτουργεί πάνω στο τούνελ για να αποκρύψει την πληροφορία η οποία μεταφέρεται και επίσης στα κλειδιά και ειδικά στο VPN χρησιμοποιείται ο αλγόριθμος του Diffie-Hellman (OnlineLMS,2017)

- Diffie-Hellman key exchange

  Ονομάζεται και αλλιώς εκθετική ανταλλαγή κλειδιού, είναι μια μέθοδος ψηφιακής κρυπτογράφισης η οποία χρησιμοποιεί αριθμούς υψωμένους σε συγκεκριμένες δυνάμεις για να παράγει κλειδιά αποκρυπτογράφισης στην βάση των στοιχείων τα οποία δεν μεταδίδονται άμεσα , κάνοντας το «σπάσιμο» μαθηματικά απαιτητικό. Ο αλγόριθμος αυτός περιορίζεται στο οτι υστερεί της αυθεντικοποίησης . Οι επικοινωνίες που

χρησιμοποιούν τον αλγόριθμο αυτό καθ'αυτό είναι επιρεπείς σε επιθέσεις τύπου man-in-the-middle,γι αύτο ο αλγόριθμος χρησιμοποιείται και με μηχανισμούς αυθεντικοποίησης χρήστη όπως οι ψηφιακές υπογραφές για να πιστοποιηθεί η ταυτότητα των χρηστών.



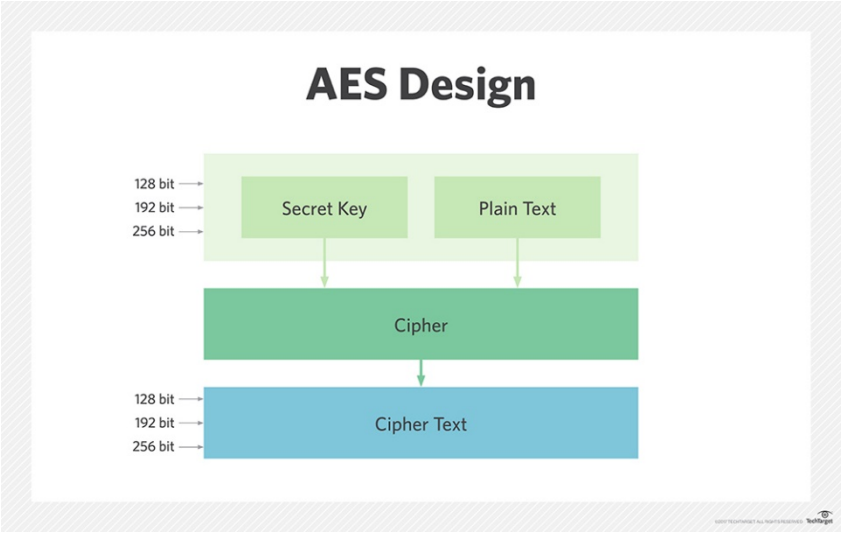
1.1 Παράδειγμα Diffie-Hellman (google.gr, 2017)

1.3.2 Tunneling

Εξ' ορισμού το VPN δημιουργεί οπως, αναφέρθηκε και εισαγωγικά , τούνελ για την προώθηση των δεδομένων με ασφάλεια. Αυτή η διαδικασία επιτρέπει την ενσωμάτωση (encapsulation) του πακέτου απο τον τον ένα τύπο πρωτοκόλλου μέσα στο πακέτο ενός διαφορετικού πρωτοκόλλου . Για παράδειγμα το VPN χρησιμοποιεί το PPTP, το οποίο θα αναφέρουμε παρακάτω, για να ενσωματώσει IP πακέτα μέσα απο ένα δημόσιο δίκτυο όπως το ίντερνετ.

- Advanced Encryption Data(AES) (Rouse M,2017)

  Αποτελεί τον διάδοχο του Data Encryption Standard(DES) και σχεδιάστηκε για να είναι εύκολο να εφαρμοστεί και σε hardware και σε software όπως επίσης και σε περιορισμένα μέσα όπως π.χ μια smart card και προσφέρει αρκετή προστασία εναντίον διαφόρων επιθέσεων.Αποτελεί επίσης τον πιο διάσημο αλγόριθμο κρυπτογράφισης καθώς χρησιμοποιείται απο την κυβέρνηση των ΗΠΑ για να μεταφερθούν απόρρητες πληροφορίες και επιπλέον είναι πιο ασφαλής απο τον DES ,γρηγορότερος στην κρυπτογράφιση που το καθιστά ιδανικό για εφαρμογές ,υλικολογισμικό(firmware) και hardware που απαιτούν είτε μικρή καθυστέριση ή υψηλή διεκπαιρεότητα όπως τα firewall και τα router.Αποτελείται απο τρία μπλοκ κώδικα AES-128bit, AES-192bit, AES-256bit. Κάθε ένα απ αυτά τα μπλόκ κρυπτογραφεί/αποκρυπτογραφεί τα δεδομένα με μέγεθος του μπλοκ αυτού 128bit χρησιμοποιώντας κλειδιά 128,192 και 256 αντίστοιχα.



1.2 Σχέδιο AES (Rouse M,2017)

Χρησιμοποιώντας συμμετρικό(γνωστό και ως κρυφό κλειδί) κώδικα χρησιμοποιεί το ίδιο κλειδί για να κρυπτογραφεί/αποκρυπτογραφεί για να είναι σε θέση ο αποστολέας και ο παραλήπτης να γνωρίζουν και να χρησιμοποιούν το ίδιο κρυφό κλειδί. Το κάθε κλειδί έχει το δικό του αριθμό μετασχηματισμών(rounds) . Για τα 128bit έχουμε 10 μετ , για 192-12μετ και 256-14. Με τον όρο μετασχηματισμό(round) εννοούμε τα βήματα επεξεργασίας τα οποία περιλαμβάνουν αλλαγή,μετάθεση και μίξη στο κείμενο που δίνεται ως είσοδο, στην περιπτωσή μας τα μυνήματα που μεταφέρονται στο δίκτυο, και μετατροπή του σε κωδικοποιημένο κείμενο.

1.3.3 Εξακρίβωση Χρήστη

Στην τεχνολογία VPN υπάρχει η δυνατότητα και για δευτερεύουσα αυθεντικοποίηση χρήστη. Αυτό μπορεί να γίνει μέσω κάποιου ή συνδυασμών κάποιων από τα ακόλουθα:

1. Όνομα Χρήστη – Κωδικός
2. Γεννήτρια Ψευδοτυχαίων αριθμών
3. Κωδικοί Καρτών

1.3.4 Έλεγχος ακεραιότητας δεδομένων

Περιοδικά ο διακομιστής αλλά και ο πελάτης μπορεί να πραγματοποιήσει έλεγχο ακεραιότητας των δεδομένων που λαμβάνει έτσι ώστε να καταλάβει αν τα δεδομένα έχουν με κάποιο τρόπο αλλαχτεί και να αναγνωρίσει πιθανή παραβίαση ασφαλείας. Ο έλεγχος αυτός γίνεται μέσω συναρτήσεων κατακερματισμού (hash).Αποτελεί δηλαδή ένα τρόπο για να ελέγξουμε αν τα δεδομένα μας έχουν ή δεν έχουν τροποποιηθεί Χρησιμοποιείται κυρίως για να παρέχει ενα ψηφιακό αποτύπωμα για οποιοδήποτε τύπο δεδομένων ετσι ώστε να διασφαλίζει οτι η πληροφορία δεν έχει αλλαχτεί κατα την διάρκεια της μετάδοσης.

- Είναι ένας μοναδικός αριθμός μονής κατεύθυνσης μαθηματικά παραγμένος απο μία ακολουθία κειμένου στην οποία εφαρμόζεται ένας μαθηματικός τύπος

- Ο μοναδικός αριθμός υπολογίζεται βάση ένος απλού κείμενου δεδομένων
- Το αρχικό μύνημα δεν μπορεί να επαναδομηθεί χρησιμοποιώντας αυτό τον μοναδικό αριθμό παρόλη την γνώση του αλγορίθμου
- Ο μοναδικός αυτός αριθμός λειτουργεί ως «δαχτυλικό» αποτύπωμα του μυνήματος(OnlineLMS,2017)

Μερικές γνωστές συναρτήσεις κατακερματισμού για τον έλεγχο των δεδομένων είναι οι MD5 και ο SHA

- MD5(Message Digest 5)

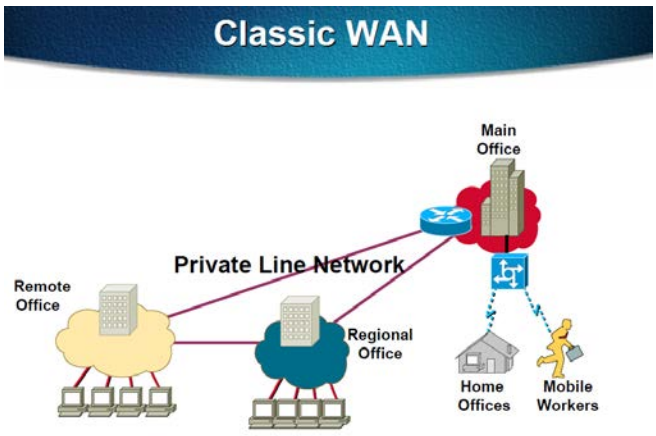
Η είσοδος του αλγορίθμου αυτού είναι ένα κείμενο τυχαίου μεγέθους και η εξοδός του το ψηφιακό αποτύπωμα με μέγεθος 128bit.(Cetinkaya, 2016). Αποτελεί προέκταση του MD4 ο οποίος ήταν ένας πολύ γρήγορος αλγόριθμος και επίσης επιρεπής σε επιθέσεις. Ο MD5(RFC 1321) είναι πιό αργός και συντηρητικός ως προς την σχεδιασή του για να παρέχει την καλύτερη δυνατή ασφάλεια. Επιπλέον δεν είναι ανεκτικός σε συγκρούσεις (collision resistant) . Ως σύγκρουση ορίζουμε όταν δύο τιμές κατακερματισμού είναι παρόμοιες ή ίδιες.(Techopedia, 2017)

- SHA (Secure Hashing Algorithm)

Αποτελεί απο τους πιο γνωστούς αλγορίθμους κατακερματισμού και ελέγχου για την ακεραιότητα των δεδομένων και χρησιμοποιείται σε πολλές εφαρμογές ασφαλείας και πρωτόκολλα όπως το IPSec, SSL,TLS,SSH κ.α. Αποτελεί εξέλιξη του MD5 και αποτελείται και απο διάφορες εκδόσεις απο την αρχική έκδοση SHA-0 ,SHA-1 και ως σήμερα χρησιμοποιούνται οι εκδόσεις SHA2 και SHA3 . Αυτή που χρησιμοποιείται και τώρα είναι η έκδοση SHA-2. Συνδυάζει ένα κλειδί και ένα τμήμα μη κρυπτογραφημένων δεδομένων μήκους 128 bit.Ο SHA-2 αποτελείται και αυτός απο εκδόσεις που καθορίζονται απο τον αριθμό των bits του μυνήματος που μπορεί να κατακερματίσει. Οι πιο γνωστές εκδόσεις είναι SHA-224,256,384,512.

1.4 Πως δημιουργείται ένα VPN

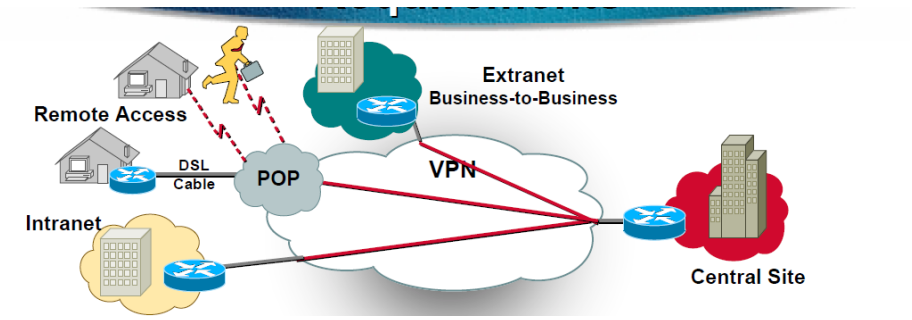
Όπως αναφέρθηκε στο υποκεφάλαιο 1.2 αρχικά στα πρώϊμα στάδια του VPN οι εταιρίες και οργανισμοί κυρίως που έκαναν χρήση της τεχνολογίας αυτής νοίκιαζαν με υψηλό κόστος ιδιωτικές γραμμές ωστε να έχουν πρόσβαση και σε γραφεία που ήταν σε απομακρυσμένα σημεία είτε σε κάποια αλλη χώρα ή πόλη. Όμως, χάρις στην εξέλιξη του διαδικτύου και των τεχνολογιών που το διέπουν απο τις ακριβές ιδιωτικές γραμμές περάσαμε στο VPN .



1.3 Κλασσικό WAN (Cisco sys,2000)

Για να δημιουργηθεί ένα VPN δίκτυο πρώτο και κύριο και απαραίτητο είναι να υπάρχει ένας διακομιστής ο οποίος παρέχει τις VPN υπηρεσίες. Αυτός ο διακομιστής μπορεί να είναι είτε κάποιος δημόσιος επί πληρωμή όπως, υπάρχουν πολλά site τα οποία προσφέρουν αυτές τις υπηρεσίες, είτε να στηθεί εκεί που χρειάζεται ένας ιδιόκτητος διακομιστής.

Η πρώτη επιλογή είναι αυτή που συνήθως επιλέγεται από ιδιώτες και μικρές επιχειρήσεις, ενώ η δεύτερη προτιμάται από μεγάλες εταιρίες.



1.4 Σύνδεση με χρήση VPN (Cisco sys,2000)

Κατόπιν, πρέπει να ληφθούν υπόψιν τα χαρακτηριστικά του δικτύου VPN τα οποία χρειάζονται. Για κάθε σύνδεση VPN μεταξύ των δύο άκρων θα πρέπει να αποφασιστούν τα ακόλουθα στοιχεία της σύνδεσης ανάλογα με τον τύπο και τι υποστηρίζει το κάθε πρωτόκολλο:

- Κρυπτογράφηση
- Πιστοποίηση Χρήστη
- Πιστοποίηση Διακομιστή
- Μηχανισμοί Ασφαλείας
- Ακεραιότητα Δεδομένων

Κατόπιν ενεργοποιείται η δυνατότητα της σύνδεσης στο VPN δίκτυο και της ασφαλούς μεταφοράς των δεδομένων.

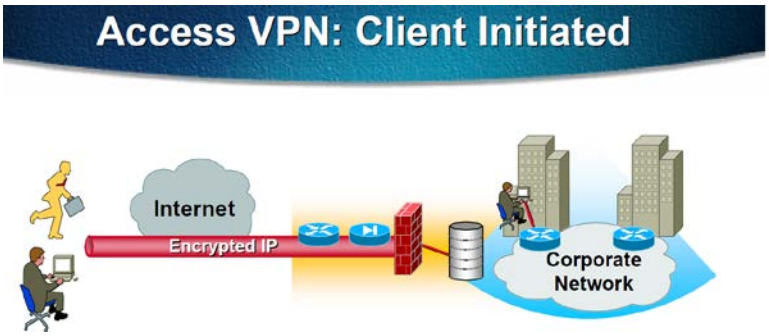
1.5 Τύποι VPN δικτύων

Τα VPN δίκτυα μπορούν να διαχωριστούν σε δύο διάφορες κατηγορίες ανάλογα με τον τρόπο τον οποίο υλοποιούνται. Οι τύποι δικτύων είναι:

1.5.1 VPN απομακρυσμένης πρόσβασης

Αυτή είναι μια σύνδεση με βάση το χρήστη η οποία χρησιμοποιείται για οικιακή χρήση ή για ένα χρήστη ο οποίος δουλεύει απομακρυσμένα και θέλει να συνδεθεί στο ιδιωτικό δίκτυο της

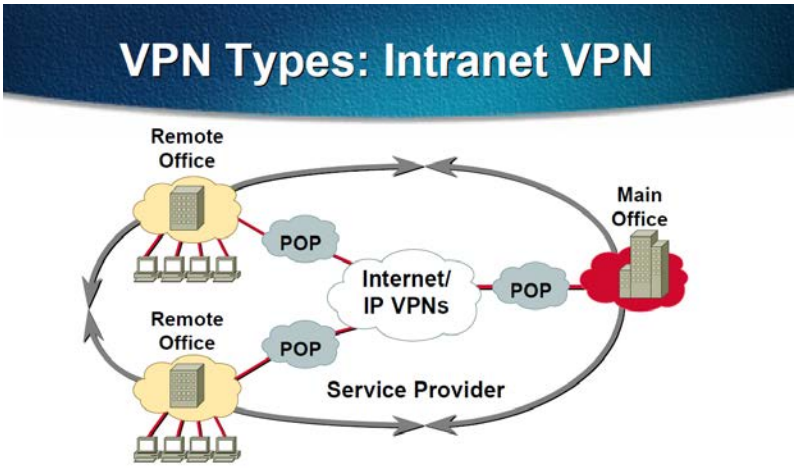
εταιρίας. Αυτού του είδους το VPN επιτρέπει ασφαλείς, κρυπτογραφημένες συνδέσεις μεταξύ του εταιρικού δικτύου και του απομακρυσμένου χρήστη.



1.5 VPN απομακρυσμένης πρόσβασης (Cisco sys,2000)

1.5.2 Intranet VPN

Σε αυτή την υλοποίηση, το VPN χρησιμοποιείται για να δημιουργήσει συνδέσεις μεταξύ συγκεκριμένων τοποθεσιών όπως ένα δίκτυο γραφείων.Αυτού του τύπου VPN με βάση το δίκτυο ενώνει κάτω απο ένα και μόνο ιδιωτικό δίκτυο πολλαπλές απομακρυσμένες τοποθεσίες. Επίσης, η υλοποίηση αυτή μειώνει το κόστος γραμμής και τα λειτουργικά κόστη που είναι ένα απο τα κύρια χαρακτηριστικά μια εταιρίας δηλαδή να έχει όσο μεγαλύτερη παραγωγή με το μικρότερο δυνατό κόστος και ως προς αυτή την κατεύθυνση το VPN και οι τεχνολογίες του έχουν συμβάλλει καθοριστικά.

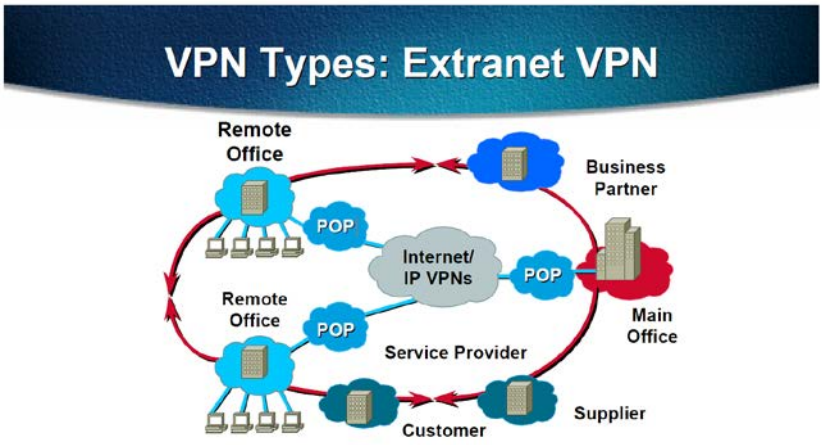


1.6 Intranet VPN (Cisco sys, 2000)

1.5.3 Extranet VPN

Εδώ το VPN χρησιμοποιείται για να συνδέσει εταιρικούς συνεργάτες,όπως προμηθευτές και πελάτες, έτσι ώστε όλοι να μπορούν να δουλεύουν με ασφαλές δεδομένα σε ένα κοινόχρηστο περιβάλλον. Μπορούμε να χαρακτηρίσουμε αυτού του είδους το VPN ως μια αντικατάσταση των δικτύων WAN(Wide Area Network). Η συντήρηση ενός τέτοιου δικτύου είναι ακριβή

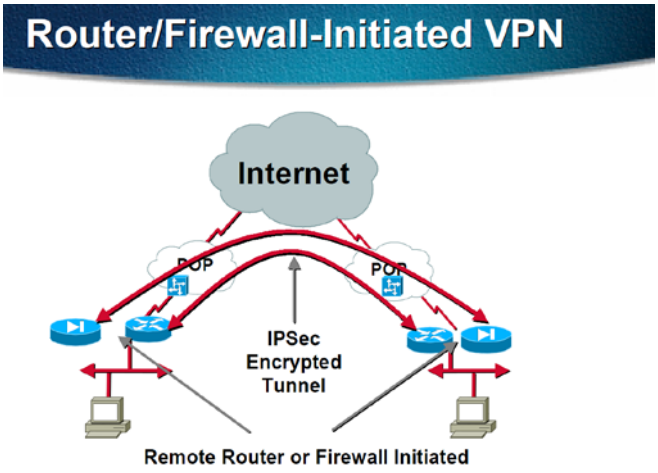
είδικά αν τα δίκτυα που το αποτελούν είναι γεωγραφικά διάσπαρτα. Το VPN προσφέρει μεγαλύτερη και με μικρότερο κόστος επεκτασιμότητα απο τις παραδοσιακές ενοικιαζόμενες ιδιωτικές γραμμές.



1.7 Extranet VPN (Cisco sys,2000)

1.5.4 Router/Firewall VPN

Είναι ένα δίκτυο που είναι εξοπλισμένο με δυνατότητες υλοποίησης τείχου προστασίας (firewall) και δυνατότητες VPN .Αυτού του είδους το VPN κάνει χρήση των μηχανισμών ασφαλείας που υπάρχουν στο firewall για να περιορίσει την πρόσβαση στο εσωτερικό δίκτυο. Αυτός ο τύπος χρησιμοποιείται και απο το intranet και το extranet και επίσης παρέχει ειδοποιήσεις σε πραγματικό χρόνο ,αυθεντικοπόιηση χρήστη και αναλυτικά log files(αρχεία ελέγχου πρόσβασης).



1,8 Router/Firewall VPN (Cisco sys,2000)

1.6 Τεχνικά χαρακτηριστικά δικτύων VPN

Για την λειτουργία των VPN δικτύων χρειάζονται διάφορα στοιχεία δικτύου τα οποία επιτρέπουν τη δημιουργία του τούνελ μέσα από το οποίο ο χρήστης θα μπορέσει να μεταφέρει με ασφάλεια και ανωνυμία τα δεδομένα του.

Πρώτο και κύριο είναι το λογισμικό το οποίο θα τρέξει στην πλευρά του χρήστη και θα μπορέσει να πραγματοποιήσει όλες τις διαδικασίες(ανταλλαγή πρωτοκόλλων, αυθεντικοποίηση, ενσωμάτωση πακέτων) οι οποίες είναι απαραίτητες για να γίνει η διαπραγμάτευση και η δημιουργία του τούνελ. Συνήθως κάθε πάροχος υπηρεσιών VPN έχει το δικό του λογισμικό το οποίο επιτρέπει όλες αυτές τις λειτουργίες.

Δεύτερο είναι **ο διακομιστής** ο οποίος παρέχει την υπηρεσία VPN. Ο διακομιστής πρέπει να είναι ικανός να παρέχει στον χρήστη όλες τις υπηρεσίες οι οποίες χρειάζονται με ασφάλεια.

Τέλος πρέπει όλοι οι **δρομολογητές** και τα **firewall** τα οποία βρίσκονται ανάμεσα στον χρήστη και τον διακομιστή να μπορούν να υποστηρίζουν υπηρεσίες VPN. Συνήθως οι περισσότεροι σύγχρονοι δρομολογητές μπορούν να διαχειριστούν τέτοιου είδους δεδομένα και τα firewall αφήνουν αυτού του είδους τα δεδομένα να περάσουν.

Εξαιρέσεις υπάρχουν σε χώρες στις οποίες δεν επιτρέπεται η χρήση τέτοιων υπηρεσιών που οι ίδιοι οι πάροχοι υπηρεσιών διαδικτύου υλοποιούν τα firewall κατά τέτοιο τρόπο ώστε να μην καθίσταται δυνατή η δημιουργία αυτού του τούνελ. Τέτοιες χώρες είναι χώρες με υψηλή λογοκρισία στο διαδίκτυο όπως η Κίνα, το Ιράκ, η Τουρκία, το Ιράν, η Ρωσία (απαγόρευση χρήσης μόνο σε ορισμένες διευθύνσεις ιστού) , η Συρία και η Βόρεια Κορέα.

Επιπλέον υπάρχουν και υπηρεσίες οι οποίες δεν υποστηρίζουν τη χρήση VPN όπως το Hulu και το Netflix.

1.7 Πρωτόκολλα VPN

Για την υλοποίηση ενός “τούνελ” στο VPN μπορούν να χρησιμοποιηθούν διαφορετικά πρωτόκολλα ανάλογα με τον τρόπο και τα χαρακτηριστικά τα οποία θέλουμε να έχει ο τύπος σύνδεσης με την VPN υπηρεσία. Συγκεκριμένα τα πρωτόκολλα τα οποία μπορούν να χρησιμοποιηθούν είναι:

1.7.1 IPSec (Internet Protocol Security)

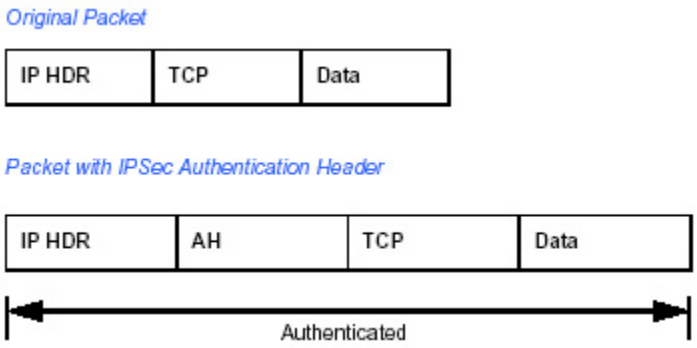
Το πρωτόκολλο IPSec είναι μια ομάδα πρωτοκόλλων τα οποία αρχικώς ορίζονται στα RFC 1825, 1826 και 1827.Το RFC(request for comment) είναι μια ομάδα έγγραφων κειμένων η οποία εμπεριέχει πολλές πτυχές του δικτύου όπως πρωτόκολλα, διαδικασίες, προγράμματα καθώς επίσης σημειώσεις,γνώμες ακόμα και χιούμορ(ieft.org,2017). Αναπτύχθηκε απο την IETF(Internet Engineering Task Force) για ασφαλή μεταφορά πληροφοριών στο επίπεδο 3 του OSI μέσω του δημοσίου και απροστάτευτου δικτύου το ίντερνετ.Είναι απο τους πιο ασφαλείς εμπορικούς τρόπους που υπάρχουν διαθέσιμοι για σύνδεση μεταξύ ιστοσελίδων.Το IPSec είναι περιορισμένο μόνο να μεταδίδει IP πακέτα. Το πρωτόκολλο αυτό δίνει την δυνατότητα στο σύστημα να επιλέξει και να διαπραγματευτεί τα απαιτούμενα πρωτόκολλα ασφαλείας, τους αλγορίθμους και τα κρυφά κλειδιά που απαιτούνται για τις απαιτούμενες υπηρεσίες. Παρέχει δηλαδή βασική αυθεντικοποίηση , ακεραιότητα δεδομένων και υπηρεσίες κρυπτογράφισης με σκοπό να προστατεύσει μη εξουσιοδοτημένη τροποποίηση των δεδομένων. Επιπλέον, η ασφάλεια που παρέχει γίνεται πάνω στα IP πακέτα. Με τον όρο πακέτο εννοούμε πολλές πληροφορίες μαζί οι οποίες είναι οργανωμένες

για μετάδοση σε όλο το δίκτυο, και περιλαμβάνει μια κεφαλίδα και ένα φορτίο(payload) την πληροφορία δηλαδή που βρίσκεται μέσα σε αυτό το πακέτο.

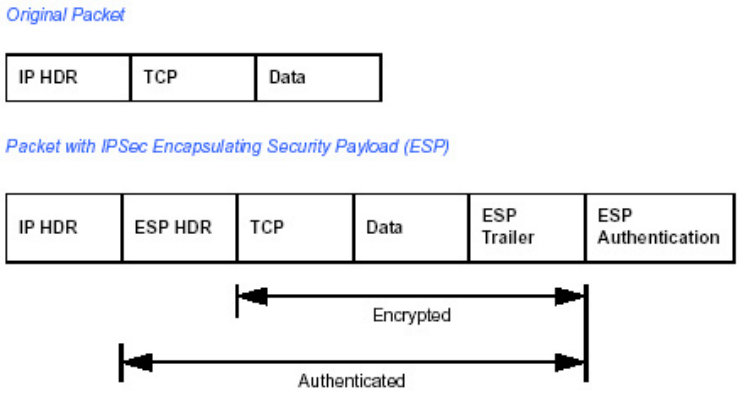
Τύποι Λειτουργίας

Το IPSec εισάγει την έννοια του Συσχετισμού Ασφαλείας(Security Association (SA)) . Αποτελεί μια λογική σύνδεση μεταξύ δύο συσκευών οι οποίες μεταφέρουν δεδομένα. Πρόσφερει ασφάλεια δεδομένων για κίνηση μονής κατεύθυνσης χρησιμοποιώντας τα καθορισμένα πρωτόκολλα του IPSec. Ένα τέτοιο τούνελ τυπικά αποτελείται απο δύο SAs, που μαζί παρέχουν ενα προστατευμένο διπλής κατεύθυνσης κανάλι δεδομένων. Τα SA επιτρέπουν σε μία επιχείρηση να έχει πλήρως τον έλεγχο ποιοί πόροι θα επικοινωνούν με ασφάλεια σύμφωνη με την πολιτική προστασίας της εταιρίας. Για να επιτευχθεί αυτό μια επιχείρηση μπορεί να «στήσει» πολλά SA με σκοπό να δημιουργήσει πολλά VPN καθώς και να αναπτύξει επιπλέον SAs μέσα στο VPN για να υποστηρίξει και άλλα τμήματα ή συνεργάτες της εταιρίας. Οι συσχετισμοί αυτόι(SA) χρησιμοποιούν δύο τρόπους λειτουργίας. Ο ένας ονομάζεται τύπος τούνελ και ο άλλος τύπος μεταφοράς. Ο τύπος τούνελ χρησιμοποιείται για gateway-to-gateway IPSec προστασία ενώ ο τύπος μεταφοράς για host-to-host IPSec προστασία. Με τον όρο gateway(πύλη δικτύου) , εννοούμε την συσκευή εκείνη η οποία είναι υπεύθυνη να ελέγχει και να παρακολουθεί την εισερχόμενη και εξερχόμενη κίνηση και κατευθύνει ανάλογα την κίνηση . Ο host(εξυπηρετητής) είναι μια συσκευή η οποία στέλνει και λαμβάνει κίνηση.(Netgear Inc, 2005)

- Τύπος μεταφοράς: Στον τύπο μεταφοράς δεν γίνεται αλλαγή των κεφαλίδων IP και αυτές που μεταφέρονται είναι οι κεφαλίδες του αρχικού πακέτου(με τις διευθύνσεις IP της πηγής και του προορισμού να μην αλλάζουν) και το επεξεργασμένο φορτίο του πακέτου(packet payload). Απ ότι φαίνεται κατα τον τύπο αυτό η κεφαλίδα IP δεν προστατεύεται και έτσι κάποιος επιτιθέμενος μπορέι να μάθει απο που πρόέρχεται και που πάει το πακέτο.

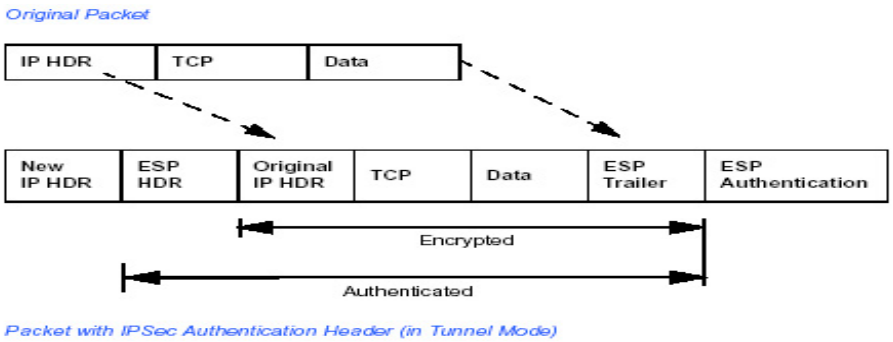


1.9 Τυπος μεταφοράς με χρήση AH(NetgearInc,2005)



1.10 Τύπος μεταφοράς με χρήση ESP (Netgear Inc,2005)

- Τύπος τούνελ: Στον τύπο αυτό όλο το πακέτο ενσωματώνεται και γίνεται το φορτίο(payload) το οποίο κάνει χρήση του IPSec. Μια νέα κεφαλίδα IP δημιουργείται η οποία περιέχει τις δύο διευθύνσεις των gateways. Οι gateways πραγματοποιούν την ενσωμάτωση /αποενσωμάτωση εκ μέρους των εξυπηρετητών. Αυτό αποτρέπει τον επιτιθέμενο απο το να υποκλέψει και να αναλύσει τα δεδομένα και επίσης δεν είναι σε θέση να γνωρίζει απο πού και πρός πάνε τα πακέτα.

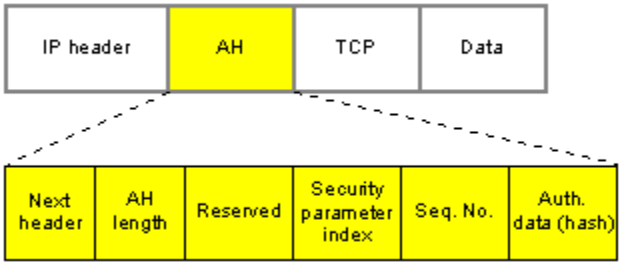


1.11 Τύπος τούνελ με AH(Netgear Inc,2005)

1.7.1.1 Το πρωτόκολλο AH

Το πρωτόκολλο AH, παρέχει αυθεντικοποίηση και ακεραιότητα των δεδομένων, τα οποία προστατεύουν απο αλλαγές στα δεδομένα , χρησιμοποιόντας τους ίδιους αλγορίθμους με το ESP. Παρέχει επίσης επιλεκτική anti-replay προστασία, η οποία προστατεύει μη εξουσιοδοτημένη αναμετάδοση των πακέτων όμως δεν παρέχει κρυπτογράφιση των δεδομένων. Εαν για κάποιο λόγο η μετάδοση των δεδομένων διακοπεί και μόνο αυτό το πρωτόκολλο χρησιμοποιείται τα περιεχομένα του μυνήματος μπορούν να διαβαστούν. Γι αυτό συνήθως το AH και το ESP χρησιμοποιούνται μαζί για επιπλέον προστασία. Η κεφαλίδα αυθεντικοποίησης(AH) ενσωματώνεται στο πακέτο ανάμεσα απο την κεφαλίδα IP και τα υπόλοιπα περιεχόμενα του πακέτου. Το φορτίο δεν αλλάζει.(Netgear Inc, 2005)

Η μορφή της κεφαλίδας του AH πρωτοκόλλου είναι :



1.12 Μορφή AH και ενσωμάτωση σε πακέτο IP(HKSAR,2008)

Τα πεδία της κεφαλίδας είναι :

Next Header : Ταυτοποιεί το επόμενο πρωτόκολλο στην στοίβα μετά την κεφαλίδα AH

Payload Length : Το μήκος των πιστοποιημένων δεδομένων σε πολλαπλάσιο των 32 bit

Reserved : Πρέπει να τεθεί σε μηδενικά πριν την αποστολή

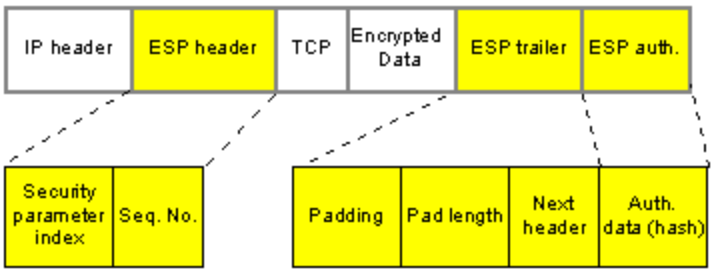
Security Parameter Index (SPI) : ένας 32bit ψευδοτυχαίος αριθμός ο οποίος ταυτοποιεί τις παραμέτρους ασφαλείας για αυτό το πακέτο. Οι SPI από 0 έως 255 είναι δεσμευμένοι από την IANA, εκτός αν το συγκεκριμένο έχει προσδιοριστεί σε RFC.

Authentication Data : Ο υπολογισμός της τιμής των πιστοποιημένων δεδομένων. Πάντα πρέπει να είναι πολλαπλάσιο των 32bit. Οι συνήθεις αλγόριθμοι που χρησιμοποιούνται είναι οι : HMAC-SHA1-96 [RFC2404], AES-GMAC with AES-128 [RFC4543], AES-XCBC-MAC-96 .

1.7.1.2 Το πρωτόκολλο ESP

Το πρωτόκολλο ESP είναι το δεύτερο κομμάτι της ομάδας πρωτοκόλλων του IPSec. Επιπλέον των χαρακτηριστικών του AH το ESP παρέχει και το απόρρητο του πακέτου. Χρησιμοποιεί αλγορίθμους κρυπτογράφισης όπως ο SHA ή ο MD5 . Οι αλγόριθμοι που χρησιμοποιεί το IPSec παράγουν ένα μοναδικό αναγνωριστικό για κάθε πακέτο και το οποίο αποτελεί την ταυτοτητά του. Αυτό επιτρέπει σε μια συσκευή να καθορίσει αν ένα πακέτο έχει αλλαχτεί. Επιπλέον, τα πακέτα που δεν έχουν αυθεντικοποιηθεί διαγράφοντε και δεν παραδίδονται στον παραλήπτη. Οι αλγόριθμοι που χρησιμοποιούνται θα πρέπει να είναι οι ίδιοι και στον αποστολέα και στον παραλήπτη. Το ESP παρέχει την κρυπτογράφιση . Χάρις σε αυτή του την λειτουργία επιτρέπει μόνο στον αποστολέα και στον παραλήπτη να διαβάσουν τα δεδομένα. Επίσης επιπρόσθετα μπορεί να πραγματοποιήσει αυθεντικοποίηση . Η αυθεντικοπόίηση αυτή που ονομάζεται ESP authentication, προσφέρει αυθεντικοποίηση και ακεραιότητα για το φορτίο και όχι για την κεφαλίδα IP.

Η μορφή της κεφαλίδας του πρωτοκόλλου ESP είναι:



1.13 Μορφή ESP και ενσωμάτωση σε πακέτο IP(HKSAR,2008)

Τα πεδία της κεφαλίδας είναι :

Security Parameter Index (SPI) : το SPI στο ESP είναι ένας αυθαίρετος αριθμός μήκους 32 bit ο οποίος ταυτοποιεί τους μηχανισμούς ασφαλείας οι οποίοι διέπουν ένα πακέτο στην μετάδοση. Αν η υλοποίηση του IPSec υποστηρίζει πολλαπλούς αποδέκτες (multicast) τότε πρέπει να γίνεται ταυτοποίηση έτσι ώστε να αποπολυπλέκονται οι ροές των δεδομένων σωστά

Sequence Number : Περιέχει έναν αριθμό μεγέθους 32 bit ο οποίος αυξάνεται για κάθε πακέτο που υπάρχει στην συγκεκριμένη μετάδοση. Αυτό από μόνο του προσφέρει προστασία ενάντια σε επιθέσεις επανάληψης (replay attack). Όταν η διαχείριση κλειδιού γίνεται με το IKEv2 τότε αυτό το πεδίο έχει μήκος 64 bit έτσι ώστε να υπάρχει καλύτερη ασφάλεια των δεδομένων. Σε αυτή τη περίπτωση μεταφέρονται μόνο τα 32 κατώτερα bit και τα ανώτερα 32 bit τα κρατάει ο αποστολέας και ο δέκτης και υπολογίζονται στην τιμή του ICV.

Initialization Vector (IV) : Το διάνυσμα αρχικοποίησης είναι ένα προαιρετικό πεδίο στην κεφαλίδα του ESP το οποίο τροφοδοτείται στον αλγόριθμο κρυπτογράφησης για την κρυπτογράφηση αποκρυπτογράφηση του πακέτου. Τέτοιο μπορεί να είναι η ώρα, κάποιος αύξων αριθμός ή γενικά κάποιο στοιχείο συγχρονισμού.

Payload Data : Περιέχει τα κρυπτογραφημένα πλέον δεδομένα του αρχικού πακέτου. Σε τύπο τούνελ περιέχει και τις κρυπτογραφημένες κεφαλίδες του αρχικού πακέτου.

Padding : Περιέχει επιπλέον bit τα οποία μπορεί να χρειάζονται όταν το αποτέλεσμα της κρυπτογράφησης είναι πολλαπλάσιο κάποιου αριθμού ή για να εξασφαλίσει ότι τα κρυπτογραφημένα δεδομένα θα τερματίζουν στο όριο των 2 byte έτσι ώστε να μένει χώρος για τα επόμενα 2 πεδία τα οποία πρέπει να είναι ευθυγραμμισμένα σε πολλαπλάσιο των 32 bit

Pad Length : Το πεδίο αυτό ορίζει το πόσα byte χρησιμοποιήθηκαν σαν padding για το κρυπτογραφημένο πακέτο

Next Header : Το πεδίο αυτό προσδιορίζει τον τύπο των δεδομένων τα οποία περιέχονται μέσα στο πεδίο Payload Data.

Integrity Check Value (ICV) ή Authentication Data: Η τιμή ελέγχου ακεραιότητας είναι μια τιμή η οποία υπολογίζεται πάνω στην κεφαλίδα του ESP πακέτου , του αρχικού πακέτου και των πεδίων που ακολουθούν το κρυπτογραφημένο πακέτο(Padding, Pad Length, Next Header). Το πεδίο αυτό σε αντιστοιχία με το όμοιο πεδίο του AH χρησιμοποιεί τους αλγόριθμους : HMAC-SHA1-96 [RFC2404], AES-GMAC with AES-128 [RFC4543], AES-XCBC-MAC-96 [RFC3566].

1.7.1.3 Διαχείριση κλειδιού (Key Management)

Το IPSec χρησιμοποιεί το Internet Key Exchange(IKE) πρωτόκολλο για να διευκολύνει και να αυτοματοποιήσει τις Συσχετίσεις Ασφαλείας (SA) και την ανταλλαγή κλειδιών μεταξύ του αποστολέα και του παραλήπτη οι οποίοι μεταφέρουν δεδομένα. Χρησιμοποιώντας κλειδιά διασφαλίζεται οτι ο παραλήπτης και ο αποστολέας μπορούν να έχουν πρόσβαση στο μύνημα.

Ένα απ τα χαρακτηριστικά που το κάνουν ασφαλές είναι με ποίο τρόπο το κλειδί αυτό θα ανανεώνεται έτσι ώστε αυτόι που επικοινωνούν να επικοινωνούν μεταξύ τους με ασφάλεια. Για να γίνει αυτό υπάρχουν δυο τρόποι ανανέωσης των κλειδιών. Ο αυτόματος και ο χειροκίνητος τρόπος.

- Χειροκίνητη διαχείριση κλειδιού: Τα κρυφά κλειδιά και οι συσχετισμοί ασφαλείας ρυθμίζονται χειροκίνητα και στους δυο χρήστες οι οποίοι πρόκειται να επικοινωνίσουν πρίν ξεκινήσει η επικοινωνία μεταξύ τους. Μόνο ο αποστολέας και ο παραλήπτης γνωρίζουν το κλειδί . Εάν τα δεδομένα αυθεντικοποίησης είναι έγκυρα, ο παραλήπτης γνωρίζει οτι η επικοινωνία του με τον αποστολέα δεν έχει τροποποιηθεί. Η υλοποίηση αυτή είναι εύχρηστη για μικρές και στατικές επιχειρίσεις . Όλα τα κλειδιά θα πρέπει να έχουν διανεμηθεί εκ των προτέρων . Εαν για κάποιο λόγο παραβιαστούν αυτά τα

κλειδιά κάποιος άλλος χρήστης μπορεί να προσποιηθεί τον νόμιμο χρήστη και να συνδεθεί στο VPN.

- Αυτόματη διαχείριση κλειδιού: Η αυτόματη διαχείριση κλειδιών γίνεται με βάση το πρωτόκολλο IKE . Αυτός ο τρόπος διαχείρισης είναι χρήσιμος για μεγάλης έκτασης χρήσης του VPN. Το IKEv2 το οποίο εκδόθηκε το 2005 διατηρεί τις περισσότερες λειτουργίες του IKEv1 και επίσης υποστηρίζει διάσχιση NAT (Network Address Translation) και επίσης παρέχει μεγαλύτερη ευελιξία. Το IKE επίσης υποστηρίζει την χρήση ψηφιακών πιστοποιητικών. Οι χρήστες αυθεντικοποιούνται υπογράφοντας τα δεδομένα με την χρήση του ψηφιακού τους κλειδιού. Ο τελικός αποδέκτης επιβεβαιώνει την υπογραφή. Το IKE δημιουργεί τότε ένα αυθεντικοποιημένο ασφαλές τούνελ μεταξύ δύο άκρων και διαπραγματεύεται τις συσχετίσεις ασφάλειας μεταξύ των δύο αυτών και ανταλλάσει τα κλειδιά. Οι συσχετισμοί ασφαλείας είναι μια ομάδα παραμέτρων η οποία περιλάμβάνει μεθόδους,κλειδιά αλγορίθμους κ.α. Επίσης παρακολουθεί τα κλειδιά και τα ενημερώνει μεταξύ των σταθμών που επικοινωνούν. Το IKE χρησιμοποιεί πρωτόκολλα όπως το ISAKMP(The Internet Association and Key Management Protocol) και το Oakley για να καθορίσει τις διαδικασίες για παραγωγή κλειδιού ,δημιουργία και διαχείριση των συσχετισμών ασφαλείας και της αυθεντικοποίησης.(Cisco sys,2000)

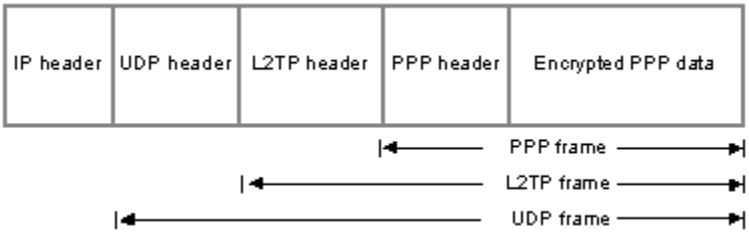
1.7.2 L2TP (Layer 2 Tunneling Protocol)

Το πρωτόκολλο L2TP(RFC 3931) είναι ένα πρωτόκολλο δημιουργίας "τούνελ" που όπως λέει και το όνομά του λειτουργεί πάνω στο 2ο επίπεδο του προτύπου OSI.

Αποτελεί συνδυασμό του PPTP και το L2F της Cisco. Μπορεί να χρησιμοποιηθεί ως πρωτόκολλο δημιουργίας τούνελ για να ενσωματώνει PPP δομές οι οποίες μεταφέρονται μέσω του ιντερνετ. Πολλαπλές συνδέσεις επιτρέπονται μεταξύ ενός τούνελ. Τα πρωτόκολλα που βρίσκονται στο επίπεδο 2 του VPN ενσωματώνουν όπως αναφέρθηκε PPP δομές και είναι ικανά να μεταδώσουν non-IP πρωτόκολλα(οπως το AppleTalk) μέσα απο ένα IP δίκτυο.

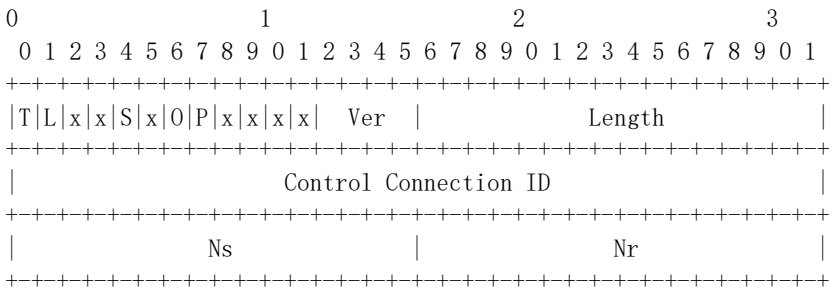
Λόγω έλλειψης κρυπτογράφισης το L2TP χρησιμοποιείται μαζί με το IPSec. Όταν το L2TP εκτελείται πάνω στο IPSec οι υπηρεσίες ασφαλειάς παρέχονται απο το IPSec, δηλαδή AH και το ESP. Όλοι οι έλεγχοι και τα δεδομένα του L2TP εμφανίζονται ως ομογενοποιημένα IP πακέτα δεδομένων στο IPSec σύστημα.

Για να λειτουργήσει το L2TP χρειάζεται ένας LNS (L2TP Network Server) ο οποίος θα παρέχει υπηρεσίες L2TP στο δίκτυο. Αυτός είναι υπεύθυνος για την δημιουργία του τούνελ μεταξύ δύο σημείων.



1.14 L2TP πακέτο(HKSAR,2008)

Η κεφαλίδα ενός L2TP πρωτοκόλου έχει την ακόλουθη μορφή(IETF tools,2017)



Όπου το κάθε πεδίο σημαίνει :

T : τύπος μηνύματος. 1 για μήνυμα ελέγχου

L : Στα μηνύματα ελέγχου είναι πάντα 1 γιατί το πεδίο Length υπάρχει πάντα

S : Το πεδίο αυτό προσδιορίζει αν τα πεδία Ns Nr υπάρχουν στην κεφαλίδα. Είναι πάντα 1

Ver : η έκδοση του πρωτοκόλλου για να γνωρίζουν οι κόμβοι LNS αν υποστηρίζεται από αυτούς

Length : Το μήκος του πακέτου από ξεκινώντας από το bit T.

CC ID (Control Connection Id) : Προσδιορίζει για ποιο τούνελ είναι αυτό το μήνυμα ελέγχου

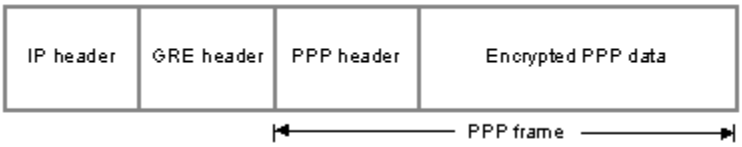
Ns : Ο αύξων αριθμός αυτού του μηνύματος ελέγχου

Nr : Ο αύξων αριθμός που περιμένει να έχει το επόμενο μήνυμα ελέγχου που θα λάβει.

1.7.3 Point-to-Point Tunneling Protocol (PPTP)

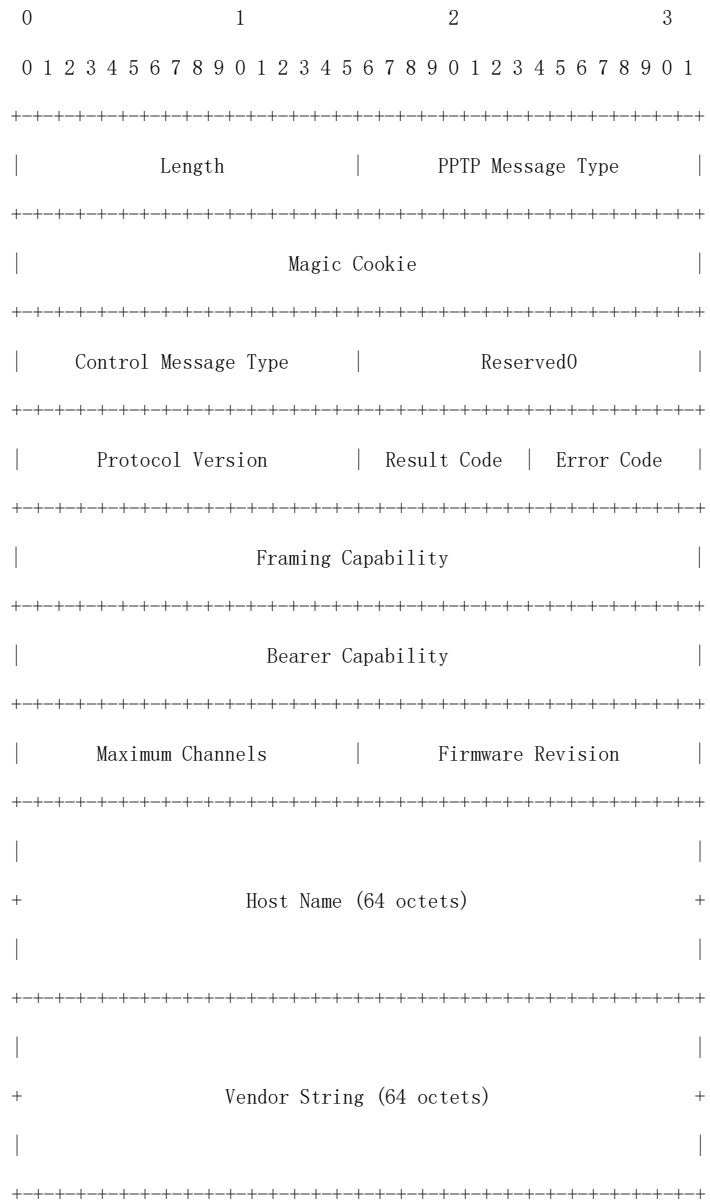
Το PPTP(RFC 2637) είναι ένα από τα πιο ευρέως διαδεδομένα πρωτόκολλα καθώς ήταν και ένα από τα πρώτα τα οποία υλοποιήθηκαν. Χρησιμοποιούνται κυρίως από χρήστες οι οποίοι τρέχουν λειτουργικό σύστημα Windows. Το PPTP είναι υλοποιημένο πάνω στο πρωτόκολλο PPP(Point-to-Point). Είναι ένα multi-protocol ,dial-up protocol, (Η Dial-up είναι μια μορφή πρόσβασης στο διαδίκτυο που χρησιμοποιεί λειτουργίες του δημόσιου τηλεφωνικού δικτύου μεταγωγής (PSTN) για να πραγματοποιήσει σύνδεση σε έναν πάροχο υπηρεσιών διαδικτύου (ISP) καλώντας έναν τηλεφωνικό αριθμό σε μια συμβατική γραμμή τηλεφώνου) που χρησιμοποιείται για να συνδεθεί στο ίντερνετ. Το PPTP συνδέεται στο επιθυμητό δίκτυο δημιουργώντας ένα εικονικό δίκτυο για κάθε απομακρυσμένο χρήστη. Οι μηχανισμοί ασφαλείας που χρησιμοποιούνται στις PPP συνδέσεις υποστηρίζονται και στο PPTP VPN . Η μεταφορά δεδομένων μέσω τούνελ για το PPTP επιτυγχάνεται μέσα απο πολλά επίπεδα ενσωμάτωσης των πακέτων. Το PPTP ενσωματώνει PPP δομές ως δεδομένα μετάδοσης μέσα απο τούνελ μέσω του δικτύου IP όπως το ιντερνετ ή ένα ιδιωτικό intranet χρησιμοποιώντας μια τροποποιημένη έκδοση του GRE. Το GRE παρέχει έλεγχο ροής και εκτόνωση στα μεταδιδόμενα PPP πακέτα. Τότε τα δεδομένα αυτά τα οποία έχουν ενσωματώσει τα πρωτόκολλα GRE-PPP ενσωματόνουν και μια κεφαλίδα IP η οποία περιέχει την διεύθυνση της πηγής και του προορισμού για τον PPTP πελάτη και εξυπηρετητή. Μόλις ο εξυπηρετητής λάβει τα δεδομένα τα επεξεργάζεται και αφαιρεί την IP,GRE και PPP κεφαλίδα και αποκρυπτογραφεί(και/ή αποσυμπιέζει) τα δεδομένα του PPP.

Η μορφή την οποία έχει το πακέτο καθώς περνάει από ένα PPTP τούνελ είναι



1.15 PPP Frame (HKSAR,2008)

Τα μηνύματα ελέγχου του είναι της μορφής(IETF,2017) :

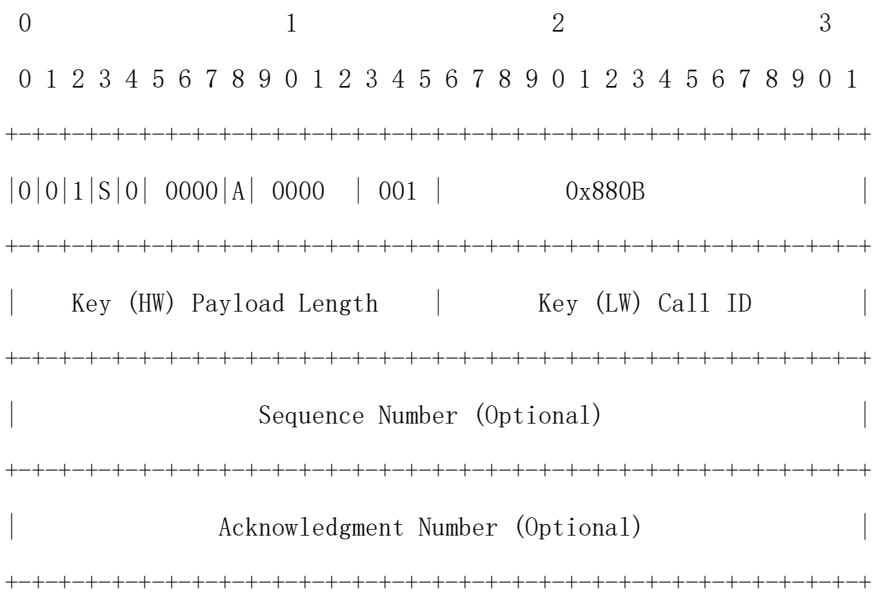


Όπου αντίστοιχα το Length είναι το μέγεθος το PPTP Message type προσδιορίζει τον τύπο του μηνύματος το Magic Cookie είναι το 0x1A2B3C4D και χρησιμοποιείται για τον έλεγχο

εγκυρότητας του λαμβανόμενου μηνύματος , το Protocol Version η έκδοση του πρωτοκόλλου που ο αποστολέας επιθυμεί να στείλει, Result Code και Error Code κωδικοί οι οποίοι προσδιορίζουν την επιτυχία ή αποτυχία της σύνδεσης και το είδος του σφάλματος , Framing Capabilities αν ο αποστολέας υποστηρίζει σύγχρονη ή ασύγχρονη αποστολή των μηνυμάτων, bearer Capability αν πρόκειται για αναλογική ή ψηφιακή σύνδεση, Maximum Channels το πόσες ταυτόχρονες συνδέσεις μπορεί να υποστηρίξει ο PNS (PPTP Network Server) , ο οποίος είναι ο διακομιστής της υπηρεσίας PPTP, το αντίστοιχο του LNS , Firmware Revision το ποια έκδοση λογισμικού λειτουργεί ο PNS, Host name την διεύθυνση του PNS, και Vendor Name τον τύπο του λογισμικού που τρέχει ο PNS για να υποστηρίξει την υπηρεσία.

Εκτός από την γενική μορφή των μηνυμάτων ελέγχου που παρουσιάστηκε παραπάνω το κάθε μήνυμα ελέγχου έχει και τις δικές του ιδιομορφίες και αυτό το κάνει ένα πρωτόκολλο ιδιαίτερα δύσκολο να υλοποιηθεί. Ταυτόχρονα με τις ελλείψεις ασφάλειας τις οποίες παρουσιάζει θεωρείται πλέον ένα παρωχημένο πρωτόκολλο για την δημιουργία VPN συνδέσεων.

Η μορφή που έχει η κεφαλίδα του GRE είναι:



Το πεδίο S καθορίζει αν το πακέτο έχει δεδομένα μέσα ή απλά είναι μια επιβεβαίωση λήψης. Αν το bit αυτό είναι 1 τότε το πακέτο περιέχει δεδομένα μέσα από την κεφαλίδα του GRE. Αντιθέτως εάν το bit A είναι ενεργοποιημένο σημαίνει ότι το πακέτο περιέχει επιβεβαίωσή λήψης κάποιου προηγούμενου και περιέχει τον αριθμό Sequence Number (Αριθμό Σειράς) του πακέτου που έλαβε στο πεδίο Acknowledge Number. Το Sequence Number μπαίνει μόνο όταν το bit S είναι 1 και προσδιορίζει τον μοναδικό αριθμό πακέτου.

1.7.4 Secure Sockets Layer (SSL) and Transport Layer Security (TLS)

Το SSL και το TLS(RFC 5246)είναι πρωτόκολλα τα οποία λειτουργούν σε υψηλότερο επίπεδο από τα προηγούμενα που έχουν αναφερθεί. Εν αντιθέσει με εκείνα το SSL/TLS λειτουργούν στο υψηλότερο επίπεδο, στο επίπεδο της μεταφοράς. Η δικτυακή πόρτα η οποία έχει εξ'ορισμού ανατεθεί στο πρωτόκολλο SSL/TLS είναι η πόρτα 443.

Υπάρχει ένα πλήθος κρυπτογραφικών χαρακτηριστικών που παρέχεται απο το SSL/TLS και περιλαμβάνει εμπιστευτικότητα,ακεραιότητα και ψηφιακές υπογραφές. Για αυτό το λόγο σε κάθε σύνδεση του χρήστη με τον διακομιστή πλέον των δεδομένων (π.χ κάποιο site) κατά την έναρξη της σύνδεσης συμφωνούνται 3 πράγματα:

- Το κλειδί το οποίο θα χρησιμοποιηθεί
- Ο τύπος την κρυπτογράφησης
- Και το κλειδί hash που θα βεβαιωθεί η ακεραιότητα των δεδομένων.

Το πρωτόκολλο που ακολουθείται ονομάζεται TLS Handshake Protocol:

1. Ο διακομιστής με τον πελάτη ανταλλάσουν μυνήματα για να επιβεβαιώσουν την επικοινωνία τους και ανταλλάσουν τυχαίες τιμές και ελέγχουν για συνέχιση της σύνδεσης.
2. Ο διακομιστής (server) στέλνει το πιστοποιητικό του στον χρήστη το οποίο περιλαμβάνει διάφορες πληροφορίες μεταξύ αυτών και το δημόσιο του κλειδί
3. Ο χρήστης στέλνει το δικό του κλειδί στον server για υπολογίσει τον κρυφό κωδικό με τον οποίο από εκείνη την στιγμή θα γίνεται η κωδικοποίηση.
4. Ο χρήστης ζητάει από τον server να κωδικοποιήσει τα δεδομένα και του συμφωνούν στα 3 χαρακτηριστικά τα οποία αναφέρθηκαν
5. Ο server στέλνει κωδικοποιημένο το μήνυμα ότι ξεκινάει η σύνδεση.
6. Η σύνδεση είναι πλέον ασφαλής και κωδικοποιημένη.(ietf,2017)

Το πρωτόκολλο SSL/TLS είναι από τα πλέον διαδεδομένα πρωτόκολλα γενικής χρήσης για την δημιουργία ασφαλών συνδέσεων στο δίκτυο. Οι περισσότερες υπηρεσίες όπως mail, ηλεκτρονικές συναλλαγές, υπηρεσίες μηνυμάτων και υπηρεσίες φωνής μέσω IP(VoIP) χρησιμοποιούν αυτά για την ασφάλεια και την προστασία των χρηστών.

1.7.5 Secure Shell (SSH)

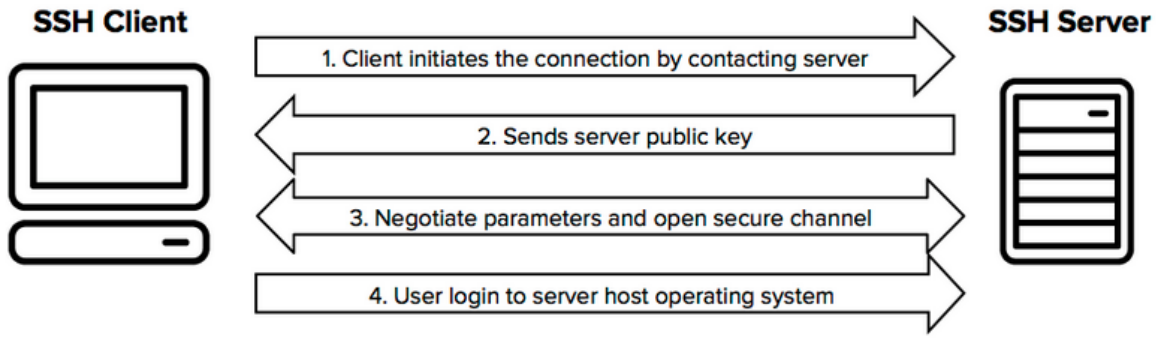
Το SSH είναι ένα πρωτόκολλο το οποίο υποστηρίζει την λειτουργία δικτυακών υπηρεσιών μέσα από μη ασφαλείς συνδέσεις. Παρέχει πολλές επιλογές για ισχυρή αυθεντικοποίηση. Του έχει ανατεθεί η δικτυακή πόρτα 22. Οι υπηρεσίες οι οποίες μπορούν να υλοποιηθούν πάνω από το πρωτόκολλο SSH είναι:

- Απομακρυσμένη σύνδεση σε υπολογιστή και έλεγχός του
- Μεταφορά αρχείων
- Για την προώθηση ή τουνελοποίηση δεδομένων μιας πόρτας μέσα από το δίκτυο (μόνο τοπικά)

Αποτελεί μια ασφαλή εναλλακτική έναντι μη-ασφαλών πρωτοκόλλων όπως το telnet και μη ασφαλείς μεταφορές αρχείων όπως το FTP.

Μόλις επιτευχθεί η σύνδεση μεταξύ του διακομιστή και του πελάτη, τα δεδομένα τα οποία μεταφέρονται κρυπτογραφούνται σύμφωνα με τους αλγορίθμους που έχουν συμφωνιθεί κατα την διαδικασία επίτευξης επικοινωνίας. Κατα την διαδικασία αυτή επίσης δημιουργείται και το κλειδί κρυπτογράφισης που θα χρησιμοποιηθεί. Η κίνηση μεταξύ των επικοινωνούντων προστατεύεται με χρήση ισχυρών κρυπτογραφικών αλγορίθμων όπως ο AES και επίσης περιλαμβάνει μηχανισμούς που εξασφαλίζει την ακεραιότητα των μεταδιδόμενων δεδομένων με την χρήση αλγορίθμων κατακερματισμού όπως ο SHA.

Η χρήση VPN τούνελ μέσω SSH δεν είναι και η πιο εύκολη λύση αλλά προτιμάται όταν θέλουμε τον απομακρυσμένο έλεγχο ενός υπολογιστή ο οποίος τρέχει λειτουργικό σύστημα LINUX.



1.16 SSH σύνδεση (ssh communication security, 2017)

1.7.6 OpenVPN

Το OpenVPN είναι ένα λογισμικό ανοιχτού κώδικα το οποίο γράφτηκε από αρχικά από τον James Yonan. Χρησιμοποιεί ένα πρωτόκολλο το οποίο βασίζεται στο SSL/TLS για την ανταλλαγή κλειδιού χρησιμοποιώντας την βιβλιοθήκη openSSL. Για την πιστοποίηση του χρήστη μπορεί να παραμετροποιηθεί ώστε να χρησιμοποιήσει είτε πιστοποιητικά ασφαλείας, είτε όνομα χρήστη και κωδικό, είτε κάποιο προεγκατεστημένο κλειδί, είτε τον συνδυασμό δυο ή παραπάνω από αυτά.

Μπορεί να υλοποιηθεί είτε στο επίπεδο 2 είτε στο επίπεδο 3 του προτύπου OSI, χρησιμοποιώντας τις συσκευές εικονικού δικτύου TUN(network TUNnel) και TAP (network tap).

Το TUN χρησιμοποιεί συσκευές οι οποίες προσομοιώνουν συσκευές στο επίπεδο δικτύου και το TAP προσομοιώνει συσκευές στο επίπεδο της ζεύξης» Η προσομοίωση αυτή συμβαίνει παίρνοντας όλα τα πακέτα τα οποία φεύγουν ή έρχονται προς τα εκείνη τη συσκευή απευθείας από την στοίβα δικτύου του λειτουργικού συστήματος.

Το OpenVPN μπορεί να χρησιμοποιήσει και αλγόριθμο συμπίεσης πάνω στα δεδομένα όπως ο LZO(Lempel–Ziv–Oberhumer).

Πλέον το OpenVPN υλοποιείται σε πολλούς δρομολογητές καθώς είναι ένα πολύ διαδεδομένο πρωτόκολλο λόγω της δυνατότητας μεγάλης παραμετροποίησης.

Το OpenVPN λόγω της δυνατότητας να δουλέψει τόσο με UDP (User Datagram Protocol) , όσο και με TCP (Transmission Control Protocol) , δουλεύει στην δικτυακή πόρτα 1194 και για τους δύο τρόπους μεταφοράς δεδομένων.


Κεφάλαιο 2
Προβλήματα Ασφάλειας
στα VPN δίκτυα

2 Προβλήματα Ασφάλειας στα VPN δίκτυα

Παρότι τα VPN αποτελούν απο τους πιο ασφαλείς τρόπους για να περιηγηθεί κάποιος στο διαδίκτυο και όπως αναφέρθηκε περιέχουν πολλά πρωτόκολλα που τα κάνουν ασφαλές παρέχοντας αυθεντικοποίηση , ακεραιότητα των δεδομένων και κρυπτογράφιση των δεδομένων. Όμως ότι διακινείται μέσω του δικτύου είτε του ιδιωτικού είτε του δημοσίου, αποτελεί στόχο για επίδοξους «χάκερ» που σκοπό έχουν να κλέψουν τα πακέτα που διακινούνται και να τα πουλήσουν σε συλλέκτες δεδομένων ή να τα εκθέσουν στο διαδίκτυο(wikileaks). Χάρις όμως στις επιθέσεις μπορούμε να πούμε ότι οι οργανισμοί και οι εταιρίες που αναπτύσουν τα πρωτόκολλα και τα δίκτυα ενισχύουν και ενημερώνουν κατάλληλα τα πρωτοκολλά για ασφαλέστερη σύνδεση στο διαδίκτυο.

2.1 Ευπάθειες των VPN δικτύων

Τα VPN δίκτυα είναι κατά παραδοχή ένας από τους ασφαλέστερους τρόπους για την σύνδεση σε ένα εταιρικό δίκτυο ή σε ένα απομακρυσμένο εξυπηρετητή. Παρόλα αυτά τα πρωτόκολλά του και ο τρόπος υλοποίησης τους παίζουν πολύ μεγάλη σημασία στην ασφάλεια ενός δικτύου VPN. Επίσης, τα δίκτυα αυτά απειλούνται απο επιθέσεις hacking όπως το VPN hijacking και man-in-the-middle (επίθεση ενδιαμέσου) κ.α.

- VPN hijacking: Είναι ένα είδος ευπάθειας και είναι η μη-εξουσιοδοτημένη κατάληψη και χρήση μιας VPN σύνδεσης ενός απομακρυσμένου χρήστη, και ο επιτιθέμενος παριστάνει τον εξουσιοδοτημένο χρήστη στο δίκτυο αυτό
- Man-in-the-middle: Αυτού του τύπου οι επιθέσεις επηρεάζουν την κυκλοφορία μεταξύ των ομάδων που επικοινωνούν και περιλαμβάνει διακοπή, εισαγωγή, διαγραφή και τροποποίηση των μυνημάτων, επιστροφή των μυνημάτων πίσω στον αποστολέα, επανάληψη παλιών μυνημάτων και ανακτεύθυνση τους
- Αυθεντικοποίηση χρήστη: Αν και όπως έχει αναφερθεί τα VPN είναι ο ασφαλέστερος τρόπος σύνδεσης , δεν διαθέτουν ισχυρούς μηχανισμούς αυθεντικοποίησης χρήστη και γι αυτό πρέπει να είναι όσο το δυνατόν πιο ισχυροί για να αποτρέψει μια τυχόν μη-εξουσιοδοτημένη πρόσβαση στους πόρους του δικτύου.
- Ιοί και κακόβουλο λογισμικο: Αποτελεί κίνδυνο απο την μεριά του χρήστη ο οποίος μπορεί να έχει μολυνθεί με κάποιον ιο ή λογισικό και οι κωδικοί πρόσβασης στο VPN να έχουν παραβιαστεί. Αν μιλάμε για εταιρικό δίκτυο είτε intranet ή extranet μπορεί απο το ένα μολυσμένο τμήμα του δικτύου να εξαπλωθεί σε όλο το δίκτυο αν το anti-virus δεν είναι σε θέση να αντιμετωπίσει τον ιό/λογισμικό.
- Μη εξουσιοδοτημένη πρόσβαση: Κάποιοι χρήστες ή και δίκτυα ακόμα μπορέι να έχουν δικαιώματα πρόσβασης που πιθανόν να μην χρειάζονται.
- Προσαρμοστικότητα: Η προσαρμοστικότητα αποτελεί και αυτή ένα σημαντικό ζήτημα. Λόγου χάρι , λογισμικό συμβατό με IPsec απο δύο διαφορετικούς παρόχους υπάρχει πιθανότητα να μην μπορεί να δουλέψει σε συνεργασία με το άλλο. (HKSAR,2008)

2.2 Αυθεντικοποίηση Χρήστη

Εξ' ορισμού το VPN δεν παρέχει / εξοπλίζεται με ισχυρή αυθεντικοποίηση χρήστη. Μια VPN σύνδεση θα πρέπει να πραγματοποιείται μόνο απο έναν αυθεντικοποιημένο χρήστη. Αν η αυθεντικοποίηση δεν είναι αρκετά δυνατή για να περιορίσει την μη-εξουσιοδοτημένη πρόσβαση , μια μη-εξουσιοδοτούμενη ομάδα μπορεί να συνδεθεί στο δίκτυο και στους πόρους του. Οι πιο πολλές VPN εφαρμογές παρέχουν περιορισμένους μηχανισμούς αυθεντικοποίησης. Για παράδειγμα , το PAP που χρησιμοποιείται στο PPTP , μεταφέρει και το όνομα χρήστη και τον κωδικό ως κείμενο. Ένα τρίτο πρόσωπο μπορεί να συλλάβει αυτή την πληροφορία και να την χρησιμοποιήσει ώστε να αποκτήσει πρόσβαση στο δίκτυο.

Για την υλοποίηση της αυθεντικοποίησης χρήστη μπορούν να χρησιμοποιηθούν οι ακόλουθες μέθοδοι:

2.2.1 PAP (Password Authentication Protocol)

Το PAP είναι ένα πρωτόκολλο αυθεντικοποίησης του χρήστη το οποίο χρησιμοποιείται σε PPP(point to point) VPN συνδέσεις. Θεωρείται ένας πολύ ασθενής μηχανισμός αυθεντικοποίησης γιατί στέλνει το όνομα χρήστη αλλά και τον κωδικό του μέσα στο ίδιο πακέτο χωρίς να έχει κάποια κωδικοποίηση. Αυτό το κάνει πολύ ανασφαλές σε επιθέσεις επανάληψης ή remote client impersonation. (Microsoft technet,2001)

Η διαδικασία που ακολουθεί είναι :

Ο client στέλνει μέσα σε ένα πακέτο τις πληροφορίες του, όνομα χρήστη και κωδικό και ο διακομιστής απαντάει με ένα μήνυμα επιτυχίας ή αποτυχίας. Τα μηνύματα είναι της μορφής

Περιγραφή	1 byte	1 byte	2 byte	1byte	Κυμαινόμενο	1 byte	Κυμαινόμενο
Αίτηση Αυθεντικοποίησης	Κωδικός = 1	ID	Μήκος	Μήκος Ονόματος Χρήστη	Όνομα Χρήστη	Μήκος κωδικού	Κωδικός
Επιβεβαίωση Αυθεντικοποίησης	Κωδικός = 2	ID	Μήκος	Μήκος Μηνύματος	Μήνυμα		
Άρνηση Αυθεντικοποίησης	Κωδικός = 3	ID	Μήκος	Μήκος Μηνύματος	Μήνυμα		

Το PAP πλέον υλοποιείται όταν(Techopedia,2017):

a) Οι διακομιστές δεν υποστηρίζουν το πρωτόκολλο CHAP(Challenge-Handshake Authentication Protocol)

b) Όταν το πρωτόκολλο CHAP που έχει υλοποιηθεί δεν είναι συμβατό και στις δύο πλευρές

c) Σε συγκεκριμένες περιπτώσεις τις οποίες θα πρέπει να υπάρχει διαθέσιμος ο κωδικός σε μορφή κειμένου

2.2.2 CHAP (Challenge-Handshake Authentication Protocol)

Το CHAP(rfc 1994) αποτελεί μια άλλη μορφή αυθεντικοποίησης του χρήστη. Το πλεονέκτημά του είναι ότι το μυστικό για την αυθεντικοποίηση του χρήστη δεν μεταφέρεται μέσα από το δίκτυο αλλά το έχουν και ο χρήστης και ο διακομιστής. Το CHAP χρησιμοποιείται από PPP διακομιστές για να επαληθεύσει την ταυτότητα των χρηστών.

Το CHAP υλοποιείται με μια τριπλή ανταλλαγή μηνυμάτων μεταξύ του διακομιστή και του χρήστη. Η ανταλλαγή αυτή των μηνυμάτων ακολουθεί την ακόλουθη σειρά:

i. Μετά την φάση της πραγματοποίησης της ζεύξης ο διακομιστής στέλνει ένα μήνυμα "πρόκλησης" προς τον χρήστη.

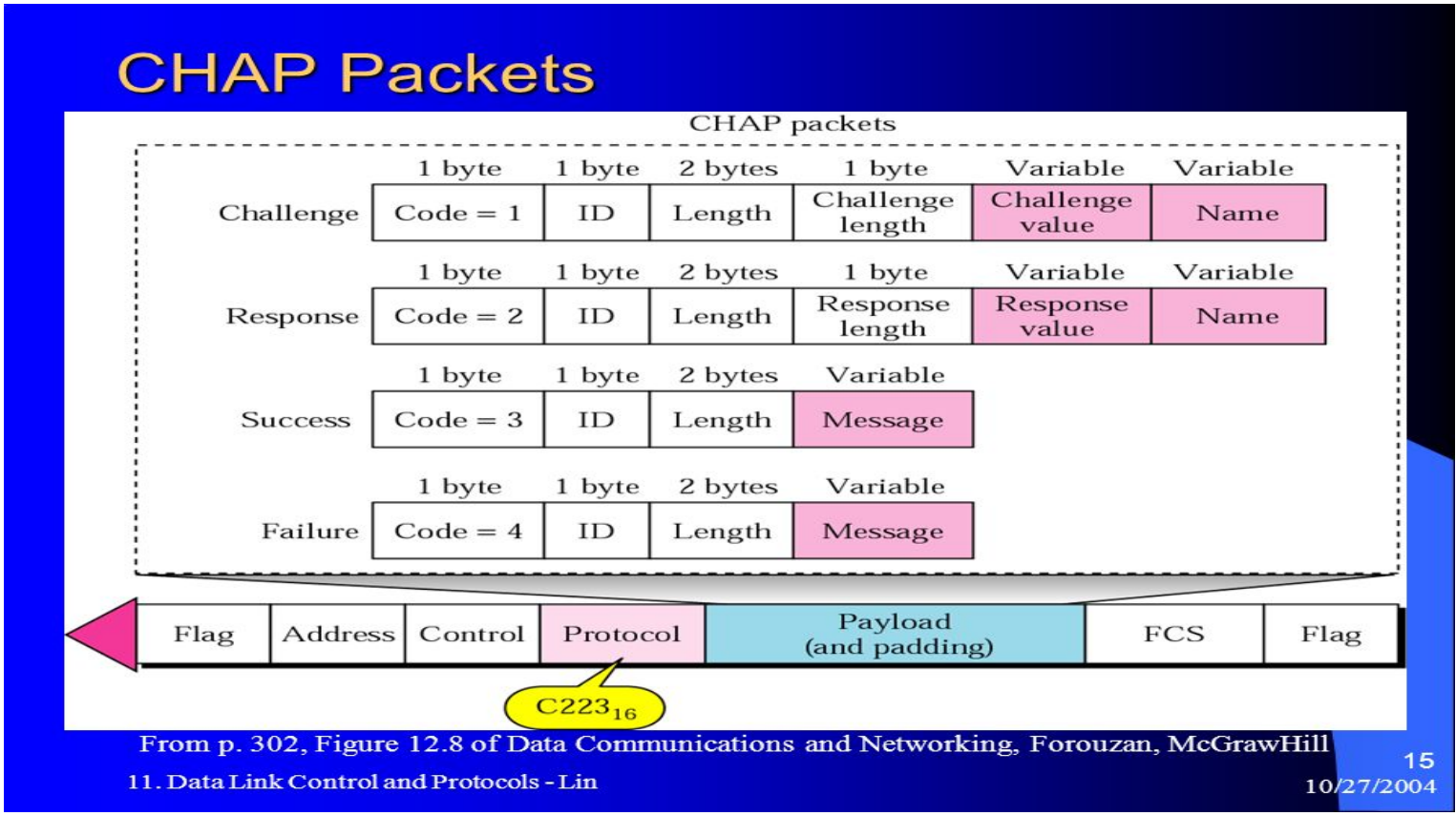
ii. Ο χρήστης απαντάει με μια τιμή η οποία έχει υπολογιστεί χρησιμοποιώντας μια συνάρτηση κατακερματισμού μονής κατεύθυνσης.

iii. Ο διακομιστής λαμβάνει αυτή την τιμή και την ελέγχει με τον δικό του υπολογισμό της τιμής κατακερματισμού. Αν η τιμή είναι ίδια τότε η αυθεντικοποίηση πραγματοποιείται. Αλλιώς θα πρέπει να γίνει τερματισμός της ζεύξης.

Περιστασιακά η διαδικασία επαναλαμβάνεται για την επαλήθευση και την συνέχιση της σύνδεσης του χρήστη.

Για το CHAP με βάση το RFC το οποίο το διέπει μόνο μια συνάρτηση κατακερματισμού ορίζεται, αυτή του MD5. Η συνάρτηση αυτή συναποφασίζεται κατά την διάρκεια της επίτευξης της ζεύξης.

Τα πακέτα του CHAP έχουν την μορφή :



2.1 Πακέτα CHAP (Gilbert C, 2016)

2.2.3 MS-CHAP (Microsoft CHAP)

Το MS-CHAP (rfc 2433) είναι η εκδοχή της Microsoft για το πρωτόκολλο CHAP. Το MS-CHAP ακολουθεί το πρωτόκολλο CHAP και ορίζεται για PPP συνδέσεις.

Το πρωτόκολλο αυτό φτιάχτηκε για την σύνδεση απομακρυσμένων σταθμών εργασίας που τρέχουν λειτουργικό σύστημα Windows. Υλοποιήθηκε με την σκοπιμότητα να υλοποιήσει μηχανισμούς σαν αυτούς που έχουν οι τοπικοί χρήστες του δικτύου ενώ ταυτόχρονα να του επιτρέψει την χρήση κρυπτογράφησης και αλγορίθμων κατακερματισμού.

Το MS-CHAP υλοποιείται κατά το βήμα 3 της επίτευξης της ζεύξης το οποίο αντί στην θέση του αλγορίθμου να έχει την τιμή 5 η οποία ορίζεται για το απλό CHAP έχει την τιμή 0x80.

Άλλη μια διαφορά μεταξύ των δύο πρωτοκόλλων είναι ότι η απάντηση στο μήνυμα (κωδικός 2) είναι σε μια μορφή η οποία έχει σχεδιαστεί για να είναι συμβατή με τα προϊόντα δικτύων της εταιρίας. Επιπλέον το MS-CHAP δεν χρειάζεται να έχει αποθηκευμένο τον κωδικό ούτε σε απλή μορφή , ούτε κωδικοποιημένο σε μορφή η οποία θα μπορεί να αντιστραφεί. Έχει ακόμα μηχανισμούς οι οποίοι ελέγχονται από τον διακομιστή για επαναπροσπάθεια και αλλαγή κωδικού.

Τα πακέτα αλλαγής κωδικού έχουν στο πεδίο του κωδικού την τιμή 5 περιέχουν σε κωδικοποιημένη μορφή τον παλιό και τον νέο κωδικό που χρησιμοποιείται. Υπάρχουν 2 εκδόσεις

του μηνύματος αλλαγής κωδικού στην νεότερη από τις οποίες υπάρχει ταυτόχρονα και αλλαγή της συνάρτησης κατακερματισμού που χρησιμοποιείται. Το ποια έκδοση από τις δύο θα χρησιμοποιηθεί καθορίζεται από τον κωδικό λάθους στο μήνυμα αποτυχίας. Αν υποστηρίζεται η έκδοση 2 τότε στο πεδίο του μηνύματος θα υπάρχει η τιμή 2.

Σαν αλγόριθμοι κατακερματισμού χρησιμοποιούνται ο Windows NT hash(συνάρτηση βασισμένη στον MD4) και ο LM hash και για την κρυπτογράφηση των δεδομένων χρησιμοποιείται ο αλγόριθμος DES(Data Encryption Standard). Το DES χρησιμοποιεί ένα κλειδί μήκους 56 bit και 8 bit για έλεγχο και με αυτά κωδικοποιεί τα δεδομένα σε μπλοκ των 64 bit. Ο κατακερματισμός των δεδομένων που χρειάζονται για την αυθεντικοποίηση τροφοδοτούνται στην συνάρτηση κρυπτογράφησης και αυτό είναι το τελικό αποτέλεσμα το οποίο στέλνεται στον διακομιστή για την επιβεβαίωση της αυθεντικοποίησης.

2.2.4 MS CHAP v2

Το MS-CHAPv2(rfc 2759) είναι η δεύτερη έκδοση του πρωτοκόλλου το οποίο υλοποίησε η Microsoft. Ο κωδικός που ανταλλάσσεται κατά την διάρκεια της επίτευξης της ζεύξης έχει την τιμή 0x81 και υλοποιήθηκε το 2000.

Οι διαφορές μεταξύ των δύο εκδόσεων είναι ότι είναι ότι εκτός από τον χρήστη ο οποίος επαληθεύεται απέναντι στον διακομιστή και ο διακομιστής επαληθεύει την γνησιότητα του απέναντι στον χρήστη. Επιπλέον το κάθε κρυπτογραφικό κλειδί σε αυτή την έκδοση υπολογίζεται με βάση και με μια τυχαία συμβολοσειρά. Κάθε φορά που πραγματοποιείται η αυθεντικοποίηση του χρήστη τότε αυτή η συμβολοσειρά αλλάζει.

Η διαδικασία που ακολουθείται είναι:

i. Ο διακομιστής στέλνει το μήνυμα πρόκλησης προς τον χρήστη.

ii. Ο χρήστης απαντάει με την απάντησή του και με μια πρόκληση προς τον διακομιστή

iii. Ο διακομιστής απαντάει αν είναι επιτυχής η απάντηση του χρήστη με την δική του απάντηση

iv. Αν η απάντηση που έδωσε ο διακομιστής είναι επιτυχής τότε η αυθεντικοποίηση επιτυγχάνεται.

Και οι δύο τρόποι αυθεντικοποίησης για την κρυπτογράφηση των δεδομένων χρησιμοποιούν τον αλγόριθμο DES (Data Encryption Standard) και σαν συνάρτηση κατακερματισμού την Windows NT Hash. (Scheiner B, 1999)

2.2.5 EAP(Extensible Authentication Protocol)

Το EAP(rfc 3748,5247) είναι ένα πλαίσιο (framework) το οποίο χρησιμοποιείται ευρέως τόσο σε ασύρματες ζεύξεις όσο και σε PPP ζεύξεις.Ουσιαστικά παρέχει ένα μηχανισμό επιπλέον ασφαλείας για ήδη υπάρχοντες τρόπους αυθεντικοποίησης χρήστη. Η διαδικασία η οποία ορίζει το EAP είναι η εξής:

i. Ο διακομιστής στέλνει μια αίτηση (Request) για να αυθεντικοποιήσει τον χρήστη. Η αίτηση έχει ένα πεδίο τύπου το οποίο καθορίζει το τι ζητάει από τον χρήστη. Για παράδειγμα στην αίτηση μπορεί να ζητήσει :

a) Ταυτοποίηση

b) Ειδοποίηση

c) Μόνο απάντηση

d) MD5-Πρόκληση

e) OTP (One time Password)

f) GTC (Generic Token Card)

g) AKA (Authentication and Key Agreement)

h) SIM (Subscriber Identity Modules)

i) PSK (Pre-Shared Key)

j) SAKE (Shared-secret authentication and key exchange) και

k) TLS (Transport Layer Security)

Αρχικά θα ζητηθεί να γίνει ταυτοποίηση η οποία μπορεί να παραληφθεί αν μπορεί να γίνει ταυτοποίηση χρήστη με κάποιο άλλο τρόπο.

ii. Ο χρήστης στέλνει μια απάντηση (Response) στον διακομιστή δίνοντας τα στοιχεία τα οποία ζήτησε. Όπως και στην αίτηση το πακέτο που περιέχει την απάντηση περιέχει και τον τύπο της αίτησης στην οποία απαντάει.

iii. Ο διακομιστής στέλνει μια καινούρια αίτηση με κάποιο άλλο τύπο και ο χρήστης απαντάει με την δική του απάντηση. Αυτό συνεχίζει όσο θεωρεί ο διακομιστής ότι χρειάζεται για να επαληθεύσει τον χρήστη. Κάθε βήμα αποτελεί κλείδωμα στην επικοινωνία μεταξύ τους εάν ο διακομιστής δεν λάβει απάντηση σε αυτό που αιτήθηκε.

iv. Η ανταλλαγή μηνυμάτων συνεχίζεται μέχρι ο διακομιστής να μην μπορεί να επαληθεύσει τον χρήστη ή να γίνει επιτυχής επαλήθευση όλων των βημάτων που απαιτούνται για την αυθεντικοποίηση του χρήστη. Και στις δύο περιπτώσεις ο διακομιστής απαντάει με το κατάλληλο μήνυμα είτε επιτυχίας είτε αποτυχίας.

Οι τύποι πακέτων οι οποίοι αποτελούν το EAP είναι αυτοί με κωδικούς :

1 -> Αίτηση

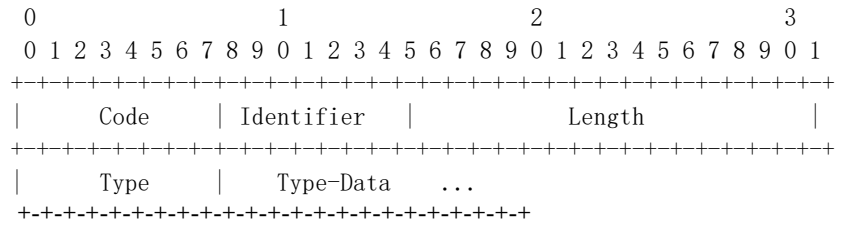
2 -> Απάντηση

3 -> Επιτυχία

4 -> Αποτυχία

Το EAP παρέχει υποστήριξη για αύξοντες αριθμούς πακέτων το οποίο αποτελεί μια επιπλέον ασφάλεια για τα δεδομένα και τον διακομιστή αλλά και τον χρήστη.

Η μορφή που έχουν τα πακέτα αίτησης(1) και απάντησης(2) του EAP είναι :



Όπου το πεδίο Identifier πρέπει να είναι ίδιο στην απάντηση όπως είναι στην αίτηση. Αν ο διακομιστής δεν περιμένει απάντηση με την τιμή που περιέχει σε αυτό το πεδίο τότε πρέπει να την αγνοεί. Επιπλέον κάθε καινούρια αίτηση πρέπει να έχει διαφορετική τιμή από την προηγούμενη. Το πεδίο Length περιέχει το μήκος όλου του πακέτου και το πεδίο Type περιέχει μία από τις τιμές που αναφέραμε πιο πριν. Τα δεδομένα που ακολουθούν αλλάζουν ανάλογα με τον τύπο της αίτησης και ορίζονται το κάθε ένα στο δικό του RFC.

Τα πακέτα επιτυχίας ή αποτυχίας περιλαμβάνουν μόνο τα 32 πρώτα bit της μορφής την οποία έχουν και τα υπόλοιπα πακέτα του πρωτοκόλλου.

Το EAP επιπλέον επειδή πολλές φορές μπορεί να στηρίζεται στην είσοδο του χρήστη υποστηρίζει μηχανισμούς επανάληψης της μετάδοσης. Συνήθως οι επαναμεταδόσεις μηνυμάτων οι οποίες επιτρέπονται είναι από 3-5. Επιπλέον αν το μέσο μετάδοσης είναι αξιόπιστο τότε θα έπρεπε να τεθεί ο αριθμός επαναμεταδόσεων στο άπειρο αλλά ο χρήστης θα πρέπει να κρατάει ένα χρόνο μέσα στον οποίο μπορεί να περιμένει για την αίτηση έτσι ώστε να μην μένει κλειδωμένος επ'άπειρο.

Για την επιπλέον κρυπτογράφηση των δεδομένων υλοποιήθηκε το πρωτόκολλο PEAP (Protected EAP) το οποίο ενσωματώνει τα πακέτα του EAP μέσα σε ένα ασφαλές και κρυπτογραφημένο τούνελ. Το PEAP περιγράφεται μόνο ως πρόχειρο στο draft-josefsson-pppext-eap-tls-eap-10

2.2.8 TLS(Transport Layer Security)

Το TLS(rfc 5246) αποτελεί ένα πρωτόκολλο το οποίο λειτουργεί στο επίπεδο εφαρμογής του προτύπου OSI. Όπως αναφέραμε αποτελεί διάδοχο του SSL. Για την εκκίνηση της χρήσης του πρωτοκόλλου ο χρήστης πρέπει να αυθεντικοποιηθεί από τον διακομιστή.

Κατά την έναρξη της επικοινωνίας συμφωνούν για την έκδοση του πρωτοκόλλου και τους αλγόριθμους κρυπτογράφησης που θα χρησιμοποιηθούν. Η διαδικασία αυθεντικοποίησης είναι προαιρετική και χρησιμοποιεί κρυπτογραφικές μεθόδους δημοσίου κλειδιού για να παραχθούν οι κοινοί μυστικοί κωδικοί.

Η διαδικασία αυθεντικοποίησης ξεκινάει μετά το πέρας της επικοινωνίας του χρήστη και του διακομιστή για την συμφωνία των αλγορίθμων, την ανταλλαγή τυχαίων αριθμών και του ελέγχου εάν πρόκειται για συνέχεια προηγούμενης σύνδεσης, καθώς και της ανταλλαγής των απαιτούμενων κρυπτογραφικών παραμέτρων και της συμφωνίας σε ένα κλειδί.

Για την αυθεντικοποίηση του χρήστη γίνεται ανταλλαγή πιστοποιητικών αλλά και κρυπτογραφικών παραμέτρων και η αυθεντικοποίηση είναι αμφίδρομη. Η διαδικασία που ακολουθείται είναι:

i. Ο χρήστης στέλνει το μήνυμα ClientHello το οποίο περιλαμβάνει έναν τυχαίο αριθμό ο οποίο αποτελείται από την τρέχουσα ώρα και έναν 28 bit τυχαίο αριθμό, μαζί με την έκδοση του πρωτοκόλλου που υποστηρίζει, τον αλγόριθμο κρυπτογράφησης, την μέθοδο συμπίεσης και τυχόν επιπλέον παραμέτρους οι οποίες υποστηρίζονται.

ii. Ο διακομιστής απαντάει σε αυτό το μήνυμα με τα στοιχεία τα οποία υποστηρίζει και συμφωνεί με βάση αυτά που έστειλε ο χρήστης. Εάν πρόκειται για μια σύνδεση η οποία συνεχίζεται στέλνει και τον αριθμό σύνδεσης.

iii. Ο διακομιστής στέλνει το πρωτόκολλό του στον χρήστη

iv. Ο διακομιστής κατόπιν μπορεί να στείλει ένα μήνυμα το οποίο περιέχει κάποιο κλειδί εάν οι πληροφορίες του πιστοποιητικού δεν είναι επαρκείς για να μπορέσει ο χρήστης να προχωρήσει στην ανταλλαγή ενός κλειδιού.

v. Ο διακομιστής στέλνει ένα αίτημα πιστοποιητικού προς τον χρήστη ζητώντας του το πιστοποιητικό του εάν δεν πρόκειται για ανώνυμη σύνδεση.

vi. Ο διακομιστής στέλνει το μήνυμα ServerHelloDone το οποίο ενημερώνει τον χρήστη ότι τελείωσε από την δική του πλευρά την διαδικασία αυθεντικοποίησης και ανταλλαγής μηνυμάτων.

vii. Ο χρήστης απαντάει με το πιστοποιητικό του εάν αυτό έχει ζητηθεί από τον διακομιστή. Εάν δεν υπάρχει κάποιο πιστοποιητικό διαθέσιμο στέλνει ένα μήνυμα το οποίο δεν περιλαμβάνει κάποιο πιστοποιητικό.

viii. Ο χρήστης στέλνει και συμφωνεί πάνω στο κλειδί που έστειλε ο διακομιστής είτε με απευθείας μετάδοσή του, είτε μέσω των παραμέτρων του αλγορίθμου Diffie Hellman το οποίο θα επιτρέψει την συμφωνία των δυο πλευρών πάνω στο ίδιο κλειδί.

ix. Προτελευταίο μήνυμα που ανταλλάσσουν οι δυο πλευρές είναι του τύπου Change Cipher στο οποίο συμφωνούν πάνω στο κύριο κλειδί της επικοινωνίας των δύο πλευρών το οποίο είναι διαφορετικό από το προηγούμενο κλειδί το οποίο χρησιμοποιούταν κατά την ανταλλαγή αυτών των μηνυμάτων.

x. Τελευταίο μήνυμα είναι και από τις δύο πλευρές το μήνυμα τέλους Finished με το οποίο δηλώνουν ότι η διαδικασία ανταλλαγής μηνυμάτων τελείωσε και ήταν ή επιτυχής και η σύνδεση επιτεύχθηκε είτε αποτυχής και η σύνδεση θα πρέπει να τερματιστεί.

Αυτές οι μέθοδοι μπορούν να είναι υλοποιημένες και ακόλουθα σημεία:

1. Στον διακομιστή της υπηρεσίας VPN

Όταν η υπηρεσία αυθεντικοποίησης είναι στον ίδιο τον διακομιστή τότε η διαδικασία γίνεται αυτόματα από τον ίδιο.

2. Σε κάποιον AAA διακομιστή. Αυτοί οι διακομιστές είναι υπεύθυνοι για την αυθεντικοποίηση, την εξουσιοδότηση και την χρέωση.Τέτοιοι διακομιστές είναι:

a) RADIUS (Remote Authentication Dial-In User Service Server)

Το RADIUS αποτελεί ένα πρωτόκολλο το οποίο χρησιμοποιείται για τη διαχείριση AAA υπηρεσιών. Η υπηρεσίες αυθεντικοποίησης περιγράφονται στο RFC 2865. Σε αυτό περιγράφεται η λειτουργία του μαζί με τα πρωτόκολλα PAP και CHAP. Το RADIUS λειτουργεί χρησιμοποιώντας πακέτα τα οποία αποτελούνται από AVP (Attribute Value Pair). Στην περίπτωση του PAP τα δεδομένα στέλνονται κρυπτογραφημένα και κατακερματισμένα με τον αλγόριθμο MD5 στον διακομιστή ο οποίος τα συγκρίνει με τα στοιχεία που έχει και επιστρέφει αν είναι σωστά ή όχι. Στην περίπτωση του CHAP τα πακέτα που στέλνονται αποτελούνται από την τιμή 3 στον τύπο του μηνύματος και περιέχει την ταυτοποίηση του χρήστη και την απάντηση που έδωσε στην πρόκληση του διακομιστή της αυθεντικοποίησης. Ο διακομιστής της αυθεντικοποίησης ακόμα στέλνει το μήνυμα τύπου 60 το οποίο περιέχει την πρόκληση.

b) Diameter Server

Το Diameter αποτελεί μια αναβάθμιση του πρωτοκόλλου RADIUS. Υλοποιήθηκε έτσι ώστε να υποστηρίζει το SCTP (Stream Control Transmission Protocol) στην θέση που το RADIUS λειτουργούσε με UDP, να έχει την δυνατότητα διαπραγμάτευσης να υποστηρίζει νέες εντολές αλλά και να είναι ευθυγραμμισμένο στα 32 bit.

Μια από αυτές τις νέες εντολές είναι η υποστήριξη του EAP μέσω αυτού το οποίο οριστικοποιήθηκε στο RFC 4072. Για την ασφάλεια των πακέτων Diameter μεταξύ του διακομιστή της αυθεντικοποίησης και του AAA διακομιστή χρησιμοποιείται τούνελ TLS, παλαιότερα IPSec, ενώ στο RFC 6733 ορίζεται ότι όταν στο επίπεδο μεταφοράς χρησιμοποιείται SCTP τότε τα πακέτα θα πρέπει να περνάνε από DTLS (Datagram Transport Layer Security) τούνελ.

2.3 Ευπάθειες Χρήστη

Κατά την χρήση του VPN ο χρήστης παρόλο που στήνεται το τούνελ κατά τέτοιο τρόπο ώστε να είναι ανώνυμος και να προστατεύονται τα στοιχεία ταυτότητάς του, δεν του παρέχεται η απόλυτη ανωνυμία σε κανένα επίπεδο. Υπάρχουν πολλά κομμάτια τα οποία ακόμα δεν είναι ασφαλής και τα δεδομένα του ιδιωτικά.

Σε όλες τις πρωταρχικές φάσεις της έναρξης της ζεύξης μεταξύ του χρήστη και του διακομιστή της υπηρεσίας υπάρχουν κομμάτια τα οποία μπορούν να "σπαστούν" από κάποιο επίδοξο ο οποίος θέλει να αποκομίσει κάποια στοιχεία.

Τα στοιχεία που μπορούν είναι η IP του χρήστη, η τοποθεσία του αλλά και στοιχεία τα οποία μπορεί να μοιραστεί νομίζοντας ότι βρίσκεται σε μια ασφαλή σύνδεση όπως στοιχεία καρτών ή άλλα.

Κατά την διάρκεια της αυθεντικοποίησης του χρήστη επίσης μπορούν να υποκλαπούν στοιχεία όπως κωδικοί πρόσβασης στις υπηρεσίες VPN γεγονός το οποίο μπορεί να αποβεί μοιραίο καθώς κάποιος μπορεί να αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες ενός ιδιωτικού δικτύου.

Οι ευπάθειες αυτές προκαλούνται συνήθως είτε από ελλιπείς υλοποιήσεις των πρωτοκόλλων, είτε από τους μηχανισμούς κρυπτογράφησης που χρησιμοποιούνται είτε από κάποιο πρόγραμμα το οποίο χρησιμοποιεί ανασφαλείς συνδέσεις για την μεταφορά των δεδομένων, όπως π.χ. κάποια πόρτα η οποία δεν ανήκει στο VPN. Επιπλέον ο χρήστης είναι ευάλωτος σε περίπτωση που ο διακομιστής της υπηρεσίας VPN δεν έχει τα απαραίτητα μέτρα ασφαλείας όπως για παράδειγμα firewall.

Μία μελέτη έδειξε ότι το 90% των VPN που χρησιμοποιούν SSL για την υλοποίηση του VPN δεν τηρούν ενημερωμένους μηχανισμούς κρυπτογράφησης για την ασφάλεια των χρηστών τους.Αντίστοιχα το 74% δεν χρησιμοποιούν ενημερωμένους αλγόριθμους ψηφιακής υπογραφής. Τέλος το 41% χρησιμοποιεί πιστοποιητικά τα οποία είναι μεγέθους 1024 bit τα οποία θεωρούνται ανασφαλή. (Layden J,2016)

Κάθε μηχανισμός με την πάροδο του χρόνου γίνεται ολοένα και πιο ευάλωτος στις επιθέσεις, λόγω της προόδου της τεχνολογίας, καθιστώντας τους χρήστες ευάλωτους σε διάφορες επιθέσεις.

2.4 Ιοί και κακόβουλο Λογισμικό

Τα VPN δίκτυα δεν αποτελούν πανάκεια για την ασφάλεια ούτε του χρήστη ούτε του δικτύου που συνδέεται εάν αυτό είναι ιδιωτικό. Αυτό δεν σημαίνει όμως ότι οι χρήστες VPN δέχονται περισσότερες επιθέσεις από ότι κάποιος άλλος χρήστης.

Λόγω του γεγονότος ότι ο χρήστης είναι συνδεδεμένος μέσω τούνελ το οποίο αποκρύπτει τα πραγματικά στοιχεία του δικτύου του, κυρίως την διεύθυνση IP του, είναι πολύ πιο προστατευμένος από την απευθείας σύνδεσή του στο διαδίκτυο. Αυτό γίνεται γιατί τα στοιχεία τα οποία ανταλλάσσει είναι κρυπτογραφημένα, οπότε δεν μπορούν να υποκλαπούν κάνοντας τα απρόσβλητα σε επιθέσεις οι οποίες στοχεύουν συγκεκριμένα δεδομένα. Ακόμα οποιαδήποτε επίθεση η οποία πάει να γίνει στον χρήστη ουσιαστικά χτυπάει πρώτα στους μηχανισμούς ασφαλείας του διακομιστή VPN. Εάν αυτοί οι μηχανισμοί είναι σωστά υλοποιημένοι και παρέχουν όλους τους μηχανισμούς ασφάλειας τότε ο χρήστης έχει την μέγιστη ασφάλεια.

Πολλοί όμως δωρεάν πάροχοι υπηρεσιών VPN δεν τηρούν τα απαραίτητα μέτρα ασφαλείας και είναι εύκολος στόχος σε ιούς και κακόβουλο λογισμικό το οποίο διαμοιράζουν στην συνέχεια στους χρήστες τους. Επιπλέον τέτοιοι πάροχοι γίνονται στόχος χάκερ λόγω των ελλείψεών τους σε ασφάλεια γιατί μπορούν να πάρουν μαζικά δεδομένα από όλους τους χρήστες της υπηρεσίας. Τέλος

πολλοί δωρεάν πάροχοι μπορεί να βάζουν οι ίδιοι κακόβουλο λογισμικό στους υπολογιστές των χρηστών όπως πληθώρα διαφημίσεων αλλά και να συλλέγουν προσωπικά δεδομένα και να τα πουλάνε σε συλλέκτες δεδομένων.

Γενικά πρέπει να γίνεται σωστή επιλογή για το ποια υπηρεσία VPN θα χρησιμοποιήσει ο χρήστης αλλά και να έχει λάβει και αυτός όλα τα μέτρα ασφαλείας, όπως firewall και anti-virus.

2.5 Μη εξουσιοδοτημένη Πρόσβαση

Όπως αναφέραμε για τα VPN δίκτυα ο κάθε χρήστης θα πρέπει να αυθεντικοποιείται ενάντια στον διακομιστή τον οποίο θέλει να συνδεθεί με βάση κάποια μοναδικά αναγνωριστικά όπως όνομα χρήστη και κωδικό πρόσβασης ή κάποιο μοναδικά παραγόμενο κωδικό είτε με την χρήση κάποιου πιστοποιητικού.

Αυτό όμως από μόνο του δεν μπορεί να ασφαλίσει επαρκώς τα δεδομένα τα οποία κινούνται μεταξύ του διαδικτύου μέσω του τούνελ και του δικτύου. Αναφέροντας συγκεκριμένα για εταιρικά δίκτυα και απομακρυσμένη πρόσβαση σε αυτά τότε επιβάλλεται ο κάθε χρήστης να έχει πρόσβαση σε συγκεκριμένους πόρους δικτύου οι οποίοι απευθύνονται μοναδικά σε αυτόν και στα στοιχεία τα οποία έχει δώσει.

Η μη εξουσιοδοτημένη πρόσβαση μπορεί να αποτελέσει μεγάλο κενό ασφάλειας σε κάθε δίκτυο και να οδηγήσει σε απόσπαση ιδιαιτέρως ευαίσθητων πληροφοριών, όπως οικονομικά συμβόλαια, τεχνικά σχέδια κ.α.

Τέτοιο κενό ασφαλείας μπορεί να το εκμεταλλευτούν επίσης αποσπώντας τα στοιχεία αυθεντικοποίησης κάποιου χρήστη.Για την αποτροπή τέτοιων κενών η προτιμότερη λύση είναι να μην γίνεται αποθήκευση στοιχείων του χρήστη σε διακομιστή VPN αλλά και χρήση περισσότερο ασφαλών πρωτοκόλλων. Η χρήση κάποιου AAA διακομιστή ο οποίος είναι ασφαλισμένος και μη προσβάσιμος από κάποιον εκτός του δικτύου είναι η ασφαλέστερη λύση. Λύσεις ακόμα αποτελούν, εάν δεν θεωρείται απαραίτητη η χρήση κάποιου τέτοιου διακομιστή, η χρήση ασφαλέστερων πρωτοκόλλων αυθεντικοποίησης αλλά και η αποθήκευση των κωδικών με τέτοιο τρόπο έτσι ώστε να μην μπορούν να αποκρυπτογραφηθούν από κάποιον ο οποίος δεν ξέρει τον αλγόριθμο κρυπτογράφησης.

Ακόμα μπορούν να υλοποιηθούν μηχανισμοί αποκλεισμού ταυτόχρονης σύνδεσης χρηστών από δυο διαφορετικές διευθύνσεις ή περιστασιακή αλλαγή των μεθόδων ασφαλείας όπως για παράδειγμα τα πιστοποιητικά ή οι κωδικοί των χρηστών.

2.6 Προσαρμοστικότητα

Η ευκολία και η ο τρόπος με τον οποίο λειτουργούν τα VPN δίκτυα τα κάνουν εύκολα να υλοποιηθούν πάνω σε οποιοδήποτε μέσο. Δεν έχει σχέση το τι θα είναι το δίκτυο στο οποίο βρίσκεται ο χρήστης, ιδιωτικό ή δημόσιο. Δεν έχει ούτε σημασία εάν είναι ενσύρματο ή ασύρματο.

Η μόνη προϋπόθεση για την υλοποίηση ενός VPN δικτύου είναι ότι όλα τα στοιχεία δικτύου τα οποία βρίσκονται ανάμεσα στον χρήστη και τον διακομιστή να υποστηρίζουν την τεχνολογία. Επιπλέον βέβαια για την λειτουργία όλων των στοιχείων σε ένα VPN τούνελ θα πρέπει τα υλοποιημένα πρωτόκολλα στον χρήστη να υπάρχουν και στον διακομιστή.

2.7 Ζητήματα ασφαλείας στα VPN δίκτυα

Παρόλο που τα δίκτυα VPN υλοποιούνται έτσι ώστε να είναι ασφαλή για τον χρήστη και να προστατεύουν την ιδιωτικότητά του πολλές φορές δεν καλύπτουν όλες τις απαραίτητες προϋποθέσεις. Ειδικά οι δωρεάν πάροχοι VPN τις περισσότερες φορές παρέχουν ελλειπή ασφάλεια για τον χρήστη έτσι ώστε τελικά να καθιστούν την χρήση του δικτύου VPN μη ουσιαστική.

Από τα πιο σημαντικά ζητήματα ασφαλείας είναι η διαρροή της διεύθυνσης IP του χρήστη μέσα από το τούνελ VPN. Μέσα από μια μελέτη σε 14 από τους μεγαλύτερους παρόχους VPN παρατηρήθηκε ότι όταν ο χρήστης συνδέεται χρησιμοποιώντας το πρωτόκολλο IPv6 τότε η διεύθυνσή του είναι ακόμα εμφανής στους κόμβους τους οποίου περνάνε τα δεδομένα του, ή ο DNS ο οποίος χρησιμοποιεί μπορεί να έχει πέσει θύμα επίθεσης.(Vasile, 2015)

Ένα ακόμα ζήτημα είναι οι επιθέσεις DoS(Denial-of-Service) οι οποίες θα αναπτυχθούν παρακάτω . Αποτελούν τις πιο κοινές επθέσεις στους servers(εξυπηρετητές). Συνήθως μειώνουν την διαθεσιμότητα των πόρων του δικτύου. Ο επιτιθέμενος δημιουργεί μεγάλες υπολογιστικές διεργασίες, οι οποίες ονομάζονται πλήμμύρες(flood), με τεράστιο όγκο διπλών πακέτων με σκοπό το θύμα της επίθεσης αυτής να μην μπορεί να χρησιμοποιήσεις τις υπηρεσίες του δικτύου για πολλές ίσως μέρες, Υπάρχουν πολλών ειδών τέτοιες επιθέσεις όπως UDP ,ICMP , SYN πλήμμύρες κ.α. Γι αυτό και υπάρχουν διάφορα συστήματα τα οποία είναι υπέυθυνα να αντιμετωπίζουν αυτές τις επιθέσεις.(Madhavi R, 2015)

Επιπλέον άλλο ζήτημα ασφαλείας είναι η επίθεση του ενδιάμεσου. Η επίθεση το ενδιάμεσου είναι όταν κάποιος συνδέεται στην πορεία μετάδοσης των δεδομένων με σκοπό να τα υποκλέψει. Παρόλο που μεταφέρονται ασφαλώς μέσα από το τούνελ αν οι αλγόριθμοι κρυπτογράφησης δεν είναι σωστά υλοποιημένοι τότε μπορεί να έχει κάποιος τρίτος πρόσβαση στα δεδομένα. Για αυτό το λόγο πολλοί πάροχοι υπηρεσιών VPN έχουν την ασφάλεια να διακόπτουν την σύνδεση εάν βλέπουν κάποια αλλαγή στην δρομολόγηση των πακέτων ή αλλαγή στην διεύθυνση IP του χρήστη.

Πέρα από τα ζητήματα υπηρεσιών πολλές εφαρμογές όπως το WebRTC το οποίο είναι μια τεχνολογία η οποία επιτρέπει τηλεδιάσκεψη και τον διαμοιρασμό συσκευών μέσω του φυλλομετρητή(browser) έχουν κενά ασφαλείας τα οποία εμφανίζουν την διεύθυνση IP του χρήστη.

Εκτός όμως από τον χρήστη ζητήματα ασφαλείας μπορούν να προκύψουν και στα δίκτυα τα οποία συνδέεται ο χρήστης αν αυτά είναι ιδιωτικά δίκτυα. Τέτοια ζητήματα ασφαλείας είναι κυρίως το ότι μέσα από το VPN ο χρήστης παίρνει δικαιώματα πρόσβασης στο δίκτυο είτε τα χρειάζεται είτε όχι. Η χρήση VPN πολλές φορές στα ιδιωτικά δίκτυα αποτελεί κενό ασφαλείας ειδικά όταν υπάρχουν συνδέσεις ταυτόχρονα σε πολλές τοποθεσίες και η κάθε μια υλοποιεί το δικό της VPN. Θα πρέπει όλες οι πολιτικές προστασίας να είναι διάφανες και ενημερωμένες σε όλες τις τοποθεσίες ταυτόχρονα. Τέλος εάν σε αυτά τα δίκτυα έχουν πρόσβαση και εξωτερικοί συνεργάτες μέσω VPN θα πρέπει οι πολιτικές ασφαλείας να είναι ακόμα πιο μεγάλες και ο κατακερματισμός του δικτύου σε περιοχές πρόσβασης ανάλογα με τον VPN διακομιστή που συνδέονται ακόμα μεγαλύτερος, έτσι ώστε να μην είναι δυνατή η διαρροή ευαίσθητων πληροφοριών.(Akamai, 2017)


Κεφάλαιο 3
Επιθέσεις στους
διακομιστές VPN

3 Επιθέσεις στους διακομιστές VPN

Παρότι έχουμε δει οτι χρήστες και εταιρίες χρησιμοποιούν τα VPN για την ασφαλή τους περιήγηση σε τοποθεσίες δικτύου και πόρους δεν παύουν να απειλούνται απο επιθέσεις που συμβαίνουν στα δίκτυα γενικά με τις πιο συχνές επιθέσεις να είναι τύπου DoS . Γενικά οι επιθέσεις που συμβαίνουν στα VPN δίκτυα είναι είτε επιθέσεις εισβολής είτε επιθέσεις DoS.Οι επιθέσεις εισβολής είναι επιθέσεις που έρχονται έξω απο το VPN παρέχοντας μη-εξουσιοδοτημένη πρόσβαση στο VPN ενώ οι επιθέσεις DoS γίνονται είτε απ έξω είτε απο μέσα απο το δίκτυο με σκοπό να διακόψουν την επικοινωνία μεταξύ των εξουσιοδοτημένων χρηστών και έχουν πολλούς και διάφορους τύπους ανάλογα το επίπεδο που στοχεύουν.Οι επιθέσεις εισβολής παρακάμπτουν την ασφάλεια και συνέπειες τέτοιων επιθέσεων αφορούν την εμπιστευτικότητα , την ακεραιότητα και την διαθεσιμότητα των πόρων του δικτύου. Απο την άλλη, οι επιθέσεις DoS αποτελούν πιο σοβαρό κίνδυνο. Μια τέτοια επίθεση αποστέλλει τεράστιο όγκο δεδομένων στο θύμα της επίθεσης. Ο όγκος αυτός μπορεί να σταλθεί ως πολύ μικρα πακέτα ή ως μια ασταμάτητη ροή. Τέτοιες επιθέσεις(DoS) στοχεύουν είτε στα πρωτόκολλα(πχ IPSec) είτε στην δομή του δικτύου(DNS). Επιθέσεις στα πρωτόκολλα αποτελούν οι επιθέσεις TCP SYN flood, ping of death, teardrop(fragmented packet attack). Ο επιτιθέμενος λαμβάνει υπόψιν του τις ατέλειες που υπάρχουν στα πρωτόκολλα και τις υπηρεσίες τους και επιτίθεται στην μετάδοση των πακέτων ενώ, στις επιθέσεις δομής ο επιτιθέμενος επιτίθεται όπως είπαμε στις δομές του δικτύο όπως ο δρομολογητής ή τους διακομιστές . Οι επιθέσεις αυτές σκοπό έχουν να δημιουργήσουν καθυστέρηση , χάσιμο πακέτων και επηρεάζουν τον ρυθμό παράδοσης των πακέτων.Ένα πακέτο που μεταδίδεται μέσω του ίντερνετ μπορεί να ακολουθήσει πολλά διαφορετικά μονοπάτια και δεν υπάρχει τρόπος να πρακολουθήσουμε την κινησή του καθώς για το ίντερνετ δεν υπάρχει κεντρική διαχείριση που να πραγματοποιεί αυτό τον έλεγχο. Επιθέσεις DoS στην δομή του δικτύου έχουν επιρροή στην λειτουργία του ίντερνετ και κατ' επέκταση στην λειτουργία του VPN. Επιπρόσθετα, οι επιθέσεις δομής σε άλλους δύο τύπους επιθέσεων σε επιθέσεις στο ρυθμό μετάδοσης(bandwidth) και σε επιθέσεις που εξαντλούν τους πόρους του δικτύου. Στην πρώτη περίπτωση ο επιτιθέμενος μπλοκάρει τον εξουσιοδοτημένο χρήστη με το να επικοινωνήσει με το δίκτυο και καταναλώνει τους πόρους του με σκοπό να μην μπορεί να λειτουργήσει όπως πρέπει. Μεγάλη κατανάλωση στον ρυθμό μετάδοσης προκαλεί κατάρρευση του δικτύου.Ενώ οι επιθέσεις που εξαντλούν τους πόρους του δικτύου στοχεύουν σε κατανάλωση της CPU , RAM και μπορεί να προκαλέσει απώλεια διαθεσιμότητας ή περιεργη συμπεριφορά στο δίκτυο λόγω καθυστέρησης στην επεξεργασία των αιτημάτων. (Saraswathi ,2012)

3.1 Τύποι DoS επιθέσεων

Η διαφορά μεταξύ μιας DoS και μιας DDoS επίθεσης είναι οτι στην επίθεση DoS ο επιτιθέμενος χρησιμοποιεί μια σύνδεση στο ίντερνετ για να φανερώσει μια αδυναμία του λογισμικού ή να «πλήμμυρίσει» τον χρήστη με ψεύτικα αιτήματα με σκοπό να εξαντλήσει τους πόρους της μνήμης RAM ή του επεξεργαστή. Η συνήθης πρακτική για DoS επιθέσεις είναι η ταυτόχρονη αποστολή πολλών αιτημάτων δικτύου προς αυτόν τον διακομιστή έτσι ώστε να μην μπορεί να εξυπηρετήσει τα αιτήματα τα οποία μπορεί να εξυπηρετήσει.

Οι τύποι DoS επιθέσεων οι οποίοι υπάρχουν είναι (Wikipedia,2017):

a) DDoS (Distributed DoS)

Μια Distributed DoS επίθεση είναι μια επίθεση η οποία ξεκινάει ταυτόχρονα από πολλούς υπολογιστές προς έναν συγκεκριμένο διακομιστή έτσι ώστε να γίνει υπερχείλιση με αιτήματα και να μην μπορεί το δίκτυό του να εξυπηρετήσει κανονικά αιτήματα. Αυτό ουσιαστικά καθιστά

αδύνατο το να αντιμετωπιστεί η επίθεση φιλτράροντας τα αιτήματα τα οποία προέρχονται από μία και μόνο πηγή. Άλλος τρόπος είναι τα μηνύματα τα οποία έρχονται στον διακομιστή να στέλνονται με ψεύτικες διευθύνσεις IP πηγής, διαδικασία που ονομάζεται IP Spoofing.

b) Προχωρημένη διαρκής επίθεση DoS (Advanced Persistent DoS)

Οι APDoS επιθέσεις είναι εξελιγμένες επιθέσεις DoS οι οποίες πραγματοποιούνται από άτομα τα οποία έχουν υλικό μεγάλων δυνατοτήτων . Συνήθως είναι επιθέσεις οι οποίες στοχεύουν το HTTP επίπεδο ακολουθούμενη από επιθέσεις οι οποίες στοχεύουν την βάση δεδομένων του διακομιστή. Συνήθως αποτελούνται από δεκάδες εκατομμύρια επιθέσεις το δευτερόλεπτο. Τέτοιες επιθέσεις μένουν εκεί μέχρι να τις σταματήσει αυτός που την άρχισε ή να μπορεί ο διακομιστής να αμυνθεί απέναντι σε αυτές.

Για να αναφερθούμε πιο αναλυτικά στο πως πραγματοποιείται μια τέτοια επίθεση θα πρέπει να δούμε λεπτομερέστερα από τι τύπους αποτελούνται αυτές οι επιθέσεις. (Imperva inc,2017).Οι επιθέσεις που αναφέρονται παρακάτω είναι τύπου DDoS επιθέσεις.

3.1.1 Επιθέσεις με βάση τον όγκο των δεδομένων.

Τέτοιες επιθέσεις είναι οι UDP πλημμύρες (rfc 2827) , οι πλημμύρες ICMP καθώς και άλλες επιθέσεις οι οποίες χαρακτηρίζονται από πλημμύρες με spoofed πακέτα. Σκοπός αυτών των επιθέσεων είναι να μειώσουν το ρυθμό μεταφοράς δεδομένων(bandwidth) του διακομιστή και μετριέται σε bits ανά δευτερό λεπτο (BPS).

Για να μπορέσουμε να αντιμετωπίσουμε τις επιθέσεις θα αναλύσουμε το πως λειτουργούν(Imperva inc,2017).

- UDP flood

Η επίθεση τύπου UDP flood(πλημμύρα) είναι ένας τύπος επίθεσης κατά την διάρκεια του οποίου στέλνονται διάφορα UDP πακέτα σε διαφορετικές πόρτες του επιτιθέμενου διακομιστή. Αυτός μη βρίσκοντας υπηρεσία στην αντίστοιχη πόρτα στέλνει πίσω ένα μήνυμα ότι ο προορισμός δεν είναι προσβάσιμος. Καθώς ολοένα και περισσότερα πακέτα φτάνουν στον διακομιστή καθίσταται αδύνατο το να απαντήσει στα αιτήματα των κανονικών χρηστών.

- ICMP flood

Άλλος ένας τύπος επίθεσης παρόμοιος με το UDP flood είναι και το ICMP flood. Σε αυτή τη περίπτωση τα πακέτα είναι τύπου ping. Αυτά τα πακέτα γεμίζουν τον διακομιστή με πακέτα στα οποία ο επιτιθέμενος γνωρίζει ότι πρέπει να απαντηθούν. Με αυτό το τρόπο γίνεται μεγάλη χρήση τόσο του εισερχόμενου όσο και του εξερχόμενου εύρους της σύνδεσης του διακομιστή οδηγώντας σε DoS.

- NTP Amplification

Για τον παγκόσμιο συγχρονισμό των πακέτων στο διαδίκτυο υπάρχουν ειδικοί διακομιστές οι οποίοι υλοποιούν σαν κύριοι το πρωτόκολλο NTP(Network Time Protocol). Όταν κάποιος πόρος δικτύου θέλει να λάβει την ώρα υλοποιεί το πρωτόκολλο NTP σαν πελάτης και ζητάει από τον διακομιστή κύριο του πρωτοκόλλου την ώρα. Αποτελεί ένα από τα παλαιότερα πρωτόκολλα που έχουν υλοποιηθεί . Εκτός όμως από αυτή τη λειτουργία χρησιμοποιώντας την εντολή monlist μπορεί να ζητηθεί από αυτούς η λίστα των 600 τελευταίων υπολογιστών οι οποίοι συνδέθηκαν.

Η επίθεση τύπου NTP amplification αποστέλλει monlist αιτήματα προς τον διακομιστή κύριο του NTP με διεύθυνση πηγής τον διακομιστή ή χρήστη στον οποίο γίνεται αυτή η επίθεση . επειδή οι απαντήσεις σε αυτά τα αιτήματα είναι πολύ μεγαλύτερες από τα αιτήματα τα ίδια , η κίνηση δεδομένων προς τον διακομιστή θύμα της επίθεσης ενισχύεται οδηγώντας σε μείωση απόδοσης για σωστά αιτήματα.

3.1 2 Επιθέσεις οι οποίες στοχεύουν συγκεκριμένο Πρωτόκολλο

Αυτός ο τύπος επιθέσεων έχει ο σκοπό την κατανάλωση των πόρων του διακομιστή καταναλώνοντας πόρους του.Μετριέται σε πακέτα ανά δευτερόλεπτο (Pps) και συνήθεις τέτοιοι τύποι είναι πλημμύρες SYN(μηνυμάτων συγχρονισμού)(rfc 4987), επιθέσεις κατακερματισμένων πακέτων, η επίθεση Ping of Death, επιθέσεις τύπου Smurf DDoS.

- SYN flood

Η επίθεση τύπου SYN(synchonize) flood εκμεταλλεύεται την τριπλή ανταλλαγή μηνυμάτων του πρωτοκόλλου συγχρονισμού του TCP. Κατά την ανταλλαγή μηνυμάτων του πρωτοκόλλου ο επιτιθέμενος στέλνει μηνύματα συγχρονισμού στον διακομιστή τον οποίο θέλει να επιτεθεί σε όλες τις πόρτες. Συνήθως τα πακέτα έχουν πλαστή διεύθυνση IP. Με αυτό το τρόπο ο διακομιστής απαντάει σε όλα τα αιτήματα με το μήνυμα γνωστοποίησης συγχρονισμού. Επειδή είτε η διεύθυνση του αποστολέα είναι πλαστή είτε επειδή εν γνώσει του δεν απαντάει με μήνυμα επιβεβαίωσης ή απόρριψης του συγχρονισμού ο διακομιστής μένει σε κατάσταση να περιμένει αυτό το μήνυμα. Έτσι όταν κάποιος χρήστης θέλει να ξεκινήσει συγχρονισμό όλες οι πόρτες είναι κατειλημμένες οδηγώντας στην αδυναμία του διακομιστή να δεχτεί το αίτημα αυτό.

- Ping of Death

Ο τύπος επίθεσης ping of death εκμεταλλεύεται την υλοποίηση του πρωτοκόλλου IP όσον αφορά το μέγεθος του πακέτου. Ένα συνηθισμένο πακέτο ping είναι μεγέθους 64 byte. Στο ping of death στέλνει ο επιτιθέμενος μεγάλα πακέτα τύπου ping τα οποία είναι κατακερματισμένα σε πακέτα μεγέθους 64 byte. Όταν ο διακομιστής που λαμβάνει τα πακέτα προσπαθεί να τα επανασυνδέσει τότε μπορεί να συμβεί υπερχείλιση στον buffer, οδηγώντας σε κατάρρευση το σύστημα και πιθανώς επιτρέποντας την εγκατάσταση κακόβουλου κώδικα.Αυτός ο τρόπος επίθεσης αποτελεί προπομπό του ICMP flood όσο τα firewall και άλλοι μηχανισμοί ανίχνευσης επιθέσεων γινόταν όλο και πιο αποτελεσματικοί στο να αναγνωρίζουν τέτοιους τύπους επιθέσεων.

- Smurf DDoS

Μια smurf DDoS επίθεση εκμεταλλεύεται τους μηχανισμούς μετάδοσης σε όλους τους χρήστες του δικτύου (broadcast) στέλνοντας σε όλους ταυτόχρονα πακέτα τύπου ICMP χρησιμοποιώντας την κατασκευασμένη διεύθυνση IP του διακομιστή που θέλει να προσβάλει. Κατόπιν όλοι όσοι πήραν αυτό το μήνυμα απαντάνε σε αυτή τη διεύθυνση δημιουργώντας μεγάλη ροή δεδομένων προς αυτόν τον διακομιστή, καταναλώνοντας πολλές φορές όλους τους πόρους δικτύου του ανάλογα με τον αριθμό των κόμβων που πήραν μέρος στην επίθεση.

- Fragmented packet attack

Ο κατακερματισμός των πακέτων είναι ένας μηχανισμός για την αποστολή μεγάλων πακέτων μέσα από ένα δίκτυο το οποίο έχει πεπερασμένο μέγεθος αποστολής(MTU maximum transmission unit). Όταν ένα μεγάλο πακέτο αποστέλλεται τότε κατακερματίζεται σε μικρότερα πακέτα τα οποία μπορούν να αποσταλούν μέσα από το δίκτυο. Όταν αυτά φτάνουν στον τελευταίο προορισμό τους αυτά επανασυνδέονται έτσι ώστε να συνθέσουν το αρχικό πακέτο.

Οι επιτιθέμενοι εκμεταλλεύονται αυτό το μηχανισμό έτσι ώστε να στέλνουν πολύ μεγάλα πακέτα στο δίκτυο τα οποία συνέχεια είναι ατελή μπλοκάροντας έτσι τους μηχανισμούς χειρισμού πακέτων του διακομιστή καταναλώνοντας του πόρους του.

3.1.3 Επιθέσεις που στοχεύουν το επίπεδο της εφαρμογής

Οι επιθέσεις του επιπέδου της εφαρμογής είναι συνήθεις επιθέσεις οι οποίες στοχεύουν συγκεκριμένες ευπάθειες των στοχευμένων υπηρεσιών. Στόχοι τέτοιων επιθέσεων μπορεί να είναι τα Windows, ένας Apache διακομιστής κ.α. Αυτές οι επιθέσεις μετριούνται σε αιτήματα το δευτερόλεπτο (Rps) και συνήθεις τύποι είναι : low-and-slow επιθέσεις όπως η R.U.D.Y. ή Slowloris, GET/POST επιθέσεις κ.α.

- HTTP (GET/POST) flood

Οι επιθέσεις τύπου HTTP flood είναι επιθέσεις οι οποίες στοχεύουν το επίπεδο εφαρμογής του δικτύου και του διακομιστή. Στη συνήθη πρακτική είναι επιθέσεις οι οποίες αποτελούνται από εντολές GET και POST του πρωτοκόλλου HTTP. Στέλνονται σε ένα διακομιστή διάφορα αιτήματα του πρωτοκόλλου είτε με τυχαία URL είτε με τυχαίες αναζητήσεις τις οποίες ο διακομιστής δεν μπορούσε να απαντήσει. Αυτό οδηγεί τον διακομιστή να καταναλώνει περισσότερους πόρους για να απαντήσει σε κάθε αίτημα. Συνήθως τα μηνύματα τύπου POST παίρνουν περισσότερους πόρους και για αυτό θεωρούνται πιο αποτελεσματικά. Παρόλα αυτά μια τέτοια επίθεση θέλει πολύ προεργασία και καλή γνώση λειτουργίας του πρωτοκόλλου.

- Low and Slow attacks

Οι low and slow επιθέσεις σε ένα διακομιστή είναι επιθέσεις σε νηματικούς διακομιστές με σκοπό την εξάντληση των πόρων του. Ο τρόπος λειτουργίας τους είναι η μετάδοση με χαμηλή ταχύτητα δεδομένων , τόσο χαμηλή έτσι ώστε να φτάνει το επόμενο μήνυμα λίγο πριν λήξει ο χρόνος αναμονής του διακομιστή. Έτσι καθώς στέλνονται ολοένα και περισσότερα αιτήματα και τα παλιά μένουν στην αναμονή οι πόροι του διακομιστή σιγά σιγά εξαντλούνται.

Οι κύριες επιθέσεις low and slow είναι οι:

Slowloris: το εργαλείο αυτό συνδέεται σε ένα διακομιστή και στέλνει σταδιακά μερικώς ολοκληρωμένες κεφαλίδες HTTP. Αυτό οδηγεί τον διακομιστή να κρατάει ανοιχτή τη σύνδεση έτσι ώστε να λάβει και τις υπόλοιπες κεφαλίδες

R.U.D.Y. (Are you dead yet?): Το εργαλείο αυτό στέλνει συνεχή POST αιτήματα προς έναν διακομιστή έτσι ώστε να γεμίσει φόρμες. Το εργαλείο αυτό ενημερώνει τον διακομιστή πόσα δεδομένα να περιμένουν αλλά τα στέλνει πολύ αργά.

Sockstress: Η Sockstress επίθεση εκμεταλλεύεται μια ευπάθεια στην τριπλή ανταλλαγή μηνυμάτων κατά την έναρξη του πρωτοκόλλου TCP το οποίο οδηγεί σε μια αόριστη σύνδεση.

Για όλες αυτές τις επιθέσεις υπάρχει τρόπος να αντιμετωπιστούν χρησιμοποιώντας κατάλληλες ρυθμίσεις στα firewall των διακομιστών αλλά και με σωστή και ενημερωμένη υλοποίηση των πρωτοκόλλων επικοινωνίας.

3.2 Επιθέσεις στους DNS διακομιστές

Δεν είναι μόνο όμως οι κανονικοί διακομιστές οι οποίοι πέφτουν θύματα επιθέσεων. Ένα από τα σημαντικότερα στοιχεία σε κάθε δίκτυο είναι οι DNS διακομιστές. Αυτοί οι διακομιστές έχουν τον ρόλο του μεταφραστή μεταξύ των αιτημάτων και του που θα πάνε. Κατά την έναρξη λειτουργίας ενός δρομολογητή αυτός ζητάει από τον DNS τις καταχωρήσεις που χρειάζεται έτσι ώστε να μπορεί να μεταφράσει κάθε ηλεκτρονική διεύθυνση, π.χ. το site του TEI Ηπείρου (www.teiep.gr) στην διεύθυνση IP (195.130.72.52) έτσι ώστε να μπορεί μετά να στέλνει τα αιτήματα απευθείας σε αυτό το site πιο γρήγορα γεμίζοντας την cache του με τους λεγόμενους πίνακες δρομολόγησης(routing tables). Επιπλέον στο δίκτυο για να μην ζητούνται συνέχεια τα στοιχεία από κάθε χρήστη, συνήθως οι παροχείς δικτύου έχουν διακομιστές με το όνομα recursive resolvers και ο χρήστης ζητάει από αυτούς πρώτα την δρομολόγηση και αυτοί κατόπιν από τον DNS.

Μια επίθεση σε έναν DNS διακομιστή μπορεί εύκολα να ρίξει το διαδίκτυο μιας ολόκληρης περιοχής ή ενός παροχέα υπηρεσιών διαδικτύου ανάλογα με το τι DNS χρησιμοποιεί ο κάθε χρήστης.

Οι μηχανισμοί ασφαλείας στους DNS διακομιστές είναι αρκετά αυξημένοι αλλά και πάλι υπάρχουν τύποι επιθέσεων οι οποίες μπορούν να πραγματοποιηθούν σε τέτοιους διακομιστές.

Ο πρώτος τύπος επίθεσης δεν είναι τύπου DDoS αλλά είναι αρκετά επικίνδυνος για τους χρήστες οι οποίοι χρησιμοποιούν τον συγκεκριμένο διακομιστή. Η επίθεση αυτή ονομάζεται DNS spoofing. Σκοπός αυτής της επίθεσης είναι το να ανακατευθυνθούν οι χρήστες σε κάποιο πλαστό site με σκοπό την απόσπαση στοιχείων από αυτόν. Ο τρόπος με τον οποίο υλοποιείται είναι είτε πειράζοντας τον DNS διακομιστή (DNS Hijacking), είτε ξεκινώντας μια επίθεση “δηλητηριασμού” (DNS Poisoning) η οποία αλλάζει τα στοιχεία του διακομιστή(όπως η επίθεση Kaminsky), είτε υλοποίηση μιας επίθεσης ενδιάμεσου εάν υπάρχει πρόσβαση στο δίκτυο του διακομιστή είτε με πολλούς άλλους τρόπους.(NIST, 2008)

Όσον αφορά επιθέσεις DDoS εναντίων DNS διακομιστών αυτές μπορούν να διαχωριστούν σε DNS amplification επιθέσεις και DNS flood επιθέσεις. Όπως οι αντίστοιχες κανονικές επιθέσεις τύπου flood ο διακομιστής DNS δέχεται ένα υπερβολικά μεγάλο αριθμό αιτημάτων με σκοπό το να εξαντληθούν οι πόροι του, όπως μνήμη ή επεξεργαστική ισχύς. Αυτές οι επιθέσεις εκμεταλλεύονται το γεγονός ότι τα αιτήματα προς έναν διακομιστή DNS εξυπηρετούνται μέσω του πρωτοκόλλου UDP και δεν χρειάζεται να είναι έγκυρα για να φτάσουν μέχρι τον διακομιστή.

Τέτοια επίθεση είναι η επίθεση τύπου NXDOMAIN στην οποία ο επιτιθέμενος στέλνει πολλαπλά αιτήματα για διευθύνσεις δικτύου οι οποίες δεν υπάρχουν. Ο διακομιστής προσπαθεί να βρει αυτές τις διευθύνσεις κάνοντας αιτήματα προς άλλους διακομιστές. Κατά την διάρκεια η μνήμη cache του διακομιστή γεμίζει με μηνύματα τύπου NXDOMAIN (non-existent domain) καθιστώντας τον πολύ αργό να απαντήσει σε σωστά αιτήματα άλλων χρηστών. Επιπλέον γίνεται ακόμα πιο αργός καθώς καταναλώνει πόρους για να στείλει τα αιτήματα τα οποία χρειάζονται για να βρεθεί η μη έγκυρη διεύθυνση η οποία ζητάται.Άλλη επίθεση τύπου DDoS εναντίων DNS η οποία αποτελεί παραλλαγή της NXDOMAIN είναι η Phantom Domain επίθεση. Σε αυτή τη περίπτωση αντί να ζητούνται διευθύνσεις οι οποίες δεν υπάρχουν, ζητούνται διευθύνσεις οι οποίες βρίσκονται σε διακομιστές έτσι φτιαγμένους από τον επιτιθέμενο ώστε να μην απαντάνε ποτέ ή να απαντάνε στα ζητούμενα αιτήματα πάρα πολύ αργά. (Fulton S, 2015)

3.3 Επιθέσεις με χρήση DNS διακομιστών

Με τον ρόλο τον οποίο έχουν οι DNS διακομιστές και με την πρόσβαση σε όλο το διαδίκτυο εκτός από θύματα μιας επίθεσης μπορούν να γίνουν και μεσάζοντες υποβοηθώντας τον επιτιθέμενο να πραγματοποιήσει μιας μεγαλύτερης κλίμακας DDoS επίθεσης στο θύμα του. Τέτοιες επιθέσεις είναι οι επιθέσεις τύπου DNS Amplification, DrDoS , οι DNS tunneling αλλά και οι επιθέσεις Slow Drip (Random subdomain).

Οι επιθέσεις τύπου DNS Amplification ουσιαστικά χαρακτηρίζουν επιθέσεις ασυμμετρικές, δηλαδή το μέγεθος των δεδομένων που ζητείται και καταλήγει στον υπό επίθεση κόμβο είναι πολλαπλάσια του μεγέθους του αιτήματος. Τέτοιου τύπου επιθέσεις είναι με αποστολή μηνυμάτων σε πολλαπλούς διακομιστές DNS οι οποίοι απαντούν στον υπό επίθεση κόμβο. Άλλος τρόπος είναι με την χρήση του πρωτοκόλλου EDNS0 το οποίο επιτρέπει την αποστολή μεγάλων μηνυμάτων DNS, είτε χρησιμοποιώντας το πρωτόκολλο που αποστέλλει κρυπτογραφημένα μηνύματα DNSSEC. Τέλος ένας πολύ εύκολος τρόπος για DNS Amplification επίθεση είναι η αποστολή μηνύματος με αίτημα τύπου “ANY” το οποίο επιστρέφει όλες τις πληροφορίες δρομολόγησης του διακομιστή.Όλα αυτά μπορούν να προσφέρουν μια ενίσχυση του μεγέθους 70:1 , για 60 byte αίτημα να ληφθεί μια απάντηση μεγέθους 4000 byte. Αντίστοιχη επίθεση με την DNS Amplification είναι η DrDoS (Distributed Reflection DoS) η οποία απλά δεν έχει τον ενισχυτικό χαρακτήρα των amplification επιθέσεων.

Οι επιθέσεις Slow Drip μοιάζουν πολύ με τις επιθέσεις Phantom domain με την διαφορά όμως ότι χρησιμοποιούνται πραγματικές διευθύνσεις (π.χ. www.teiep.gr) στις οποίες προστίθεται ένα τυχαίο πρόθεμα αντί για το www (π.χ. xyz123.teiep.gr). Όταν πολλά τέτοια αιτήματα φτάνουν σε έναν DNS διακομιστή καθώς δεν τα βρίσκει και πρέπει να αιτηθεί για αυτά στην διεύθυνση που ορίζεται δεσμεύει πόρους για κάθε τέτοιο αίτημα. Ακολούθως όλα αυτά τα αιτήματα οδηγούνται στον διακομιστή της διεύθυνσης πραγματοποιώντας μια DDoS επίθεση. Έτσι ο επιτιθέμενος έχει πραγματοποιήσει ταυτόχρονα δύο επιθέσεις σε μια.

Τέλος μια ακόμα επίθεση με την χρήση των DNS είναι η DNS tunneling επίθεση. Ο επιτιθέμενος σε αυτή την επίθεση ενσωματώνει κανονικά πακέτα σε πακέτα τύπου DNS έτσι ώστε να προσπεράσει τυχόν firewall τα οποία υπάρχουν έτσι ώστε να περάσει αιτήματα όπως απομακρυσμένης πρόσβασης ή γενικώς να εισάγει κακόβουλο λογισμικό στον υπό επίθεση κόμβο.Με αυτό το τρόπο δεν μπορεί να γίνει εύκολα αντιληπτός και μπορεί να αποκτήσει πρόσβαση στον υπό επίθεση κόμβο.

3.4 Στοιχεία κίνησης δικτύου επιθέσεων

Οι κυβερνοεπιθέσεις γίνονται ολοένα και περισσότερες όσο περνάει ο καιρός. Το γεγονός αυτό ενισχύεται καθώς ολοένα και περισσότερες συσκευές μπορούν να χρησιμοποιηθούν για την πραγματοποίηση τέτοιων επιθέσεων. Οι συσκευές IoT πληθαίνουν και καθώς οι μηχανισμοί ασφαλείας που χρησιμοποιούν είναι σχετικά μικροί είναι εύκολα θύματα για την εγκατάσταση κακόβουλου λογισμικού το οποίο υποβοηθάει τις επιθέσεις.

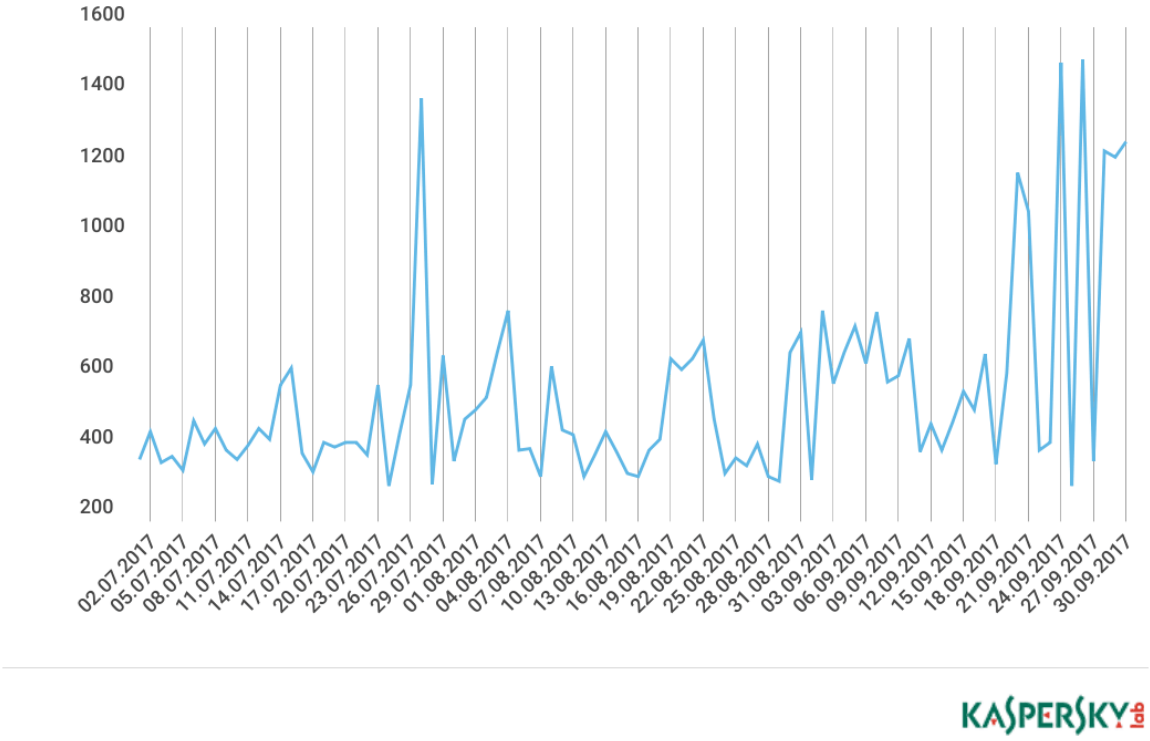
Ένα από τα κακόβουλα λογισμικά τα οποία αναπτύσσονται αυτή τη περίοδο και προσβάλει συσκευές IoT είναι το botnet Mirai(Wikipedia,2017). Αυτό είναι υπεύθυνο για πολλές επιθέσεις οι οποίες πραγματοποιούνται. Η λειτουργία του είναι απλή. Ψάχνει σε ένα εύρος διευθύνσεων IP για συσκευές IoT. Εάν βρει κάποια τέτοια συσκευή τότε χρησιμοποιώντας μια λίστα από προεγγραμένα ονόματα χρήστη και κωδικούς πρόσβασης αποκτάει πρόσβαση σε αυτή τη συσκευή και αναπαράγει τον εαυτό του. Όταν δοθεί εντολή τότε ξεκινάει επίθεση προς κάποιον στόχο. Αυτό το λογισμικό ήταν υπεύθυνο για την μεγαλύτερη επίθεση DDoS της ιστορίας οι οποίες πραγματοποιήθηκαν το

2016. Στις 21 Οκτωβρίου η εταιρία Dyn έπεσε θύμα επίθεσης για μια περίπου ημέρα. Η Dyn κατέχει τους αρκετούς DNS διακομιστές του συνολικού διαδικτύου. Από αυτή την επίθεση "έπεσαν" πολλές ιστοσελίδες όπως το CNN, Twitter, Netflix κ.α. Η δύναμη της επίθεσης υπολογίστηκε στα 1,2 Tbps , δηλαδή 12 * 10^{12} bytes ανά δευτερόλεπτο και χρησιμοποιούσε περίπου 150000 μολυσμένες συσκευές.(Woolf N, 2016)

Χρησιμοποιώντας στοιχεία από μια από τις μεγαλύτερες εταιρίες ηλεκτρονικής ασφάλειας την Kaspersky μπορούμε να δούμε ότι για το 3ο τρίμηνο του 2017(Kaspersky Lab,2017):

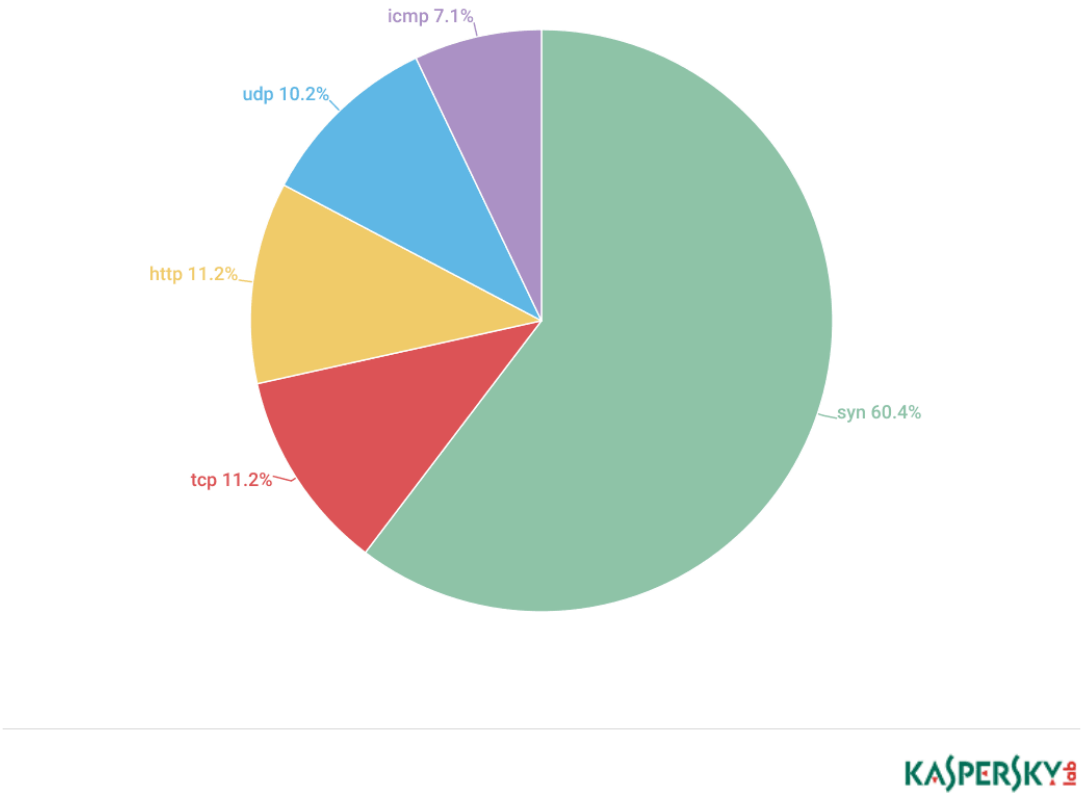
- Η μεγαλύτερη επίθεση DDoS είχε διάρκεια 215 ώρες και το 99,6% των επιθέσεων είχαν διάρκεια μικρότερη από 50 ώρες.
- Οι περισσότερες επιθέσεις ήταν τύπου SYN και HTTP flood.

Γραφικώς μπορούμε να δούμε τις επιθέσεις ανά μέρα:



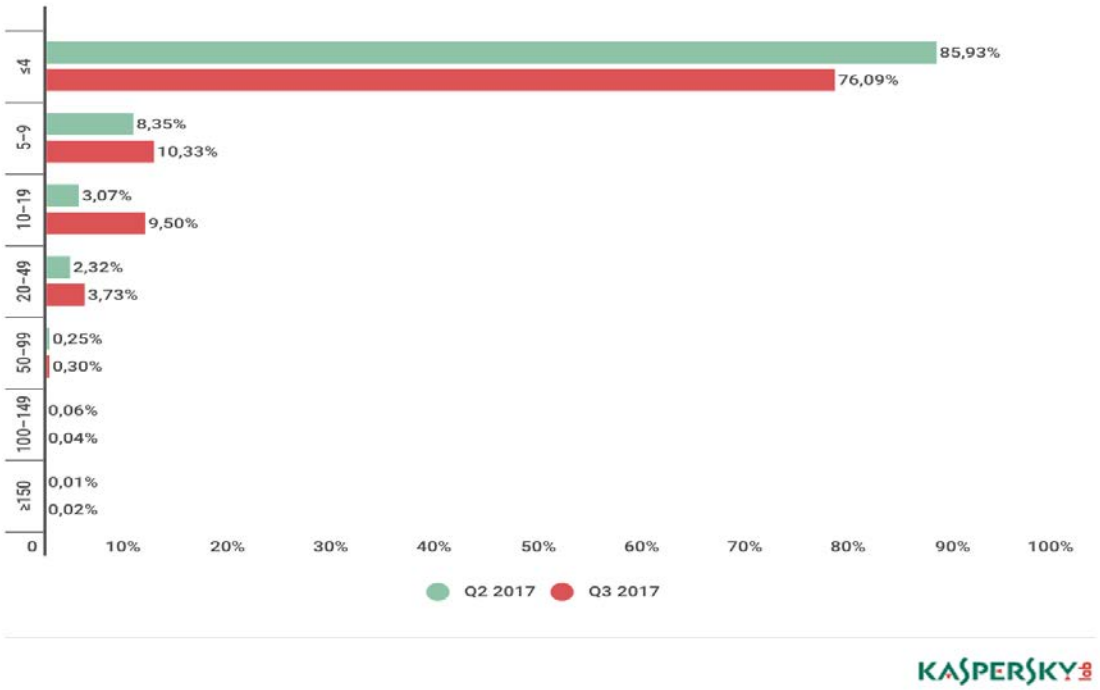
3.1 Επιθέσεις ανα ημέρα (Kaspersky,2017)

Τις επιθέσεις ανά τύπο :



3.2 Επιθέσεις ανα τύπο (Kaspersky ,2017)

Και τις επιθέσεις κατά διάρκεια:



3.3 Επιθέσεις κατα διάρκεια(Kaspersky,2017)

Από την εταιρία Akamai για τις 27 Νοεμβρίου 2017 στο διαδραστικό χάρτη επιθέσεων της έχουν αναφερθεί οι ακόλουθοι αριθμοί επιθέσεων ανά γεωγραφική περιοχή:

- Ηνωμένο Βασίλειο 611876 επιθέσεις
- Γαλλία 344403 επιθέσεις
- Ρωσία 327934 επιθέσεις
- Βιρτζίνια, Η.Π.Α. 1836719 επιθέσεις
- Καλιφόρνια, Η.Π.Α 1206484 επιθέσεις
- Νιου Τζέρζει, Η.Π.Α. 724012 επιθέσεις
- Κίνα 1601637 επιθέσεις
- Ταϊβάν 631639 επιθέσεις

Με βάση τα προηγούμενα δεδομένα μπορούμε να βγάλουμε το συμπέρασμα ότι οι περισσότερες επιθέσεις προέρχονται από την Αμερική ή τις χώρες την Νοτιοανατολικής Ασίας οι οποίες κατέχουν τον τεχνολογικό εξοπλισμό με τον οποίο μπορεί μια τέτοια επίθεση να καταστεί δυνατή. Επιπλέον οι περισσότερες επιθέσεις δεν έχουν μεγάλη διάρκεια, μικρότερη απο 4 ώρες γιατί κατα κύριο λόγο οι μηχανισμοί αντιμετώπισής τους μπορούν αναγνωρίζοντάς τες να ανταποκριθούν με τέτοιο τρόπο ώστε να τις αποκρούσουν.

Όσον αφορά τον τύπο των επιθέσεων ο κύριος σκοπός τους είναι να εκμεταλευτούν τις αδυναμίες στην υλοποίηση συγκεκριμένων πρωτοκόλλων όπως το TCP και το HTTP. Οι τρόποι αυτοί είναι πιο εύκολα υλοποιήσιμοι μέσω bots αλλά και πιο δύσκολο στο να ανιχνευτούν.


Κεφάλαιο 4
Ανίχνευση Επιθέσεων

4.Ανίχνευση Επιθέσεων

Για να γίνει σωστή αντιμετώπιση των προαναφερόμενων επιθέσεων θα πρέπει να γίνει αποτελεσματικός εντοπισμός για την προστασία του συστήματος.Ένα απο τα προβλήματα που μπορεί, μια απο τις επιθέσεις που είδαμε, είναι η δημιουργία zombie. Πιο αποτελεσματικά συστήματα αποτελούν τα συστήματα ανίχνευσης επιθέσεων βασισμένα στο δίκτυο.Τα συστήματα αυτά κάνουν και χρήση γραφιμάτων επιθέσεων για να φανερωθούν κενά του συστήματος που μπορεί να οδηγήσουν σε επικίνδυνες καταστάσεις όπως παράδειγμα ο επιτιθέμενος να έχει αποκτήσει δικαιώματα πρόσβασης σε ένα σταθμό του δικτύου. Επίσης αυτά τα γραφήματα παράγουν όλα τα πιθανά μονοπάτια που μπορεί να γίνει μια επίθεση.Μερικές προτεινόμενες μεθοδολογίες είναι η μηχανική μάθηση, ανάλυση πολυπαραγοντικής συσχέτισης,ο κατηγοριοποιητης naïve-bayes, data mining και συμπεριφορικά προφίλ.

4.1 Ανιχνευτής Επιθέσεων

Οι επιθέσεις DDoS, όπως φάνηκε και στα διαγράμματα του προηγούμενου κεφαλαίου, αυξάνονται διαρκώς τόσο λόγω της αύξησης της προσβασιμότητας στο διαδίκτυο ολοένα και περισσότερων συσκευών, όσο και λόγω της ευκολότερης πραγματοποίησης μιας τέτοιας επίθεσης. Πλέον στο διαδίκτυο υπάρχουν χρήστες οι οποίοι πληρώνονται και διαφημίζονται με σκοπό να πραγματοποιηθεί μια τέτοια επίθεση.

Ταυτόχρονα όμως έχουν αυξηθεί και οι λύσεις ενάντια σε τέτοιες επιθέσεις. Πάρα πολλές εταιρίες ασφαλείας παρέχουν πλέον λύσεις για την ανίχνευση και αντιμετώπιση τέτοιων επιθέσεων.

4.2 Μεθοδολογίες Ανίχνευσης

Για την υλοποίηση τέτοιων προγραμμάτων ανίχνευσης και αντιμετώπισης επιθέσεων έχουν αναπτυχθεί αλγόριθμοι οι οποίοι μπορούν να χωριστούν στις εξής κατηγορίες(Nikhil S, 2014):

4.2.1 Μηχανική Μάθηση

Ο τύπος αυτός ανίχνευσης επιθέσεων παρουσιάστηκε από τον N. Wattanapongsakorn και άλλους το 2012 και αποτελούταν από μηχανισμούς μηχανικής μάθησης έτσι ώστε να αναγνωρίζει και να κατηγοριοποιεί τις επιθέσεις δικτύου. Απ τους πιο γνωστούς αλγορίθμους αποτελούν τα δέντρα απόφασης.

Η υλοποίηση αυτή αποτελούταν από έναν μηχανισμό προεπεξεργασίας των πακέτων όπου εξάγει σημαντικά χαρακτηριστικά για να δημιουργήσει ένα πίνακα δεδομένων μεταξύ ενός συγκεκριμένου χρονικού ορίου, ένα κομμάτι το οποίο έκανε την κατηγοριοποίηση και έλεγχε αν τα δεδομένα απο το κομμάτι προεπεξεργασίας αποτελούν επίθεση ή κανονική κίνηση δικτύου και τέλος στο κομμάτι προστασίας οπού εκεί μπλοκάρεται η ip του αποστολέα και μπλοκάρει όλα τα πακέτα απο την συγκεκριμένη ip ή την πόρτα στην οποία γινόταν η επίθεση .(Wattanapongsakorn, 2012)

4.2.2 Ανάλυση πολυπαραγοντικής συσχέτισης

Μια άλλη προσέγγιση για την ανίχνευση των επιθέσεων είναι η υλοποίηση του Zhiyuan Tan και των συνεργατών του. Το συστημά τους αποτελούνταν απο τρία μέρη.Στο πρώτο μέρος δημιουργούταν ένα βασικό κομμάτι το οποίο περιείχε πακέτα και δεδομένα τα οποία αποτελούσαν την βάση αναφοράς του μοντέλου συσχέτισης το οποίο χρησιμοποιείται. Στο δεύτερο κομμάτι συσχετιζόταν τα εισερχόμενα δεδομένα με βάση τα δεδομένα αναφοράς και κίνησης τα οποία είχαν παραχθεί στο πρώτο μέρος. Εάν δεν υπήρχε συσχέτιση μεταξύ τους τότε αυτά προωθούταν στο τρίτο κομμάτι το οποίο ήταν το κομμάτι λήψης αποφάσεων και επιλεγόταν εάν αυτά τα δεδομένα θα θεωρηθούν κανονικά δεδομένα τα οποία έχουν ληφθεί για την εκπαίδευση του μοντέλου ή αποτελούν δεδομένα τα οποία δεν είναι αναμενόμενα και αναγνωρίζονται ως επίθεση.(Zhiyuan, 2012)

4.2.3 Κατηγοριοποιητής Naive Bayes

Ο κατηγοριοποιητής Naive Bayes αναλύει τα δεδομένα πακέτων TCP χρησιμοποιώντας δεδομένα σαν δεδομένα αναφοράς δεδομένα τα οποία έχουν προκύψει από τεχνικές data mining. Χρησιμοποιεί πιθανοτικές μεθόδους οι οποίες βασίζονται στην πιθανότητα η οποία προκύπτει από τον τύπο του Bayes. Ανάλογα με την πιθανότητα η οποία υπολογίζεται τα δεδομένα χωρίζονται σε κανονικά ή δεδομένα επίθεσης. Ακολούθως γίνεται κατηγοριοποίησή τους σε διάφορους τύπου επιθέσεων. Αυτό το μοντέλο αναπτύχθηκε από τον A. Kumaravel και τον M.Narisha και παρουσιάστηκε όπως τα προηγούμενα το 2012.(Narisha, 2012)

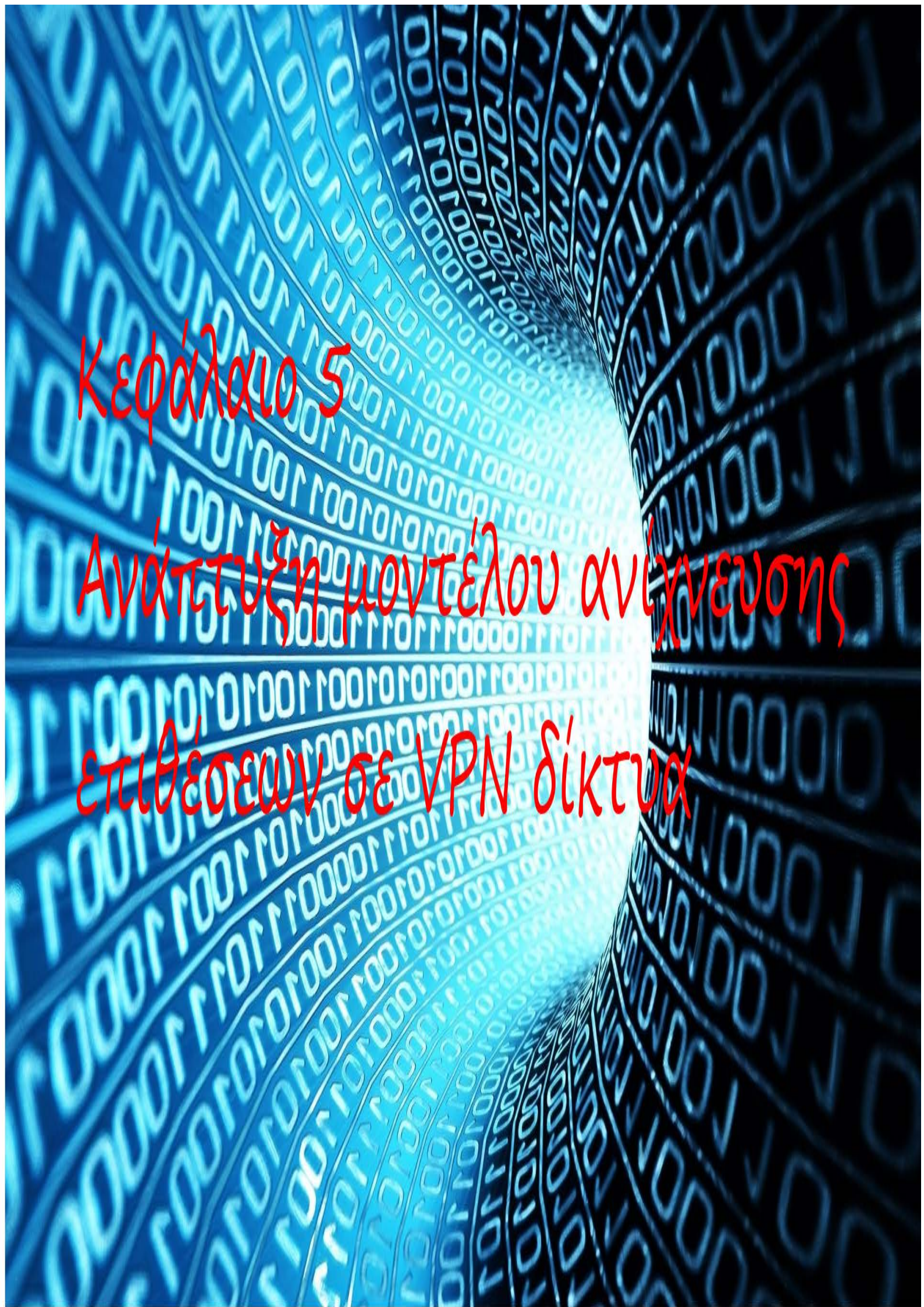
4.2.4 Συμπεριφορικά Προφίλ

Ένα ακόμη μοντέλο για την ανίχνευση επιθέσεων σε δίκτυα είναι αυτό της της δημιουργίας συμπεριφορικών προφίλ.Για κάθε χρήστη του δικτύου συλλέγονται δεδομένα για συγκεκριμένο χρονικό διάστημα από τα οποία δημιουργείται το ξεχωριστό προφίλ κάθε χρήστη. Αφού δημιουργηθεί αυτό το προφίλ τότε η κίνησή του συγκρίνεται με αυτό και εάν παρουσιαστεί κάποια ανωμαλία με βάση τον αλγόριθμο εντοπισμού ανωμαλιών ο οποίος χρησιμοποιείται σε αυτή τη κίνηση τότε αυτή καταγράφεται ώς επίθεση.Το μοντέλο αυτό παρουσιάστηκε από τον Risto Vaarandi.(Varaandi R, 2013)

4.2.5 Data Mining

Η τελευταία μέθοδος χρησιμοποιεί τις τεχνικές τις εξόρυξης δεδομένων για να κάνει αναγνώριση τους. Ο αλγόριθμος ξεκινάει συλλέγοντας δειγματα εκπαίδευσης τα οποία αποθηκεύονται σε μία βάση δεδομένων. Η διαδικασία της εξόρυξης αποτελείται από την συγκέντρωση των δεδομένων, την οπτικοποίησή τους καθώς και την αναγνωρισή των προτύπων επίθεσης. Τέλος γίνεται κατηγοριοποίηση των δεδομένων των επιθέσεων και αξιολόγηση της απόδοσής του αλγορίθμου. Για αυτη τη μέθοδο έχουν προταθεί διάφορες υλοποιήσεις με κύριες την υλοποίηση των Zhengmin Xia και άλλων(Zhengmin X, 2012), η οποία βασίζεται στον υπολογισμό της αυτοσυσχέτισης με βάση το εκθετικό του Hurst, και η άλλη των P. Jongsuebsuk, N. Wattanapongsakorn, C. Charnsripinyo(Jongsuebsuk N, 2013) η οποία χρησιμοποιεί ασαφείς γενετικούς αλγόριθμους οι οποίοι έχουν την δυνατότητα να αναγνωρίζουν και καινούριους τύπους επιθέσεων.

Επιλέξαμε να αναπτύξουμε τον ανιχνευτή μας κάνοντας χρήση της μηχανικής μάθησης και των δέντρων αποφάσεων καθώς έχει μεγάλα ποσοστά επιτυχίας να ανιχνέυσει σωστά μια επίθεση κάνοντας χρήση του μηχανισμού απόφασης σε πραγματικό χρόνο.

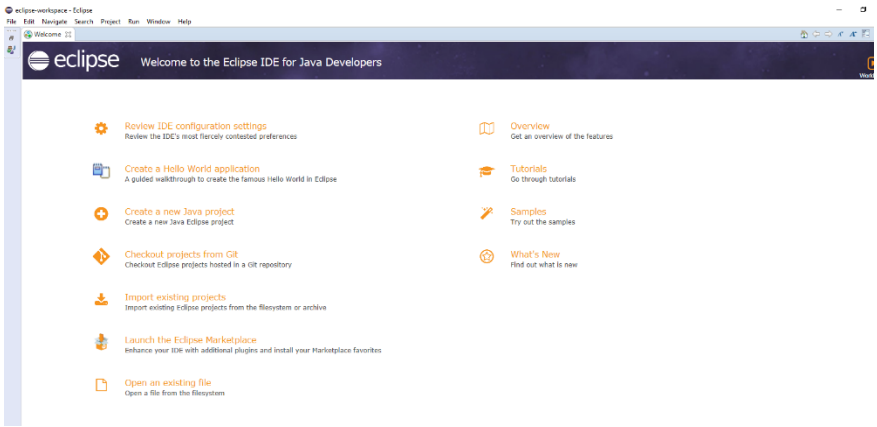

Κεφάλαιο 5
Ανάπτυξη μοντέλου ανίχνευσης επιθέσεων σε VPN δίκτυα

5. Ανάπτυξη μοντέλου Επιθέσεων σε VPN δίκτυα

Προκειμένου να εντοπίσουμε τις ευπάθειες που αναφέρθηκαν στα προηγούμενα κεφάλαια αναπτύξαμε ένα μοντέλο ανίχνευσης επιθέσεων που αναφέρεται στην παράγραφο 5.3 κάνοντας χρήση της μηχανικής μάθησης με χρήση δέντρων απόφασης που μας οδήγησε με επιτυχία στο να εντοπίσουμε διαφόρους τύπους επίθεσεων με κύριες τις επιθέσεις πλημμύρας βάση και των διαγραμμάτων στο κεφάλαιο 3 παρ 3.4. Παρατηρούμε ότι οι επιθέσεις τέτοιου τύπου είναι οι πιο συχνές VPN καθώς δεν απαιτούν υλικό μεγάλο δυνατοτήτων παρα μόνο καλή γνώση των ευπαθειών των διαφόρων πρωτοκόλλων που υλοποιούνται. Κατα την έναρξη οριοθετούμε τον αριθμό των πακέτων που απο κει και πάνω θα θεωρηθεί επίθεση και ο ανιχνευτής μας σε πραγματικό χρόνο(κάθε δευτερόλεπτο) ελέγχει την κινησή μας με σκοπό να εντοπίσει πακέτα που είναι μεγαλύτερα του ορίου. Εφόσον εντοπιστούν , ο ανιχνευτής μας μας ειδοποιεί κατάλληλα με τις απαραίτητες πληροφορίες όπως ημερομηνία,ώρα, τύπος επίθεσης και κατα το κλείσιμο ή το πέρας του ελέγχου όλης της κίνησης(100000 πακέτα) διατηρούμε log file για περαιτέρω μελέτη και αντιμετώπιση του εισβολέα.

5.1 Εισαγωγή-γενικές πληροφορίες

Για να αναπτυχθεί το πρόγραμμα που θα εντοπίζει τις επιθέσεις θεώρησαμε οτι έχει στηθεί το vpn και ως επιτιθέμενοι εκτελούμε επιθέσεις εναντίον του εξυπηρετητή που παρέχει τις υπηρεσίες του vpn. Για την ανάπτυξη του ανιχνευτή κάναμε χρήση του προγράμματος Eclipse το οποίο αποτελεί μια πλατφόρμα για ανάπτυξη web και άλλων εφαρμογών . Σαν εργαλείο ανάπτυξης εφαρμογών δεν προσφέρει επαρκή λειτουργικότητα όμως, αυτό που το κάνει να ξεχωρίζει είναι η χρήση διάφορων προσθέτων (plug-in). Σαν πρόγραμμα προσφέρει ένα απλό σχετικά γραφικό περιβάλλον και έχει σχεδιαστεί για να εκτελείται σε πολλά λειτουργικά συστήματα χρησιμοποιώντας τα plugin. Η έκδοση που χρησιμοποίηθηκε είναι η έκδοση Eclipse Oxygen 2 το οποίο είναι διαθέσιμο στον ιστότοπο http://www.eclipse.org/downloads/ και έγινε φόρτωση του πακέτου Eclipse IDE for Java Developers καθώς το πρόγραμμα έχει αναπτυχθεί με χρήση java.

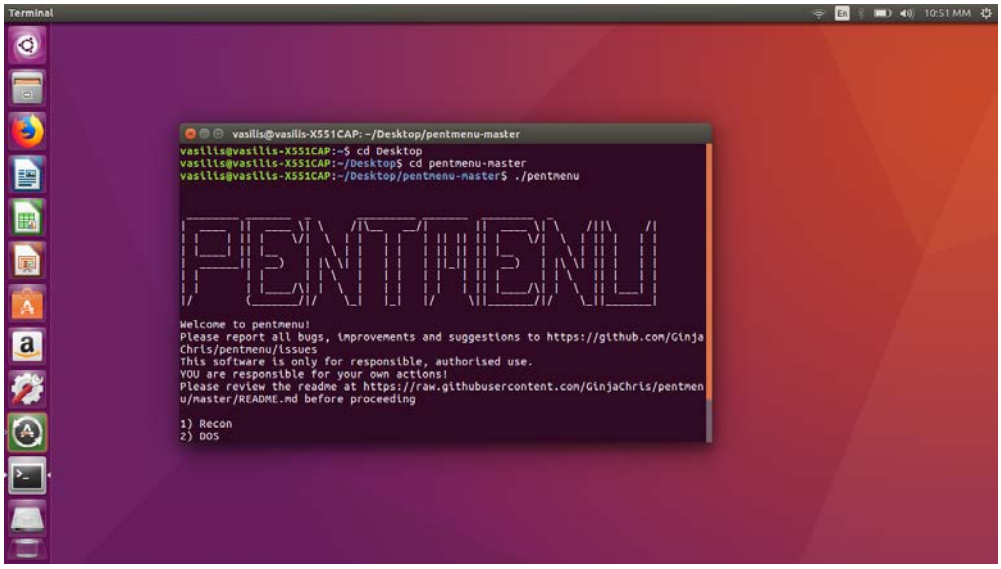


5.1 Γραφικό περιβάλλον του Eclipse

Επίσης, για να συλληφθεί η κίνηση στο δίκτυο είναι απαραίτητη η εγκατάσταση και ενός άλλου προγράμματος του winpcap προκειμένου να συλλαμβάνονται και φιλτράρονται τα πακέτα που μεταδίδονται.

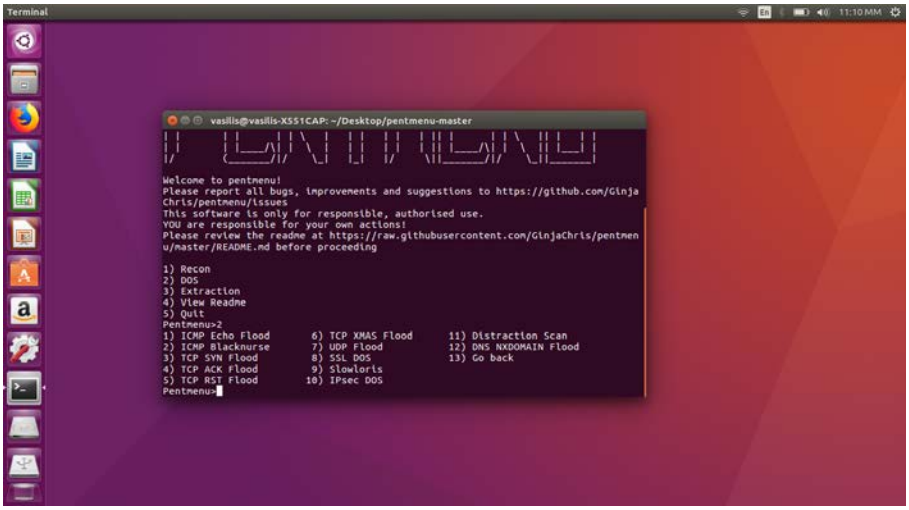
Επιπλέον, την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη jnetpcap η οποία αποτελεί μια ανοιχτού κώδικα βιβλιοθήκη η οποία παρέχει την διεπαφή για την σύλληψη και καταγραφή των πακέτων τα οποία περνάνε από μια συσκευή δικτύου.

Επιπροσθέτως το σενάριο για την λειτουργία του προγράμματος είναι οτι ένας υπολογιστής/χρήστης πραγματοποιεί την επίθεση στέλνοντας πακέτα μέσω του linux. Απο την μεριά του επιτιθέμενου εκτέλεσαμε ένα πρόγραμμα το pentmenu (https://github.com/GinjaChris/pentmenu) το οποίο δημιουργεί τις επιθέσεις.



5.2 Βήματα εκτέλεσης για το πρόγραμμα pentmenu

Στην συνέχεια στο μενού που μας εμφανίζεται επιλέγουμε την επιλογή 2 και λαμβάνουμε μια λίστα των επιθέσεων που μπορούμε να εκτελέσουμε. Το μοντέλο που αναπτύχθηκε εντοπίζει τρείς(3) τύπους επιθέσεων UDP flood, ICMP flood και τέλος TCP SYN flood .



5.3 Το μενού του pentmenu

5.2 Υλοποίηση επιθέσεων

Για την ανάπτυξη του μοντέλου θεώρησαμε ότι το σημαντικότερο σε έναν διακομιστή υπηρεσιών VPN αποτελεί η αξιοπιστία του και η δυνατότητα να αναγνωρίζονται επιθέσεις οι οποίες θα κάνουν τον διακομιστή να μην μπορεί να ανταποκριθεί σε αιτήματα των χρηστών οι οποίοι είναι συνδεμένοι με αυτόν. Ακόμα ο διακομιστής είναι το πρώτο σημείο στο οποίο μπορεί να γίνει η επίθεση καθώς εαν αποκοπεί η σύνδεση του διακομιστή με το εξωτερικό διαδίκτυο τότε όλοι οι χρήστες στους οποίους παρέχει τις υπηρεσίες του δεν θα μπορούν να τις χρησιμοποιήσουν.

Για αυτό το λόγο επικεντρώθηκαμε στις ογκομετρικές επιθέσεις οι οποίες στοχεύουν τον διακομιστή από εξωτερικές πηγές του διαδικτύου. Τέτοιες επιθέσεις μπορούν να καταστήσουν τον διακομιστή ανίκανο να ανταποκριθεί σε οποιαδήποτε υπηρεσία ζητηθεί από τους χρήστες των υπηρεσιών VPN που παρέχει.

Τέτοιες επιθέσεις είναι οι επιθέσεις τύπου πλημμύρας (flood). Επιλέξαμε να αναλύσουμε αυτές τις επιθέσεις διότι αυτές είναι οι πιο συνήθεις επιθέσεις σε VPN και οι οποίες συμβαίνουν σε παγκόσμιο επίπεδο σύμφωνα και με τα διαγράμματα της παραγράφου 3.4.

Για τα πακέτα ICMP βλέπει εάν αυτά τα πακέτα δεν είναι πακέτα τύπου reply, δηλαδή πακέτα τα οποία αποτελούν απάντηση σε κάποια πληροφορία που ζητήθηκε από τον διακομιστή και τότε αυξάνεται ο μετρητής τους.

Για τα πακέτα UDP ο μετρητής αυξάνεται σε κάθε πακέτο καθώς τα UDP πακέτα δεν χρειάζονται κάποια απάντηση ή έχουν κάποια διαδικασία αρχικοποίησης και ταυτοποίησης.

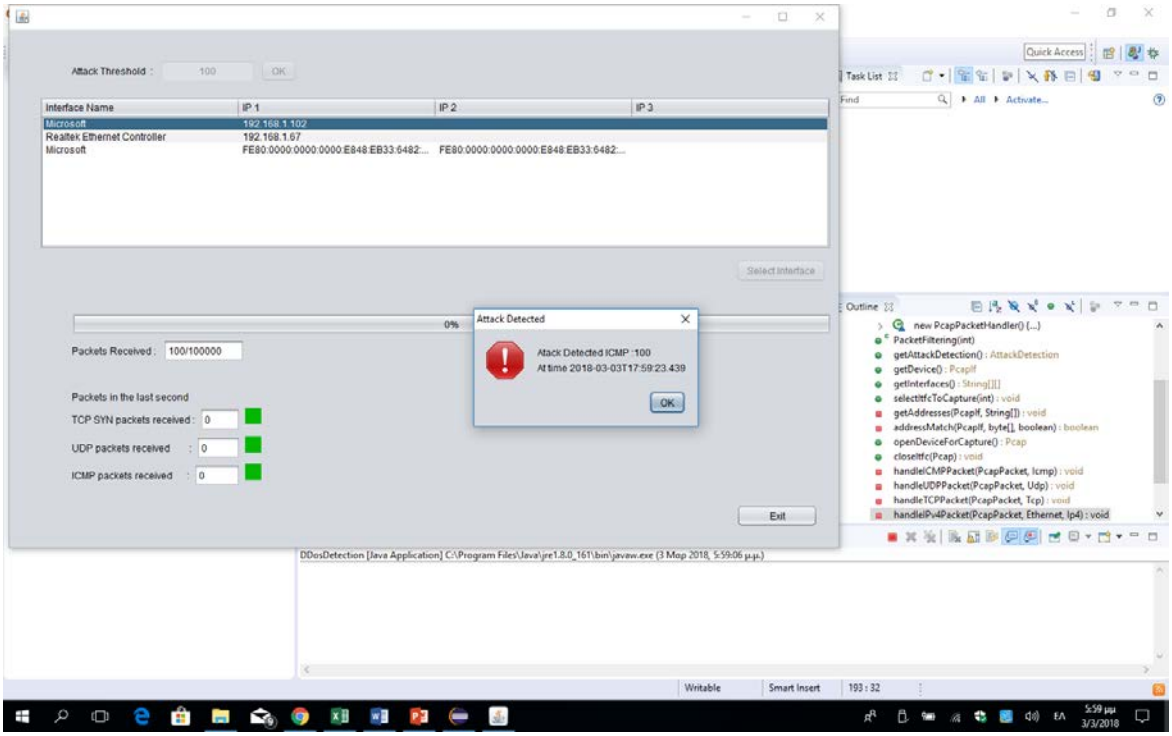
Για τα πακέτα TCP εξετάζουμε μόνο την περίπτωση στην οποία λαμβάνονται πολλά πακέτα TCP SYN. Το TCP SYN flood είναι ένας από τους δημοφιλέστερους τύπους επίθεσης χρησιμοποιώντας πακέτα TCP.

Έτσι, κατα την εκτέλεση του προγράμματος επιλέγουμε τον τύπο της επίθεσης και πατάμε enter ενώ στην συνέχεια μας εμφανίζει σε ποία ip επιθυμούμε να επιτεθούμε και επιλέγουμε την ip που μας έχει εμφανίσει το κυρίως πρόγραμμα που θα δούμε στην συνέχεια. Ως source ip ελέγχουμε την ip που έχουμε πάρει απο το δίκτυο στην εικόνα έχουμε την ip 192.168.1.64.Πατώντας το enter, ξεκινάει η επίθεση που στην συνέχεια θα δούμε πως ο ανιχνευτής μας εντοπίζει την επίθεση.

5.4 Εντολές στο pentmenu

5.3 Λειτουργία ανιχνευτή επιθέσεων

Υλοποιήθηκε ένα γραφικό περιβάλλον χρήστη το οποίο προσφέρει στον χρήστη την επιλογή του να θέσει το επίπεδο του αριθμού πακέτων το οποίο θα θεωρηθεί επίθεση. Εφ'οσον θέσουμε τον αριθμό των πακέτων που θα θεωρηθεί επίθεση αμέσως μετά μας εμφανίζεται απο ποιά συσκευή δικτύου θα γίνει η σύλληψη των πακέτων.Στην περιπτωσή μας, επειδή βρισκόμαστε σε λαπτοπ, μας εμφανίζει και την ασύρματη κάρτα δικτύου και την ενσύρματη. Επιλέγουμε ποιά κάρτα θέλουμε και επιλέγουμε "select interface" και ξεκινάει η καταγραφή των πακέτων. Στο γραφικό αυτό περιβάλλον εμφανίζεται επίσης μια μπάρα η οποία υπάρχει εκεί έτσι ώστε να δείχνει τι ποσοστό από τον αριθμό των πακέτων που θα συλλάβει, και έχει τεθεί ως 100000, έχουν συλληφθεί, καθώς και το πόσα πακέτα έχουν ληφθεί από το κάθε πρωτόκολλο μέσα στο προηγούμενο δευτερόλεπτο. Τέλος δίπλα από αυτές τις τιμές υπάρχει μια ένδειξη η οποία δείχνει εάν έχει ανιχνευθεί επίθεση για το κάθε πρωτόκολλο ή όχι.



5.5 Γραφικό περιβάλλον ανιχνευτή επιθέσεων

Ο τρόπος με τον οποίο επέλεξαμε να ανιχνεύσουμε αυτές τις επιθέσεις είναι με χρήση ενός δέντρου αποφάσεων με την χρήση μηχανικής μάθησης. Το πρόγραμμα μαθαίνει για τον κάθε τύπο πρωτοκόλλου την συμπεριφορά του και εντοπίζει ανάλογα τις επιθέσεις.

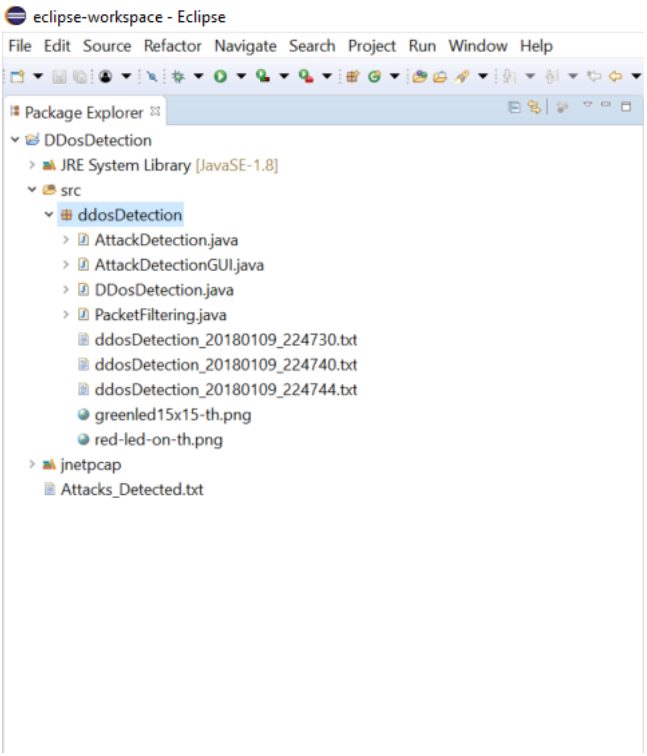
Για την υλοποίηση αυτή παρατηρούμε την κίνηση η οποία υπάρχει για καθέναν από αυτούς τους τύπους πακέτων και κατόπιν εάν για παραπάνω από 3 δευτερόλεπτα υπάρχει κίνηση η οποία ξεπερνάει το 500% της κίνησης η οποία παρατηρείται ή η στιγμιαία κίνηση για το πρωτόκολλο είναι μεγαλύτερη από το επίπεδο αριθμού πακέτων που θέσει ο χρήστης για κάθε τύπο πρωτοκόλλου τότε αυτό αναγνωρίζεται ως επίθεση και παρουσιάζεται στον χρήστη του διακομιστή. Επιπλέον καταγράφεται αυτό σε ένα αρχείο με την ώρα και τον τύπο επίθεσης.

Πέρα από την ανίχνευση με τον τύπο πακέτου υλοποιήθηκε αντίστοιχος μηχανισμός ο οποίος αναγνωρίζει και καταγράφει τις διευθύνσεις IP από τις οποίες προέρχεται αυτή η κίνηση και εάν υπάρχει αυξημένη κίνηση από κάποια συγκεκριμένη διεύθυνση αυτή καταγράφεται στο ίδιο αρχείο ανίχνευσης επιθέσεων.

5.4 Περιγραφή κώδικα

Ο κώδικας χωρίζεται στις εξής κλάσεις(Ο αντιστοιχος κώδικας βρίσκεται στο παράρτημα Α)

- DDoSDetection
- PacketFiltering
- AttackDetection



5.6 Οι κλάσεις του ανιχνευτή επιθέσεων

A) AttackDetection

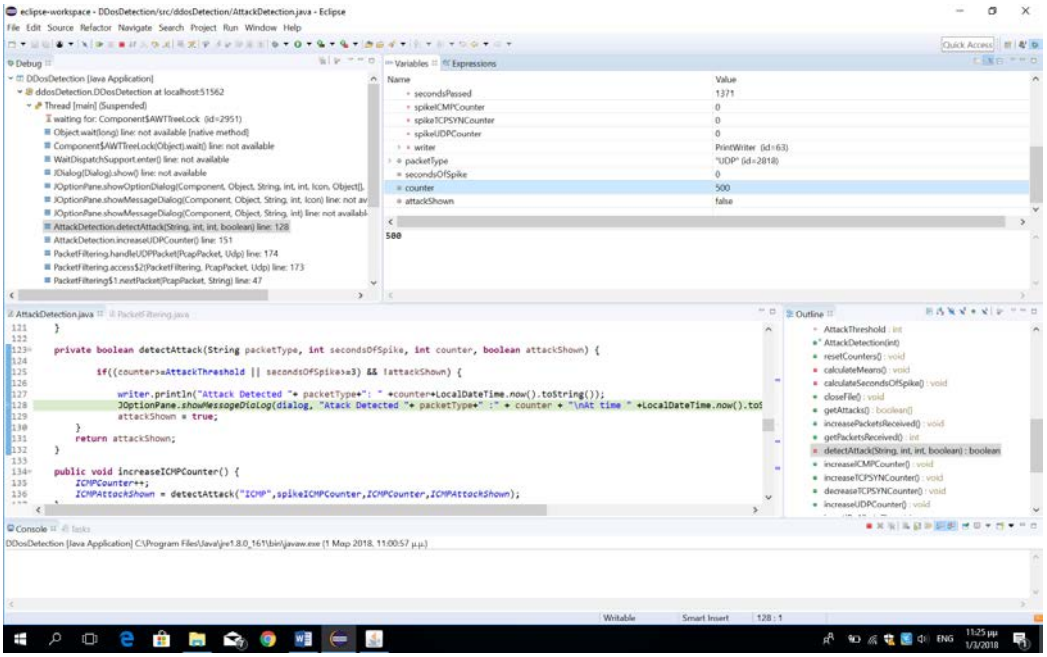
Η κλαση AttackDetection GUI υλοποιεί την διεπαφή μέ τον χρήστη. Αρχικοποιεί και θέτει όλες τις απαραίτητες μεταβλητές για την λειτουργία του γραφικού περιβάλλοντος καθώς και για τις αλλαγές οι οποίες γίνονται σε αυτό.

Επίσης, η AttackDetection περιέχει μέσα της τις μεταβλητές οι οποίες ελέγχονται για να μπορούμε να ανιχνεύσουμε κάποια επίθεση. Περιέχει έναν μετρητή για κάθε τύπο πακέτου που δεχτήκαμε (ICMP,UDP,TCP) καθώς και τις μεθόδους αύξησης των μετρητών αυτών.Επιπλέον περιέχει και έναν πίνακα κατακερματισμού Hashtable 500 θέσεων για να αποθηκεύει τις διευθύνσεις IP από τις οποίες έχουμε δεχτεί πακέτα έτσι ώστε αν έχουμε δεχτεί πολλά πακέτα από κάποια IP να μπορεί να το ανιχνεύσει. Η συνάρτηση κατακερματισμού που χρησιμοποιείται για τον πίνακα είναι αυτή που υλοποιείται από την Java αφού εμείς μετατρέψουμε την κάθε διεύθυνση IP σε συμβολοσειρά(String).

Κάθε συνάρτηση η οποία αυξάνει το μέγεθος του πακέτου καλεί και την συνάρτηση detectAttack η οποία ελέγχει αν τα πακέτα τα οποία έχουν μετρηθεί είναι μεγαλύτερα από το όριο το οποίο έδωσε ο χρήστης και αν είναι τυπώνουν μια φορά το μήνυμα και θέτουν την μεταβλητή <τύπος πακέτου>shown ίση με true έτσι ώστε στο ίδιο χρονικό διάστημα να μην ξαναβγεί το παράθυρο ότι

έγινε σε εκείνο το δευτερόλεπτο επίθεση.Στα πακέτα TCP όταν έρχεται ένα ACK πακέτο σαν απάντηση σε ένα SYN/ACK τότε ο μετρητής μειώνεται.

Επιπλεόν αρχικοποιεί και γράφει σε ένα αρχείο κάθε επίθεση η οποία έχει γίνει το πρωτόκολλο και την ώρα την οποία έγινε.



5.7 Πακέτο udp με counter(μεγεθος) πάνω απο το threshold θεωρείται επίθεση

B) PacketFiltering

Η κλάση PacketFiltering αποτελεί την κύρια κλάση του κώδικα. Περιέχει την σύλληψη των πακετων και την ανάλυσή τους. Περιέχει τις μεταβλητές της αντικειμενοποίησης(instantiation) της κλασης attackDetection καθώς και την συσκευή απο την οποία γίνεται η σύλληψη των πακέτων.

Τέλος περιέχει την συνάρτηση jpacketHandler η οποία καλείται κάθε φορά που “πιάνεται” ένα πακέτο. Η συνάρτηση αυτή για κάθε τύπο κεφαλίδας ο οποίος μπορεί να υπάρχει στο πακέτο πάει και στην αντίστοιχη συνάρτηση που επεξεργάζεται αυτά τα δεδομένα. Επιπλέον κάθε φορά καλέι την increasePacketsReceived η οποία αυξάνει κάθε φορά τον αριθμό των πακέτων που έχουν ληφθεί κατα 1.

Ο constructor της κλάσης αυτής αρχικοποιεί μόνο το αντικείμενο της κλάσης attackDetection. Η συνάρτηση getAttackthreshold λαμβάνει από τον χρήστη τον αριθμό των πακέτων που πάνω από αυτό το όριο θα θεωρηθεί επίθεση.

Έπειτα υπάρχουν οι συναρτήσεις που επιστρέφουν τα αντικείμενα ad και της συσκευής. Ακολούθως η συνάρτηση selectItfcToCapture. Η συνάρτηση καλεί την συνάρτηση findAllDevs της βιβλιοθήκης jnetpcap η οποία επστρέφει στην μεταβλητή allDevs την οποία περνάμε σαν όρισμα την λίστα όλων των πιθανών συσκευών δικτύου από τις οποίες μπορούμε να λάβουμε πακέτα. Ακολούθως ελέγχει την τιμή επιστροφής της συνάρτησης και τυπώνει όλες τις συσκευές αυτές καθώς και τις διευθύνσεις IPv4 και IPv6 τις οποίες έχουν. Έπειτα παίρνει σαν είσοδο από τον χρήστη τον αριθμό συσκευής που θέλει να κάνει capture και βάζει την μεταβλητή device την συσκευή αυτή.

Η συνάρτηση addressMatch χρησιμοποιείται από τις συναρτήσεις που ελέγχουν για τα πακέτα IPv4 και IPv6 για να ελεγχθεί αν το πακέτο είναι εισερχόμενο ή εξερχόμενο. Η συνάρτηση αυτή

παίρνει και ένα boolean όρισμα το οποίο είναι αν η διεύθυνση που θέλουμε να ελέγξουμε είναι IPv4 ή IPv6.

Επόμενη είναι η συνάρτηση openDeviceForCapture(). Η συνάρτηση αυτή ουσιαστικά ανοίγει τη συσκευή έτσι ώστε να μπορούμε να διαβάσουμε από αυτήν. Θέτει τις απαραίτητες μεταβλητές snaplen, flags,timeout έτσι ώστε να περαστούν στην συνάρτηση της βιβλιοθήκης jnetpcap openLive η οποία ανοίγει την συσκευή για να μπορούμε να πάρουμε πακέτα από αυτήν. Μετά με την setDirection θέτουμε ότι θα λάβει μόνο τα εισερχόμενα πακέτα και επιστρέφουμε την μεταβλητή τύπου Pcap η οποία είναι απαραίτητη για να μπορούμε να πάρουμε τα πακέτα.

Η closeItfc κλείνει την συσκευή την οποία έχουμε ανοίξει έτσι ώστε να μην υπάρχουν διαρροές μνήμης.

Ακολούθως είναι οι συναρτήσεις για κάθε τύπο κεφαλίδας:

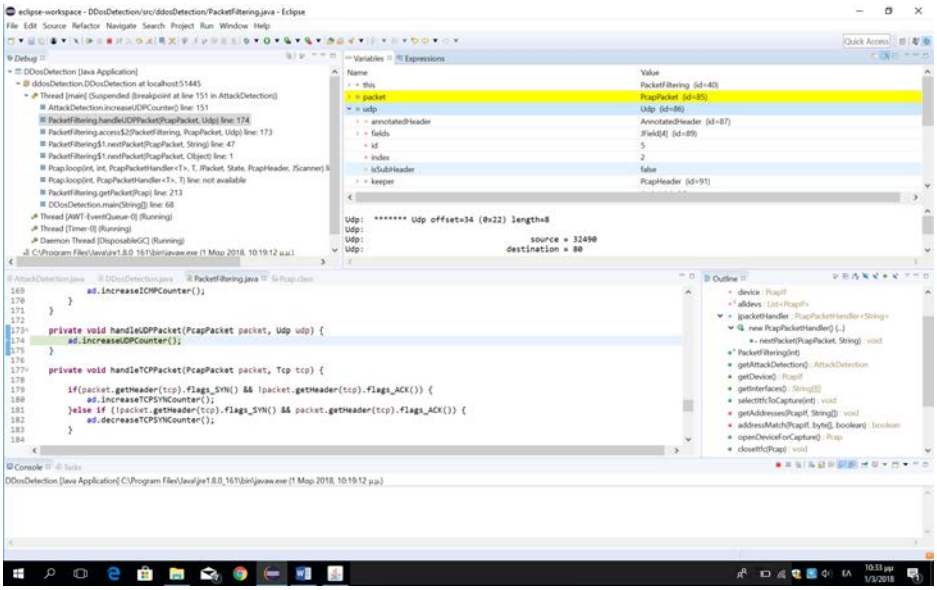
handleICMPPacket : εαν δεν είναι τύπου reply το πακέτο τότε θα είναι request οπότε αυξάνεται ο μετρητής κατα 1. (ICMP flood)

handleUDPPacket :ο μετρητής αυξάνεται κάθε φορά κατα 1 (UDP flood)

handleTCPPacket : ο μετρητής αυξάνεται κατα ένα για κάθε syn/ack πακέτο το οποίο λαμβάνουμε και μειώνεται κατα 1 για κάθε ack πακέτο που λαμβάνουμε (TCP SYN flood)

handleIPv4Packet/handleIPv6Packet : ελέγχεται αν η διεύθυνση πηγής του πακέτου είναι αυτή της συσκευής που έχουμε και εαν δεν είναι τότε αυξάνει τον μετρητή κατα 1.

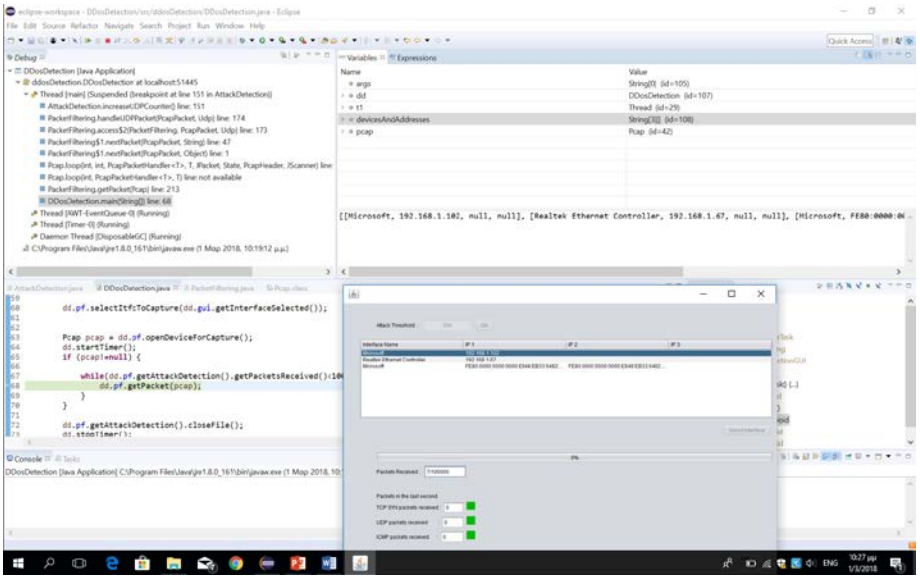
Τέλος η συνάρτηση getPacket καλεί την συνάρτηση βιβλιοθήκης loop η οποία δέχεται και επεξεργάζεται 1 πακέτο σύμφωνα με την συνάρτηση jnetpcaphandler και τον χρήστη jNetPcap. Εαν αλλάξουμε την πρώτη μεταβλητή τότε επεξεργαζόμαστε περισσότερα πακέτα αλλά μπλοκάρουμε όλο το νήμα.



5.8 Παράδειγμα χρήσης συνάρτησης handleUDPPacket

c) DDosDetection

Οι κλάση DDoSDetection αποτελεί την κύρια κλάση η οποία τρέχει όλο το πρόγραμμα. Υλοποιεί δύο αντικείμενα, ένα της κλάσης packetFiltering και ένα του GUI. Επιπλέον υλοποιεί έναν μετρητή ο οποίος μετράει κάθε φορά ένα δευτερόλεπτο.Ορίζεται η συνάρτηση - μεταβλητή timertask η οποία ορίζει το τι θα κάνει ο μετρητής κάθε φορά που περνάει ο χρόνος του. Όταν περάσει το δευτερόλεπτο που μετράει ο μετρητής τότε γίνεται ανανέωση των χρωμάτων στο γραφικό περιβάλλον που δείχνουν αν έχει γίνει επίθεση, παίρνει τους μετρητές έτσι ώστε να ανανεώσει τις τιμές τους στο γραφικό περιβάλλον και ανανεώνει τους μετρητές στο γραφικό περιβάλλον. Στην αρχή η main αρχικοποιεί τα δύο αντικείμενα με την κλήση της new και μετά περιμένει μέχρι να εισάγει ο χρήστης την τιμή στο γραφικό περιβάλλον. Ακολούθως γεμίζει τον πίνακα με τις συσκευές και μετά περιμένει να επιλέξει την συσκευή ο χρήστης από την οποία θα παίρνει τα πακέτα. Μετά ανοίγει την συσκευή και μπαίνει σε έναν ατέρμονο βρόγχο αφού ξεκινήσει τον μετρητή ο οποίος τρέχει μέχρι να πάρει 100000 πακέτα ή να κλείσει το γραφικό περιβάλλον. Όταν τελειώσει ο βρόγχος τότε κλείνει το αρχείο εγγραφής, σταματάει τον μετρητή και κλείνει την συσκευή πριν τελειώσει.



5.9 Gui και συσκευές δικτύου

Εκτελέσαμε 100 επιθέσεις για 3 διαφορετικά σενάρια για διάφορα threshold.Ύστερα πήραμε δείγμα απ το log file που διατηρεί ο ανιχνευτής μας και ξεκινώντας απο τα 500 πακέτα και αυξάνοντας κάθε φορά των αριθμό των πακέτων κατα 500 πήραμε μετρήσεις και για το όριο των 1000 και 1500 πακέτων.

Type of attack & threshold	response time
UDP-500	1.291
UDP-500	1.806
UDP-500	1.182
UDP-500	1.755
UDP-500	
TCP-500	1.215
TCP-500	1.302

TCP-500	1.941
TCP-500	1.682
TCP-500	
ICMP-500	1.48
ICMP-500	1.401
ICMP-500	1.269
ICMP-500	1.702
ICMP-500	
UDP-1000	2.817
UDP-1000	2.283
UDP-1000	2.782
UDP-1000	2.936
UDP-1000	
TCP-1000	2.367
TCP-1000	2.371
TCP-1000	2.836
TCP-1000	2.511
TCP-1000	
ICMP-1000	2.241
ICMP-1000	2.36
ICMP-1000	2.577
ICMP-1000	2.829
ICMP-1000	
UDP-1500	3.969
UDP-1500	3.02
UDP-1500	3.914
UDP-1500	3.062
UDP-1500	
TCP-1500	4
TCP-1500	3.56
TCP-1500	3.039
TCP-1500	3.02
TCP-1500	
ICMP-1500	3.088
ICMP-1500	3.955
ICMP-1500	3.98
ICMP-1500	3.961
ICMP-1500	

Πίνακας 1 Επιθέσεις και χρόνος απόκρισης

Type of attack and Threshold	Average Response time
UDP-500	1
TCP-500	1,228
ICMP-500	1,1704
UDP-1000	2,1636
TCP-1000	2,017
ICMP-1000	2,0014
UDP-1500	2,793
TCP-1500	2,7238
ICMP-1500	2,9968

Πίνακας 2 Μέσος χρόνος απόκρισης

Κατά την εκτέλεση των επιθέσεων παρατηρούμε για μικρό όριο αριθμού πακέτων ο ανιχνευτής εντοπίζει άμεσα (κοντά στο 1 δευτερόλεπτο) τις επιθέσεις ενώ όσο μεγαλώνουμε το όριο ο χρόνος αυτός αυτός είναι λίγο μεγαλύτερος (κοντά στα 2 και 3 δευτερόλεπτα αντίστοιχα).

5.5 Συμπεράσματα- προτάσεις

Αυτό που συμπεραίνουμε ύστερα και απο την ανάπτυξη της εφαρμογής είναι οτι όσο εξελίσσονται οι επιθέσεις τόσο απαιτητικός γίνεται και ο εντοπισμός τους. Για να αναγνωριστούν και να αντιμετωπιστούν οι επιθέσεις αποτελεσματικά θα πρέπει να είμαστε σε θέση να γνωρίζουμε όσο γίνεται πιο λεπτομερώς τον τρόπο που θα στήσουμε το δικτυό μας και κατ ‘ επέκταση το VPN μας . Θα πρέπει επίσης να γίνει σωστή επιλογή των μηχανισμών ασφαλείας και των πρωτοκόλλων τους τα οποία πρέπει να ενημερώνονται τακτικά και να υπάρχει τακτική παρακολούθηση για τυχόν αλλαγές ως προς την ροή των πακέτων μεταξύ των χρηστών. Διότι, μια επίθεση σε ένα απλό οικιακό χρήστη μπορεί να του προκαλέσει μια ζημία που ενδεχομένως δεν θα αποβεί μοιραία (να διαγραφτεί ενα αρχείο,να γίνει αργός ο υπολογιστής του) όμως στην περίπτωση μιας εταιρίας μπορεί να κοστίσει απο μερικές χιλιάδες ευρώ εώς εκατομμύρια ανάλογα το μέγεθος και την εκταση της ζημιάς. Γι αυτό μια λύση για να χτιστεί ιδανικά και με ασφάλεια ένα δίκτυο, όπως το VPN, θα πρέπει να «χτιστεί» απο μέσα προς τα έξω. Δηλαδή ανάλογα τις ανάγκες που έχει η επιχείρηση να χτίζονται οι κανόνες του δικτύου όπως τα δικαιώματα πρόσβασης , ο διαμοιρασμός των πόρων του δικτύου , οι κανόνες ασφαλείας. Επίσης συμπεραίνουμε οτι τα πρωτόκολλα ασφαλείας παίζουν πολύ μεγάλο ρόλο στην επικοινωνία καθώς μη ενημερωσή τους ή σωστή υλοποιησή τους αυξάνει το κίνδυνο να γίνει στόχος επίθεσης το δικτύο μας. Σχετικά με την εφαρμογή, όπως αναφέρθηκε και εισαγωγικά , εντοπίζει τρείς τύπους επιθέσεων. Θα μπορούσε λοιπόν μελλοντικά να εντοπίζει κι άλλους τύπους επιθέσεων και να αναπτυχθεί και μηχανισμός να μπλοκάρει τέτοιες επιθέσεις. Θα μπορούσε επίσης σε μια εταιρία να υπάρχει κάποια καρτέλα η οποία θα κάνει παρακολούθηση της κίνησης όλων των χρηστών και όταν θα βλέπει κάποια αλλαγή στην δρομολόγηση των πακέτων να δίνει κάποια ειδοποίηση για δράση.

Πηγές

1. HKSAR (2007), *VPN Security* ,Διαθέσιμο στον ιστότοπο https://www.infosec.gov.hk/english/technical/files/vpn.pdf Ανακτήθηκε στις 18/4//2017
2. OnlineLMS (2017),*Data integrity/hash* Διαθέσιμο στον ιστότοπο http://www.onlinelms.org/mod/book/view.php?id=24&chapterid=411. Ανακτήθηκε στις 28/4/2017
3. Cisco systems(2000),*Networkers*,Διαθέσιμο στον ιστότοπο https://www.cisco.com/networkers/nw00/pres/2608.pdf. Ανακτήθηκε στις 18/4/2017
4. IETF tools (2017), *Rfc files*, Διαθέσιμο στον ιστοτοπο https://tools.ietf.org/html/. Ανακτήθηκε στις 10/5/2017
5. Wikipedia (2017), *VPN blocking*. Διαθέσιμο στον ιστότοπο https://en.wikipedia.org/wiki/VPN_blocking . Ανακτήθηκε στις 5/6/2017
6. Netgear Inc (2005), *Virtual Private Networking Basics.* Διαθέσιμο στον ιστότοπο http://documentation.netgear.com/reference/fra/vpn/pdfs/FullManual.pdf Ανακτήθηκε στις 25/6/2017
7. Ylonen T. (2017), *SSH Protocol*, Διαθέσιμο στον ιστότοπο https://www.ssh.com/ssh/protocol/ . Ανακτήθηκε στις 25/6/2017
8. Microsoft Technet (2001), *Virtual Private Networking: An Overview*.Διαθέσιμο στον ιστοτοπο https://technet.microsoft.com/en-us/library/bb742566.aspx . Ανακτήθηκε στις 1/7/2017
9. Techopedia (2017), *Password Authentication Protocol(PAP)*. Διαθέσιμο στον ιστότοπο https://www.techopedia.com/definition/4043/password-authentication-protocol-pap . Ανακτήθηκε στις 5/7/2017
10. Rousse M (2017), *Advanced Encryption Security(AES).* Διαθέσιμο στον ιστότοπο http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard . Ανακτήθηκε στις 24/4/2017
11. Cetinkaya k. Egemen, *Data Integrity Algorithms (*Πανεπιστημιακές σημειώσεις),ΗΠΑ, χειμερινό εξάμηνο 2016, Διαθέσιμο στον ιστότοπο http://web.mst.edu/~cetinkayae/teaching/CPE5420Fall2016/CPE5420-F2016-data-integrity-algorithms.pdf . Ανακτήθηκε στις 30/4/2017
12. Gilbert C(2016), *Point-to-Point Access: PPP*. Διαθέσιμο στον ιστότοπο http://slideplayer.com/slide/8563797/ .Ανακτήθηκε στις 15/8/2017
13. Scheiner B, Mudge, Wagner(1999), *Cryptanalysis of Microsoft PPTP authentication extension (MS-CHAPv2)*, Διαθέσιμο στον ιστότοπο https://www.schneier.com/academic/paperfiles/paper-pptpv2.pdf. Ανακτήθηκε στις 18/8/2017
14. IETF.org, *Protected EAP Protocol (PEAP) Version 2*, Διαθέσιμο στον ιστότοπο https://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-10. Ανακτήθηκε στις 20/8/2017
15. Layden J (2016),*90% of SSL VPN are 'hopelessly insecure'*Διαθέσιμο στον ιστότοπο https://www.theregister.co.uk/2016/02/26/ssl_vpns_survey/ .Ανακτήθηκε στις 20/8/2017
16. Vasile C. Perta, Marco V. Barbera, Gareth Tyson, Hamed Haddadi, and Alessandro Mei(2015) , *A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients*, Διαθέσιμο στον ιστότοπο https://www.degruyter.com/downloadpdf/j/popets.2015.1.issue-1/popets-2015-0006/popets-2015-0006.pdf. Ανακτήθηκε στις 30/8/2017
17. Akamai (2017), *VPNs Are Not as Secure as You Think*, Διαθέσιμο στον ιστότοπο https://www.akamai.com/uk/en/multimedia/documents/white-paper/akamai-vpns-arent-as-secure-as-you-think-whitepaper.pdf . Ανακτληθηκε στις 30/8/2017
18. Wikipedia (2017), *Denial of Service Attacks*,Διαθέσιμο στον ιστότοπο https://www.wikiepedia.org/en/Denial-of-service_attack . Ανακτήθηκε στις 1/9/2017
19. Imperva Inc(2017) , *DDoS attacks* . Διαθέσιμο στον ιστότοπο https://www.incapsula.com/ddos/ddos-attacks/. Ανακτήθηκε στις 3/9/2017
20. Imperva Inc (2017), Attack Glossary(UDP Flood and others) .Διαθέσιμο στον ιστοτοπο https://www.incapsula.com/ddos/attack-glossary/udp-flood.html. Ανακτήθηκε στις 20/9/2017

21. Wikiwand (2017), *Smurf attack.* Διαθέσιμο στον ιστότοπο https://www.wikiwand.com/en/Smurf_attack . Ανακτήθηκε στις 25/9/2017
22. Fulton S(2015), Top 10 DNS attacks likely to infiltrate your network. Διαθέσιμο στον ιστότοπο https://www.networkworld.com/article/2886283/security0/top-10-dns-attacks-likely-to-infiltrate-your-network.html#slide11. Ανακτήθηκε στις 30/9/2017
23. National Institute of Standards and Technology-NIST (2008), *National Vulnerability Database.*Διαθέσιμο στον ιστότοπο https://nvd.nist.gov/vuln/detail/CVE-2008-1447#vulnDescriptionTitle. Ανακτήθηκε στις 5/10/2017
24. Woolf N (2016), *DDoS attack that disrupted internet was largest of its kind in history,*The Guardian, Διαθέσιμο στον ιστότοπο https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet. Ανακτήθηκε στις 7/10/2017
25. Alexander Khalimonenko, Oleg Kupreev, Kirill Ilganaev, *DDoS attacks in Q3 2017,Kaspersky Lab (RU).* Διαθέσιμο στον ιστότοπο https://securelist.com/ddos-attacks-in-q3-2017/83041/ . Ανακτήθηκε στις 27/11/2017
26. Akamai (2017), *Real-Time Web Monitor*, Διαθέσιμο στον ιστότοπο https://www.akamai.com/us/en/solutions/intelligent-platform/visualizing-akamai/real-time-web-monitor.jsp. Ανακτήθηκε στις 27/11/2017
27. Madhavi R.P(2015), *International Journal of Advance Research in Computer Science and Management Studies Network Intrusion.*. Διαθέσιμο στον ιστότοπο https://www.researchgate.net/publication/292845352_International_Journal_of_Advance_Research_in_Computer_Science_and_Management_Studies_Network_Intrusion_Detection_and_Countermeasure_Selection_in_Virtual_Private_Network_Systems. Ανακτήθηκε στις 27/11/2017
28. Nikhil S. Mangrulkar (2014) , Arvind R. Bhagat Patil , Abhijit S. Pande, *Network Attacks and Their Detection Mechanisms: A Review* Διαθέσιμο στον ιστότοπο https://pdfs.semanticscholar.org/2f95/27037224cfef2134256d86db874449ff2421.pdf. Ανακτήθηκε στις 30/11/2017.
29. N. Wattanapongsakorn, S. Srakaew, E. Wonghirunsombat, C. Sribavonmongkol, T. Junhom, P. Jongsubsook, C. Charnsripinyo (2012), *"A Practical Network-based Intrusion Detection and Prevention System"*,Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/document/6295977/?part=1 . Ανακτήθηκε στις 1/12/2017
30. Zhiyuan Tan, Aruna Jamdagni, Xiangjian He, Priyadarsi Nanda and Ren Ping Liu(2012), *"Triangle-Area-Based Multivariate Correlation Analysis for Effective Denial-of-Service Attack Detection.* Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/document/6295955/. Ανακτήθηκε στις 5/12/2017
31. A. Kumaravel, M.Niraisha, *"Multi-Classification Approach for Detecting Network Attacks"*, Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/abstract/document/6558266/ . Ανακτήθηκε στις 5/12/2017
32. Risto Vaarandi, *"Detecting Anomalous Network Traffic in Organizational Private Networks"*. Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/document/6523859/citations?tabFilter=papers . Ανακτήθηκε στις 5/12/2017
33. Zhengmin Xia, Songnian Lu and Jianhua Li (2012), *"DDoS Flood Attack Detection Based On Fractal Parameters"*, Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/document/6478475/metrics . Ανακτήθηκε στις 6/12/2017
34. P. Jongsuebsuk, N. Wattanapongsakorn, C. Charnsripinyo (2013), *"Network Intrusion Detection with Fuzzy Genetic Algorithm for Unknown Attacks"*, Διαθέσιμο στον ιστότοπο http://ieeexplore.ieee.org/abstract/document/6559603/ . Ανακτήθηκε στις 6/12/2017
35. Google.gr(2017),*Diffie-Hellman*, Διαθέσιμο στον ιστότοπο https://www.google.gr/search?q=%CE%B4%CE%B9%CF%86%CF%86%CE%B9%CE%B5+%CE%B7%CE%B5%CE%BB%CE%BC%CE%B1%CE%BD&client=firefox-b-ab&dcr=0&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiMiM29zqDZAhUohaYKHRAKBEQQ_AUICygC&biw=1600&bih=786#imgrc=_iDJ7_rBchu-FM: . Ανακτήθηκε στις 24/4/2017
36. S.Saraswathi , P.Yogesh(2012), *Mitigating Strategy to Shield the VPN Servicefrom DoS Attack,* Διαθέσιμο στον ιστότοπο http://wireilla.com/papers/ijcis/V2N2/2212ijcis05.pdf . Ανακτήθηκε στις 10/10/2017

Πηγές

Παράρτημα Α Κώδικας Μηχανισμού Ανίχνευσης Επιθέσεων

A) AttackDetection

```
package ddosDetection;

import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.time.LocalDateTime;
import java.util.Hashtable;

import javax.swing.JDialog;
import javax.swing.JOptionPane;

public class AttackDetection {

	private static int TotalPacketsReceived;

	private static int ICMPCounter;
	private static int UDPCounter;
	private static int TCPSYNCounter;
	private Hashtable<Integer,Integer> IPv4Hash;
	private Hashtable<Integer,Integer> IPv6Hash;

	private float meanICMPCounter;
	private float meanUDPCounter;
	private float meanTCPSYNCounter;

	private int spikeICMPCounter;
	private int spikeUDPCounter;
	private int spikeTCPSYNCounter;

	private static boolean UDPAttackShown;
```

```
private static boolean TCPAttackShown;
private static boolean ICMPAttackShown;
private boolean IPv4AttackShown;
private boolean IPv6AttackShown;

private PrintWriter writer;
private int secondsPassed;

private final JDialog dialog;

private int AttackThreshold;

public AttackDetection(int Threshold){
    try {
    writer = new PrintWriter("Attacks_Detected.txt", "UTF-8");
    }catch(FileNotFoundException | UnsupportedEncodingException ue) {
        System.err.println("File could not be created");
    }
    dialog              = new JDialog();
    dialog.setAlwaysOnTop(true);

    TotalPacketsReceived = 0;
    IPv4Hash            = new Hashtable<Integer,Integer>(100);
    IPv6Hash            = new Hashtable<Integer,Integer>(100);
    UDPCounter          = 0;
    ICMPCounter         = 0;
    TCPSYNCounter       = 0;
    UDPAttackShown      = false;
    TCPAttackShown      = false;
    ICMPAttackShown     = false;
    IPv4AttackShown     = false;
    IPv6AttackShown     = false;
    AttackThreshold     = Threshold;
```

```
            secondsPassed = 0;
            meanICMPCounter = 0;
            meanUDPCounter = 0;
            meanTCPSYNCounter = 0;

        }

        public void resetCounters() {
            IPv4Hash.clear();
            IPv6Hash.clear();

            calculateMeans();
            calculateSecondsOfSpike();

            UDPCounter = 0;
            ICMPCounter = 0;
            TCPSYNCounter = 0;
            UDPAttackShown  = false;
            TCPAttackShown  = false;
            ICMPAttackShown = false;
            IPv4AttackShown = false;
            IPv6AttackShown = false;



        }

        private void calculateMeans() {
            secondsPassed++;
            meanICMPCounter      = ((secondsPassed-1)*meanICMPCounter  + ICMPCounter)
/secondsPassed;
            meanUDPCounter              = ((secondsPassed-1)*meanUDPCounter   + UDPCounter)
/secondsPassed;
```

```
        meanTCPSYNCounter = ((secondsPassed-1)*meanTCPSYNCounter+
TCPSYNCounter)/secondsPassed;
    }

    private void calculateSecondsOfSpike() {

        spikeICMPCounter   =
(ICMPCounter  >5*meanICMPCounter)  ?spikeICMPCounter++  :0;
        spikeUDPCounter    =
(UDPCounter   >5*meanUDPCounter)   ?spikeUDPCounter++   :0;
        spikeTCPSYNCounter =
(TCPSYNCounter>5*meanTCPSYNCounter)?spikeTCPSYNCounter++:0;

    }

    public void closeFile() {
        writer.close();
    }

    public boolean[] getAttacks() {
        boolean[] attacks = { TCPAttackShown, UDPAttackShown, ICMPAttackShown};
        return attacks;
    }

    public void increasePacketsReceived() {
        TotalPacketsReceived++;
    }

    public int getPacketsReceived() {
        return TotalPacketsReceived;
    }

    private boolean detectAttack(String packetType, int secondsOfSpike, int counter, boolean
attackShown) {
```

```
                if((counter>=AttackThreshold || secondsOfSpike>=3) && !attackShown) {


                        writer.println("Attack Detected "+ packetType+": "
+counter+LocalDateTime.now().toString());

                        JOptionPane.showMessageDialog(dialog, "Atack Detected "+ packetType+" :"
+ counter + "\nAt time " +LocalDateTime.now().toString(), "Attack
Detected" ,JOptionPane.ERROR_MESSAGE);

                        attackShown = true;

                }

                return attackShown;

        }


        public void increaseICMPCounter() {

                ICMPCounter++;

                ICMPAttackShown =
detectAttack("ICMP",spikeICMPCounter,ICMPCounter,ICMPAttackShown);

        }


        public void increaseTCPSYNCounter() {

                TCPSYNCounter++;

                TCPAttackShown = detectAttack("TCP
SYN",spikeTCPSYNCounter,TCPSYNCounter,TCPAttackShown);



        }


        public void decreaseTCPSYNCounter() {

                TCPSYNCounter--;

        }


        public void increaseUDPCounter() {

                UDPCounter++;

                UDPAttackShown =
detectAttack("UDP",spikeUDPCounter,UDPCounter,UDPAttackShown);

        }


        public void insertIPv4(byte[] IPv4) {
```

```
        IPv4Hash.put(hashKey(IPtoString(IPv4)),
(IPv4Hash.get(hashKey(IPtoString(IPv4)))!=null)?IPv4Hash.get(hashKey(IPtoString(IPv4)))+1:0)
;

        if(IPv4Hash.get(hashKey(IPtoString(IPv4)))!=null){

            if (IPv4Hash.get(hashKey(IPtoString(IPv4)))>=AttackThreshold
&& !IPv4AttackShown){

                this.IPv4AttackShown = true;


                writer.println("Attack Detected from IP IPv4:" + IPtoString(IPv4));

                JOptionPane.showMessageDialog(dialog, "Attack Detected IPv4 :"
+IPv4Hash.get(hashKey(IPtoString(IPv4))), "Attack
Detected" ,JOptionPane.ERROR_MESSAGE);

            }

        }

    }


    public void insertIPv6(byte[] IPv6) {

        IPv6Hash.put(hashKey(IPtoString(IPv6)),
(IPv6Hash.get(hashKey(IPtoString(IPv6)))!=null)?IPv6Hash.get(hashKey(IPtoString(IPv6)))+1:0)
;

        if(IPv4Hash.get(hashKey(IPtoString(IPv6)))!=null){

            if (IPv4Hash.get(hashKey(IPtoString(IPv6)))>=AttackThreshold
&& !IPv6AttackShown){

                this.IPv6AttackShown = true;


                writer.println("Attack Detected from IP IPv6:" + IPtoString(IPv6));

                JOptionPane.showMessageDialog(dialog, "Attack Detected IPv6 :"
+IPv6Hash.get(hashKey(IPtoString(IPv6))), "Attack
Detected" ,JOptionPane.ERROR_MESSAGE);

            }

        }

    }


    public String printIPv4Table() {

        return "IPv4Table : \n" + IPv4Hash.toString();

    }
```

```
    public String printIPv6Table() {
        return "IPv6Table : \n" + IPv6Hash.toString();
    }

    public void printCounters() {
        System.out.println("Detection counters : ");
        System.out.println("Total Packets Received : " + TotalPacketsReceived);
        System.out.println("TCP SYN counters        : " + TCPSYNCounter);
        System.out.println("UDP     counters    : " + UDPCounter);
        System.out.println("ICMP    counters    : " + ICMPCounter);
    }

    private String IPtoString(byte[] IP){
        String IPstr;
        if (IP.length==4){
            IPstr = org.jnetpcap.packet.format.FormatUtils.ip(IP);
        }else{
            IPstr = org.jnetpcap.packet.format.FormatUtils.asStringIp6(IP,true);
        }

        return IPstr;
    }

    private int hashKey(String IPstr) {
        return IPstr.hashCode();
    }

    public int[] getAndResetCounters() {
        int [] counters = {TotalPacketsReceived, TCPSYNCounter, UDPCounter,
ICMPCounter};
        resetCounters();
        return counters;
    }
```

```
}
```

B) PacketFiltering

```
package ddosDetection;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;


import org.jnetpcap.Pcap;
import org.jnetpcap.Pcap.Direction;
import org.jnetpcap.PcapIf;
import org.jnetpcap.PcapSockAddr;
import org.jnetpcap.packet.PcapPacket;
import org.jnetpcap.packet.PcapPacketHandler;
import org.jnetpcap.protocol.lan.Ethernet;
import org.jnetpcap.protocol.network.Icmp;
import org.jnetpcap.protocol.network.Icmp.IcmpType;
import org.jnetpcap.protocol.network.Ip4;
import org.jnetpcap.protocol.network.Ip6;
import org.jnetpcap.protocol.tcpip.Tcp;
import org.jnetpcap.protocol.tcpip.Udp;

public class PacketFiltering {

    private AttackDetection ad;
    private PcapIf device;

    private static List<PcapIf> alldevs;

    PcapPacketHandler<String> jpacketHandler = new PcapPacketHandler<String>() {
        public void nextPacket(PcapPacket packet, String user) {
```

```
            Ethernet ethernet = new Ethernet();
            Ip4 ip = new Ip4();
            Ip6 ip6 = new Ip6();
            Udp udp = new Udp();
            Tcp tcp = new Tcp();
            Icmp icmp = new Icmp();

            ad.increasePacketsReceived();

            if (packet.hasHeader(icmp)) {
                handleICMPPacket(packet, icmp);
            }

            if(packet.hasHeader(udp)) {
                handleUDPPacket(packet,udp);
            }else if(packet.hasHeader(tcp)) {
                handleTCPPacket(packet,tcp);
            }

            if (packet.hasHeader(ip) ) {
                handleIPv4Packet(packet, ethernet, ip);
            }
            if (packet.hasHeader(ip6)) {
                handleIPv6Packet(packet, ethernet, ip6);
            }

            return;
        }
    };

    public PacketFiltering(int packetThreshold) {
```

```
        ad = new AttackDetection(packetThreshold);
    }


    public AttackDetection getAttackDetection() {
        return this.ad;
    }

    public PcapIf getDevice() {
        return this.device;
    }




    public String[][] getInterfaces() {

        alldevs = new ArrayList<PcapIf>(); // Will be filled with NICs
        StringBuilder errbuf = new StringBuilder(); // For any error msgs
        int r = Pcap.findAllDevs(alldevs, errbuf);
        if (r != Pcap.OK || alldevs.isEmpty()) {
            System.err.printf("Can't read list of devices, error is %s",
                    errbuf.toString());
            return null;
        }

        String[][] devicesAndAddresses = new String[alldevs.size()][4];
        int i = 0;
        for (PcapIf device : alldevs) {

            devicesAndAddresses[i][0] = device.getDescription();
            getAddresses(device,devicesAndAddresses[i++]);
```

```
        }


        return devicesAndAddresses;
    }

    public void selectItfcToCapture(int selectedInterface) {




    private void getAddresses(PcapIf device, String[] addresses) {
        for(int i=0;i<device.getAddresses().size() & i<3 ;i++) {
            if(device.getAddresses().get(i).getAddr().getFamily()==PcapSockAddr.AF_INET) {
                addresses[i+1] =
org.jnetpcap.packet.format.FormatUtils.ip(device.getAddresses().get(i).getAddr().getData());
            }

    if(device.getAddresses().get(i).getAddr().getFamily()==PcapSockAddr.AF_INET6){
                addresses[i+1] =
org.jnetpcap.packet.format.FormatUtils.asStringIp6(device.getAddresses().get(i).getAddr().getDat
a(),true);
            }
        }


    }


    private boolean addressMatch(PcapIf device,byte[] AddressToCompare,boolean isIPv4) {
        for(int i=0;i<device.getAddresses().size();i++) {
            if((device.getAddresses().get(i).getAddr().getFamily()==PcapSockAddr.AF_INET)
&& isIPv4){

    if(Arrays.equals(AddressToCompare,device.getAddresses().get(i).getAddr().getData())) {
                    return true;
                }
            }
```

```
    if((device.getAddresses().get(i).getAddr().getFamily()==PcapSockAddr.AF_INET6)
&& !isIPv4){

    if(Arrays.equals(AddressToCompare,device.getAddresses().get(i).getAddr().getData())) {
                        return true;
                    }
                }
            }
        return false;
    }

    public Pcap openDeviceForCapture() {
        StringBuilder errbuf = new StringBuilder(); // For any error msgs
        int snaplen = 64 * 1024; // Capture all packets, no trucation
        int flags = Pcap.MODE_PROMISCUOUS; // capture all packets
        int timeout = 10 * 1000; // 10 seconds in millis
        Pcap pcap = Pcap.openLive(device.getName(), snaplen, flags, timeout, errbuf);

        if (pcap == null) {
            System.err.printf("Error while opening device for capture: "+ errbuf.toString());
        }

        pcap.setDirection(Direction.IN);

        return pcap;
    }

    public void closeItfc(Pcap pcap) {
        pcap.close();
    }

    private void handleICMPPacket(PcapPacket packet,Icmp icmp) {
        if(!packet.getHeader(icmp).hasSubHeader(IcmpType.ECHO_REPLY.getId())){
```

```
            ad.increaseICMPCounter();
        }
    }

    private void handleUDPPacket(PcapPacket packet, Udp udp) {
        ad.increaseUDPCounter();
    }

    private void handleTCPPacket(PcapPacket packet, Tcp tcp) {

        if(packet.getHeader(tcp).flags_SYN() && !packet.getHeader(tcp).flags_ACK()) {
            ad.increaseTCPSYNCounter();
        }else if (!packet.getHeader(tcp).flags_SYN() && packet.getHeader(tcp).flags_ACK()) {
            ad.decreaseTCPSYNCounter();
        }

    }

    private void handleIPv4Packet(PcapPacket packet, Ethernet ethernet, Ip4 ip) {
        byte[] sIP;

        sIP = packet.getHeader(ip).source();

        if(!addressMatch(device,sIP,true)) {
            ad.insertIPv4(sIP);
        }

        return;
    }

    private void handleIPv6Packet(PcapPacket packet, Ethernet ethernet, Ip6 ip6) {
        byte[] sIP6;
```

```
            sIP6 = packet.getHeader(ip6).source();


            if(!addressMatch(device,sIP6,false)) {
                ad.insertIPv6(sIP6);
            }

            return;
        }

        public void getPacket(Pcap pcap) {
            pcap.loop(1, jpacketHandler, " ");
        }

}
```

c) DDoSDetection

```
package ddosDetection;

import java.util.Timer;
import java.util.TimerTask;

import org.jnetpcap.Pcap;

public class DDosDetection {

    private Timer timer;
    private TimerTask timertask;
    private PacketFiltering pf;
    private AttackDetectionGUI gui;


    public void initAll(int threshold) {
        this.timer = new Timer();
```

```
        this.timertask = new TimerTask() {
            @Override
             public void run() {
                int[] counters = pf.getAttackDetection().getAndResetCounters();
                gui.updateAttacks(pf.getAttackDetection().getAttacks());
                gui.updateCounters(counters);



            };
        };
        this.pf = new PacketFiltering(threshold);
    }


    public DDosDetection() {

    }

    public static void main(String[] args) throws InterruptedException{

        DDosDetection dd = new DDosDetection();

        dd.gui = new AttackDetectionGUI();
        Thread t1 = new Thread(dd.gui);
        t1.start();

        while(!dd.gui.getHasThreshold().get()) {
            Thread.sleep(100);
        }

        dd.initAll(dd.gui.getThreshold());
```

```
        String[][] devicesAndAddresses = dd.pf.getInterfaces();

        dd.gui.populateTable(devicesAndAddresses);

        while(!dd.gui.getHasInterfaceSelected().get()) {
            Thread.sleep(100);
        }



        dd.pf.selectItfcToCapture(dd.gui.getInterfaceSelected());



        Pcap pcap = dd.pf.openDeviceForCapture();
        dd.startTimer();
        if (pcap!=null) {

            while(dd.pf.getAttackDetection().getPacketsReceived()<100000
&& !dd.gui.systemExited()) {
                dd.pf.getPacket(pcap);
            }
        }

        dd.pf.getAttackDetection().closeFile();
        dd.stopTimer();
        dd.pf.closeItfc(pcap);



         System.exit(0);
    }

    private void startTimer() {
```

```
        this.timer.schedule(this.timertask, (long)0, (long)1000); // Schedules a timer that counts
each second


    }


    private void stopTimer() {


        this.timer.cancel();// Cancels the timer


    }


}
```