ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αναπαραγωγή κυβερνο-επιθέσεων σε περιβάλλον προσομοίωσης

Διπλωματική Εργασία

Γύφτος Γεώργιος

Επιβλέπων Καθηγητής: Δημήτρης Ασκούνης

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αναπαραγωγή κυβερνο-επιθέσεων σε περιβάλλον προσομοίωσης

Διπλωματική Εργασία

Γύφτος Γεώργιος

Επιβλέπων Καθηγητής: Δημήτρης Ασκούνης

Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Οκτωβρίου 2020.

…….……..	……	……..
Δημήτριος Ασκούνης,	Ιωάννης Ψαρράς,	Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π	Αν. Καθηγητής Ε.Μ.Π

……

Γύφτος Γεώργιος

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

Περίληψη

Η εμφάνιση κυβερνο-επιθέσεων, οι οποίες στοχεύουν στην παραβίαση συστημάτων, αυξάνεται συνεχώς. Οι επιθέσεις αυτές έχουν μεγάλο αντίκτυπο στην κοινωνία, αφού μπορεί να έχουν σαν αποτέλεσμα την απώλεια δεδομένων που αφορούν είτε την προσωπική ζωή ενός ατόμου είτε την λειτουργία διαφόρων οργανισμών. Ειδικότερα, οι επιθέσεις μπορεί να έχουν ως στόχο κρίσιμες εγκαταστάσεις, όπως δίκτυα ηλεκτρισμού, ύδρευσης ή/και άλλων σημαντικών πόρων/υπηρεσιών, επιφέροντας σοβαρές κοινωνικο-οικονομικές συνέπειες ή ακόμα και την απώλεια ανθρώπινων ζωών. Επιπλέον, οι επιθέσεις αυτές εξελίσσονται διαρκώς με αποτέλεσμα να δυσκολεύει ολοένα και περισσότερο η αντιμετώπισή τους.

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η σχεδίαση μοντέλων επιθέσεων στο πλαίσιο περιβάλλοντος προσομοίωσης, η ανάλυση αυτών των μοντέλων, η διερεύνηση της απόκρισης του συστήματος και τελικά η αξιολόγηση τους. Ο αριθμός των επιθέσεων που επιλέχθηκαν είναι τέσσερις και είναι βασισμένες σε μερικές από τις πιο γνωστές κυβερνο-επιθέσεις που πραγματοποιούνται για την παραβίαση συστημάτων. Προκειμένου να κατασκευασθεί ένα περιβάλλον προσομοίωσης των επιθέσεων χρησιμοποιήθηκε η εφαρμογή Oracle VM VirtualBox. Επίσης, για να εκτελεσθούν οι επιθέσεις επιλέχθηκε η χρήση της προγραμματιστικής γλώσσας python, αλλά και της γραμμής εντολών(terminal).

Λέξεις Κλειδιά: VirtualBox, python, κυβερνο-επιθέσεις, προσομοίωση

Summary

The incidence of cyber-attacks, which aim to breach systems, is constantly increasing. These attacks have a major impact on society, as they can result in the loss of data concerning either an individual's personal life or the operation of various organizations. In particular, attacks can target critical installations, such as electricity, water and / or other important resources / services, with serious socio-economic consequences or even the loss of human lives. In addition, these attacks are constantly evolving, making them increasingly difficult to deal with.

The purpose of this dissertation was to design attack models in the context of a simulation environment, to analyze these models, to investigate the response of the system and finally to evaluate them. The number of attacks selected is four and is based on some of the most well-known cyber-attacks for system breaches. Oracle VM VirtualBox was used to build an attack simulation environment. Also, to execute the attacks, the use of the python programming language was chosen, as well as the command line (terminal).

Key Words: VirtualBox, python, cyber-attacks, simulation

Ευχαριστίες

Το παρών έργο διεξήχθη στο Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου, υπό την επίβλεψη και την καθοδήγηση των κ. Μιχαήλ Κοντούλη και κ. Γιώργου Δούκα. Η διπλωματική αυτή ερευνά τις κυβερνο-επιθέσεις και την ανάλυση αυτών.

Θα ήθελα να ευχαριστήσω τους επιβλέποντες κ. Μιχαήλ Κοντούλη και κ. Γιώργο Δούκα για την παρότρυνση πάνω στο συγκεκριμένο αντικείμενο, καθώς και την εμπιστοσύνη που έδειξε προς το πρόσωπό μου αναθέτοντάς μου την παρούσα διπλωματική εργασία.

Τέλος, αφιερώνω το παρόν έργο στην οικογένειά μου, για την αμέριστη στήριξη και εμπιστοσύνη που μου έδειξαν στις επιλογές και στις σπουδές μου. Ευχαριστώ τον αδερφό και την αδερφή μου που συνεισέφεραν με το δικό τους τρόπο ο καθένας στην προσπάθειά μου αυτή, καθώς επίσης και τους γονείς μου, οι οποίοι μου έμαθαν να ακολουθώ πάντοτε τα όνειρά μου και να επιστρατεύομαι τη γνώση ως σύμμαχο σε όλα μου τα προβλήματα. Θα ήθελα να ευχαριστήσω, ακόμη, τους φίλους μου για τη συμπαράσταση και την υπομονή τους καθ’ όλη τη διάρκεια των σπουδών μου στο Ε.Μ.Π. Σας είμαι ευγνώμων για τον άνθρωπο στον οποίο έχω εξελιχθεί σήμερα.

Περιεχόμενα

Λίστα Εικόνων

ΚΕΦΑΛΑΙΟ 1. Εισαγωγή

Τα τελευταία χρόνια το διαδίκτυο έχει γίνει αναπόσπαστο κομμάτι της ζωής μας, καθώς το χρησιμοποιούμε όχι μόνο για προσωπικούς λόγους αλλά και για επαγγελματικούς. Μέσω του διαδικτύου μπορούμε να μιλήσουμε ή να αλληλεπιδράσουμε με άλλα άτομα και γενικά να περάσουμε τον ελεύθερο χρόνο μας, αλλά παράλληλα να εργαστούμε να πραγματοποιήσουμε τις συναλλαγές μας, και να ενημερωθούμε για οτιδήποτε επιθυμούμε. Επιπρόσθετα, η διαρκώς αυξανόμενη χρήση του διαδικτύου δημιουργεί νέες ανάγκες, για την κάλυψη των οποίων, νέες τεχνολογίες αναπτύσσονται. Ως παράδειγμα αναφέρεται η τεχνολογία Internet of Things(IoT), η οποία μπορεί να χρησιμοποιηθεί τόσο για την διευκόλυνση της καθημερινότητας μας όσο και για την φροντίδα ασθενών που χρειάζονται συνεχής φροντίδα. Για να λειτουργήσει όμως το διαδίκτυο είναι αναγκαία η αποθήκευση δεδομένων και πληροφοριών, πράγμα που δημιουργεί την επιθυμία κακόβουλων ατόμων να καταστρέψουν ή ακόμα χειρότερα, να αποκτήσουν πρόσβαση στα δεδομένα αυτά. Τα δεδομένα μπορεί να αφορούν από τον προσωπικό λογαριασμό ενός ατόμου για μία τραπεζική εφαρμογή, μέχρι ευαίσθητες πληροφορίες που αφορούν την ασφάλεια μίας χώρας. Επομένως, προκύπτει η ανάγκη ανάπτυξης αμυντικών μηχανισμών για την αντιμετώπιση των κακόβουλων αυτών ενεργειών και κατ' επέκταση την ασφάλεια των δεδομένων.

1.1. Αντικείμενο-Σκοπός

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάλυση κυβερνο-επιθέσεων και η διερεύνηση της απόκρισης του συστήματος, ώστε να διερευνηθεί κατά πόσο μπορούμε να αντιληφθούμε έγκαιρα μία επίθεση προκειμένου να την αντιμετωπίσουμε προτού δημιουργήσει πρόβλημα στο σύστημά μας.

Για την αναπαραγωγή των κυβερνο-επιθέσεων χρησιμοποιήθηκε περιβάλλον προσομοίωσης με την βοήθεια της εφαρμογής VirtualBox. Η εφαρμογή αυτή μπορεί να εγκατασταθεί σε έναν ηλεκτρονικό υπολογιστή και δίνει την δυνατότητα να δημιουργηθεί ένας εικονικός ηλεκτρονικός υπολογιστής ο οποίος λειτουργεί "μέσα" στον φυσικό ηλεκτρονικό υπολογιστή. Φυσικά ο εικονικός ηλεκτρονικός υπολογιστής έχει όρια όσο αφορά τις δυνατότητές του. Πιο συγκεκριμένα, τα χαρακτηριστικά του- για παράδειγμα η μνήμη του- καθορίζονται από το ποσοστό των αντίστοιχων χαρακτηριστικών του φυσικού ηλεκτρονικού υπολογιστή που επιθυμεί ο χρήστης να διαθέσει στον εικονικό ηλεκτρονικό υπολογιστή. Επομένως τα χαρακτηριστικά του εικονικού Η/Υ έχουν, στην καλύτερη περίπτωση, όρια ίσα με αυτά του φυσικού Η/Υ. Μπορούμε να έχουμε, επίσης, παραπάνω από έναν Η/Υ ενεργό ταυτόχρονα. Ωστόσο οι δυνατότητες των εικονικών Η/Υ θα είναι μικρότερες από όταν θα ήταν ενεργός

μόνο ένας εικονικός Η/Υ. Αυτό συμβαίνει επειδή το άθροισμα των χαρακτηριστικών των ενεργών εικονικών Η/Υ δεν είναι δυνατό να ξεπερνάει τα όρια των χαρακτηριστικών του φυσικού Η/Υ. Στην πραγματικότητα το άθροισμα των χαρακτηριστικών των ενεργών εικονικών Η/Υ δεν γίνεται ποτέ να φτάσει τα όρια αυτών του φυσικού Η/Υ γιατί ο φυσικός Η/Υ χρειάζεται να δεσμεύει πάντα ένα μέρος των χαρακτηριστικών του για να λειτουργήσει.

1.2. Οργάνωση Τόμου

Η διπλωματική εργασία είναι οργανωμένη σε 4 κεφάλαια. Η δομή τους είναι η εξής:

- Στο 2ο Κεφάλαιο παρουσιάζονται θεωρητικές έννοιες, οι οποίες συνδέονται με την εργασία και θα βοηθήσουν στην διαμόρφωση μιας γενικής εικόνας για το υπό εξέταση πρόβλημα.
- Στο 3ο Κεφάλαιο παρουσιάζονται το περιβάλλον προσομοίωσης στο οποίο εκτελέσθηκε το πείραμα, καθώς και θεωρητικές έννοιες σχετικά με το περιβάλλον προσομοίωσης.
- Στο 4ο Κεφάλαιο παρουσιάζονται οι λεπτομέρειες της υλοποίησης, ήτοι ποιες επιθέσεις αναπαράχθηκαν στο περιβάλλον προσομοίωσης, τον τρόπο με τον οποίο αναπαράχθηκαν αλλά και την ανάλυση αυτών.
- Στο 5ο Κεφάλαιο παρουσιάζεται μια αποτίμηση της εργασίας ως σύνολο, οι πιθανές επεκτάσεις της καθώς και ενδεικτικά τα επόμενα της έρευνας.

ΚΕΦΑΛΑΙΟ 2.Θεωρητικό Υπόβαθρο – Βασικές έννοιες

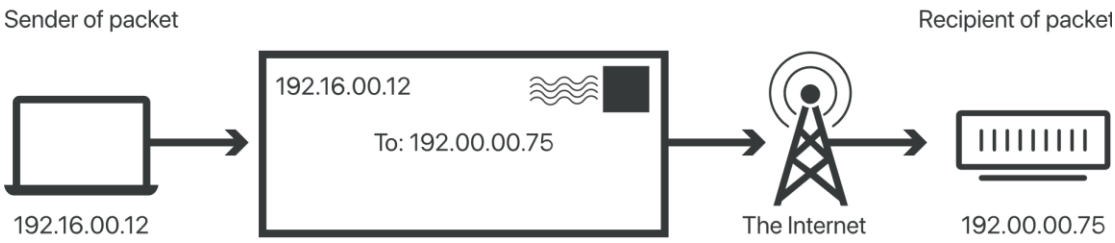
Σκοπός του παρόντος κεφαλαίου είναι να εισάγει τον αναγνώστη στις βασικότερες έννοιες που σχετίζονται με αυτή την διπλωματική εργασία. Στο πρώτο μέρος του κεφαλαίου παρουσιάζονται βασικά χαρακτηριστικά της λειτουργίας του Διαδικτύου(Internet). Στο δεύτερο μέρος του κεφαλαίου παρουσιάζονται τα αντιπροσωπευτικά είδη των κυβερνο-επιθέσεων που καταγράφονται.

2.1. Δομή και τρόπος λειτουργίας του Διαδικτύου

2.1.1. Internet Protocol (IP)

Το Internet Protocol (IP) είναι ένα πρωτόκολλο, δηλαδή ένα σύνολο από κανόνες, για την δρομολόγηση των πληροφοριών και δεδομένων που θέλουμε να ανταλλάξουμε. Τα δεδομένα αυτά προκειμένου να δρομολογηθούν είναι αναγκαίο πρώτα να διασπαστούν πρώτα σε μικρότερα κομμάτια τα οποία ονομάζονται πακέτα. IP πληροφορία εισέρχεται σε κάθε πακέτο και μέσω αυτής της πληροφορίας οι δρομολογητές (routers) μπορούν να στείλουν το πακέτο στο σωστό μέρος. Για να ξέρουν οι routers που πρέπει να στείλουν το κάθε πακέτο υπάρχουν εικονικές διευθύνσεις που ονομάζονται IP διευθύνσεις (Cloudflare – What is the Internet Protocol? , n.d.). Κάθε συσκευή, ιστοσελίδα ή εφαρμογή είναι συνδεδεμένη με μια IP διεύθυνση.

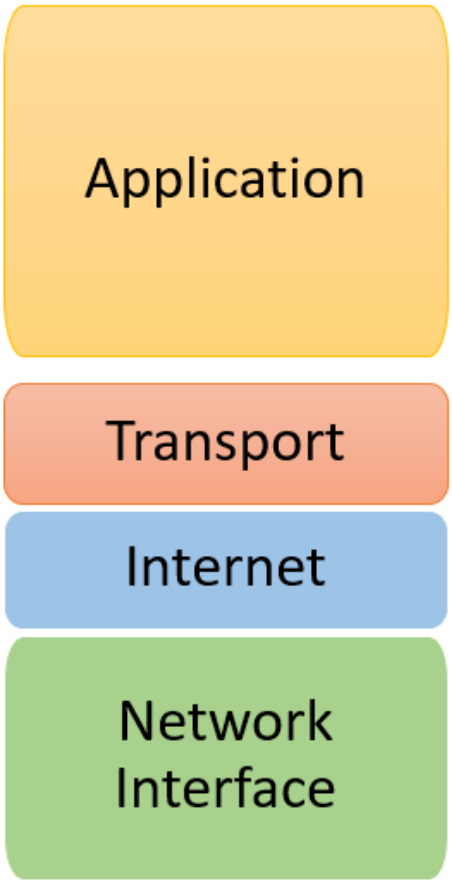
Οι IP διευθύνσεις είναι μία σειρά από χαρακτήρες και χωρίζονται σε δύο κατηγορίες, τις IPv4 και τις IPv6. Οι IPv4 διευθύνσεις είναι της μορφής “192.168.1.1”, ενώ οι IPv6 διευθύνσεις είναι της μορφής “3ffe:1893:3452:4:345:f345:f345:42fc”(Medium – How Does The Internet Work?, Steven Li, 01/08/2017) . Ωστόσο είναι δύσκολο για κάθε χρήστη του διαδικτύου, που θέλει για παράδειγμα να επισκεφθεί μία ιστοσελίδα, να ξέρει μία τέτοια σειρά από χαρακτήρες. Για τον λόγο αυτό οι IP διευθύνσεις αντιστοιχίζονται σε ονόματα τα οποία ονομάζονται domain names . Κάθε φορά που επικοινωνούν δύο ή περισσότερες συσκευές στέλνεται ένα IP πακέτο το οποίο περιέχει τόσο την IP διεύθυνση του αποστολέα όσο και του παραλήπτη. Συγκεκριμένα ένα IP πακέτο είναι ένα πακέτο από πληροφορίες, στο οποίο προστίθεται ένα IP header προτού σταλθεί. IP header είναι ένα σύνολο από πληροφορίες που αφορούν το κάθε πακέτο, όπως για παράδειγμα το μήκος του πακέτου. (Cloudflare – What is the Internet Protocol? , n.d.) Είναι σημαντικό να σημειωθεί ότι προκειμένου να μεταφερθεί ένα πακέτο πληροφορίας μέσω του διαδικτύου είναι πιθανό το πακέτο να διασπαστεί σε μικρότερα πακέτα. Παρακάτω αποτυπώνεται σχηματικά ο τρόπος αποστολής ενός πακέτου:



Εικόνα 1. Αποστολή πακέτου

2.1.2. TCP/IP Μοντέλο

Το IP πρωτόκολλο συχνά χρησιμοποιείται σε συνδυασμό με ένα άλλο πρωτόκολλα που ονομάζεται Transmission Control Protocol (TCP) πρωτόκολλο, δημιουργώντας το TCP/IP πρωτόκολλο ή πιο αναλυτικά Transmission Control Protocol/ Internet Protocol. Το TCP είναι πρωτόκολλο μεταφοράς με την βοήθεια του οποίου τα πακέτα στέλνονται στον σωστό προορισμό. Τα TCP και IP πρωτόκολλα σχεδιάστηκαν για να για να χρησιμοποιούνται μαζί, ωστόσο το IP πρωτόκολλο μπορεί να χρησιμοποιηθεί και με άλλα πρωτόκολλα όπως αυτό του UDP (Cloudflare – What is the Internet Protocol?, n.d.). Το πρωτόκολλο που έχει επικρατήσει , είναι το TCP/IP καθώς εγγυάται ότι η πληροφορία θα φτάσει άθικτη στον προορισμό παρόλο που διασπάται σε διάφορα πακέτα κατά την διάδοσή της. Το TCP/IP μοντέλο χρησιμοποιεί τέσσερα επίπεδα και κάθε επίπεδο (layer) έχει το δικό του πρωτόκολλο, ή όπως έχουμε ήδη αναφέρει το δικό του σύνολο κανόνων. Τα εν λόγω επίπεδα (layers) ανταλλάσσουν μεταξύ τους πληροφορία (Guru99 - TCP/IP Model: What is TCP IP Stack? Protocol Layers, Advantages , n.d.). Στο σχήμα που ακολουθεί αναφέρονται τα επίπεδα του TCP/IP μοντέλου:



Εικόνα 2. TCP/IP layers

2.1.3. TCP/IP Layers

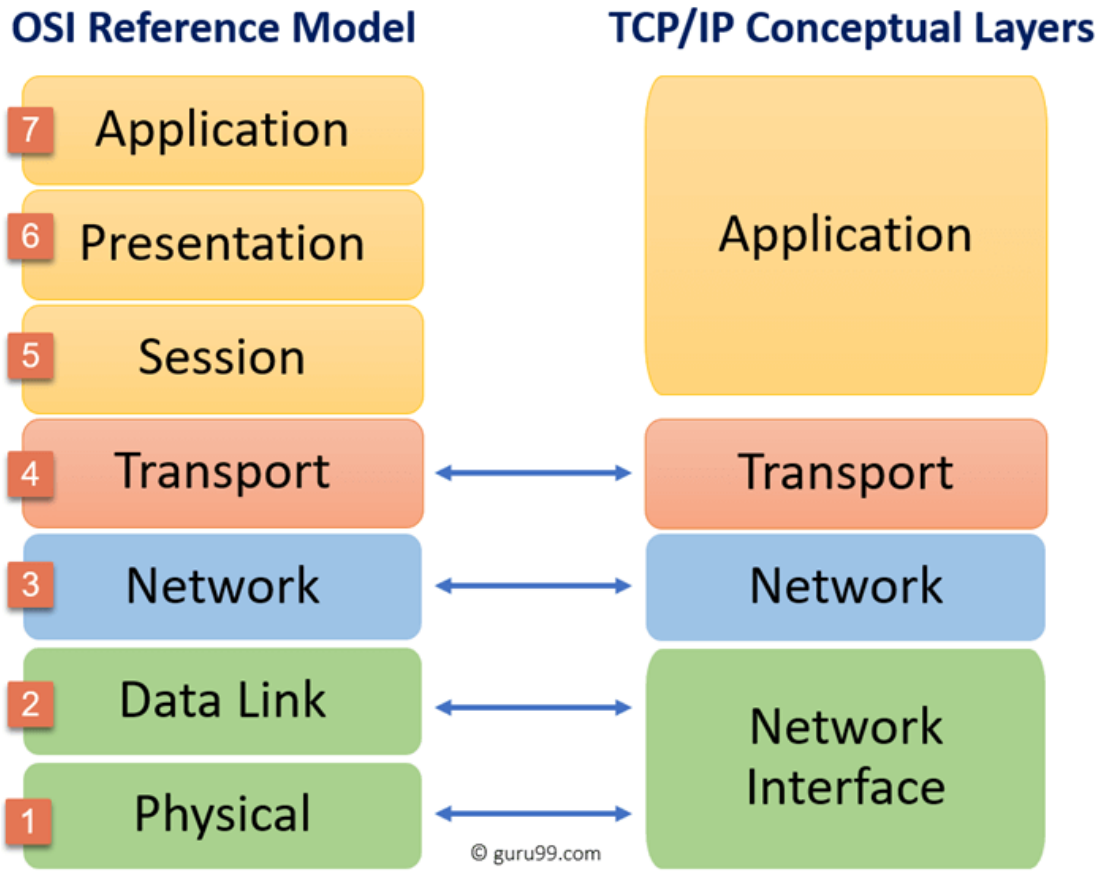
Αρχικά το Application Layer είναι το υψηλότερο επίπεδο του TCP/IP μοντέλου και είναι αυτό που είναι υπεύθυνο για την αλληλεπίδραση με μία εφαρμογή λογισμικού και για την πρόσβαση στο διαδίκτυο. Κάποια από τα πιο γνωστά πρωτόκολλα που χρησιμοποιεί το Application layer είναι τα FTP, HTTP, HTTPS.

Το επόμενο επίπεδο είναι το Transport Layer. Το επίπεδο αυτό προετοιμάζει την πληροφορία προκειμένου να φτάσει άθικτη στον προορισμό της. Για παράδειγμα καθορίζει πόσο πληροφορία θα σταλθεί και πού, καθώς και τον ρυθμό στον οποίο θα σταλθεί η πληροφορία αυτή.

Το επίπεδο που ακολουθεί είναι το Internet Layer. Το Internet Layer, γνωστό και ως Network layer, είναι υπεύθυνο για την αποστολή πακέτων από ένα δίκτυο σε ένα άλλο. Αυτό επιτυγχάνεται με την βοήθεια μεθόδων ή πρωτοκόλλων που προσφέρει το Internet Layer, όπως το IP πρωτόκολλο.

Το τελευταίο επίπεδο είναι το Network Interface Layer. Το Network Interface Layer είναι το επίπεδο που προσδιορίζει πώς θα μεταδοθεί στην πράξη η πληροφορία μέσα στο δίκτυο, δηλαδή πώς θα μεταδοθεί από μία συσκευή σε μία άλλη.

Πέρα από το TCP/IP μοντέλο υπάρχει και το OSI μοντέλο. Το OSI μοντέλο έχει περισσότερα επίπεδα από το TCP/IP (Guru99 - TCP/IP Model: What is TCP IP Stack? Protocol Layers, Advantages , n.d.) . Στο σχήμα που ακολουθεί παρουσιάζονται οι διαφορές τους:



Εικόνα 3. Διαφορές TCP/IP και OSI μοντέλων

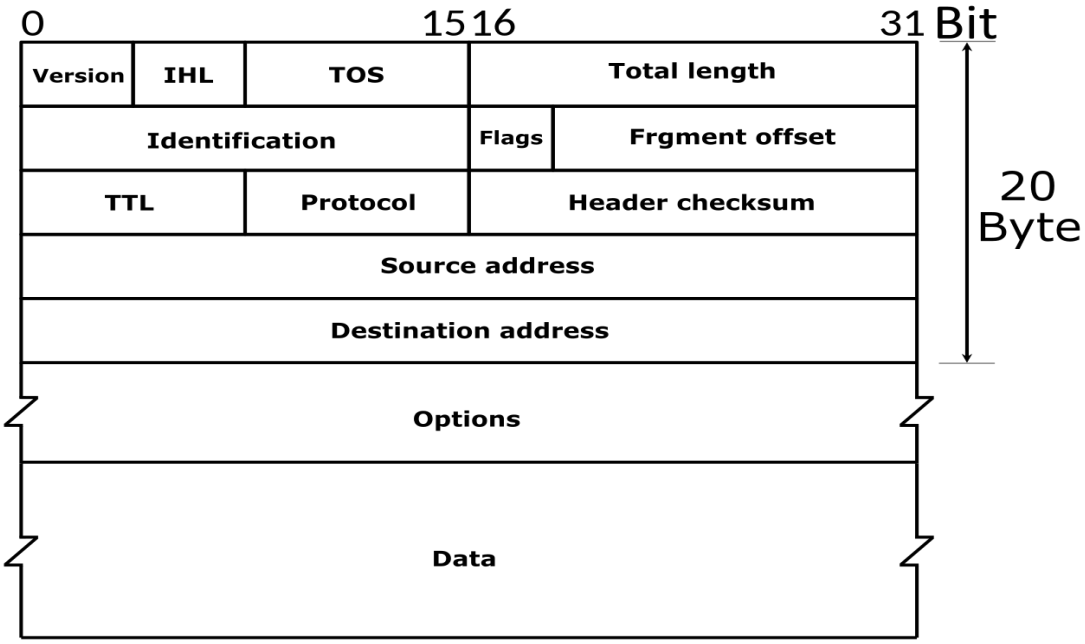
2.1.4. Δομή Πακέτου

Κάθε IP πακέτο έχει ένα μέγεθος. Το μέγεθος αυτό υπολογίζεται με την βοήθεια μίας μετρικής που ονομάζεται bits. Επίσης ένα IP πακέτο αποτελείται από πεδία, τα οποία είναι τα εξής:

- Version field (4 bits). Προσδιορίζει ποια έκδοση του Internet Protocol χρησιμοποιείται από το πακέτο.
- IP Header Length (IHL) field (4 bits). Ενημερώνει πόσες λέξεις με μήκος 32-bit βρίσκονται στο IP header.

- Type-of-service field (8 bits). Προσδιορίζει τον τρόπο με τον οποίο τα πρωτόκολλα υψηλότερου επιπέδου πρόκειται να επεξεργαστούν την πληροφορία που στέλνεται στο πακέτο (datagram).
- Total Length field (16 bits). Η τιμή του πεδίου αυτού είναι ίση με το μέγεθος του πακέτου , συμπεριλαμβανομένου των header και data κομμάτια, σε bytes, όπου 1 byte ισούται με 8 bits.
- Identification field (16 bits). Περιέχει έναν αριθμό ο οποίος προσδιορίζει το τρέχων πακέτο πληροφορίας και χρησιμοποιείται προκειμένου να ενωθούν, όταν φτάσει στον προορισμό του, τα πακέτα που αποτελούν μέρος ενός αρχικού πακέτου το οποίο διασπάστηκε προκειμένου να “ταξιδέψει” στο διαδίκτυο.
- Flags field (3 bits). Ενημερώνει τους δρομολογητές (routers) αν έχουν την δυνατότητα να διασπάσουν το πακέτο σε μικρότερα κομμάτια και αν ναι ενημερώνει για την θέση του πακέτου στην ακολουθία από τα πακέτα που έχουν διασπαστεί και είναι μέρος ενός μεγαλύτερου πακέτου. Πιο συγκεκριμένα το 1ο bit είναι πάντα 0. Το 2ο bit είναι 0 εάν το πακέτο μπορεί να διασπαστεί και 1 αν δεν μπορεί. Τέλος το 3ο bit είναι 0 είτε όταν το πακέτο είναι το τελευταίο κομμάτι ενός μεγαλύτερο πακέτου είτε όταν δεν αποτελεί μέρος κάποιου πακέτου το οποίο έχει διασπαστεί.
- Fragment Offset field (13 bits). Το Fragment Offset πεδίο χρησιμοποιείται για να υποδείξει στον προορισμό του πακέτου , το οποίο αποτελεί μέρος ενός μεγαλύτερου πακέτου που έχει διασπαστεί ή αλλιώς θραύσμα (fragment) πακέτου, πού πρέπει να τοποθετηθεί όταν όλα τα θραύσματα έχουν σταλθεί στον προορισμό και αρχικό πακέτο είναι έτοιμο να κατασκευασθεί. Στην περίπτωση που το πακέτο δεν αποτελεί θραύσμα κάποιου άλλου πακέτου τότε το πεδίο αυτό έχει την τιμή 0. Το ίδιο ισχύει και στην περίπτωση που το πακέτο είναι το πρώτο θραύσμα ενός αρχικού πακέτου.
- Time to Live (TTL) field (8 bits). Η τιμή του πεδίου αυτού δείχνει τον αριθμό των συσκευών που μπορεί το πακέτο να προσπελάσει στο διαδίκτυο. Πιο συγκεκριμένα όταν το πακέτο περνάει από μία συσκευή τότε η τιμή TTL μειώνεται κατά 1. Όταν η τιμή του TTL φτάσει στο 0 τότε το πακέτο απορρίπτεται. Για παράδειγμα εάν ένα πακέτο έχει τιμή TTL ίση με 4 τότε το πακέτο μπορεί να προσπελάσει 4 συσκευές.
- Protocol field (8 bits). Προσδιορίζει το πρωτόκολλο του επόμενου επιπέδου(layer) που θα χρησιμοποιηθεί. Το πιο συνηθισμένο είναι το TCP πρωτόκολλο. Άλλα πρωτόκολλα είναι τα UDP, ICMP.
- Checksum field (16 bits). Το Checksum πεδίο αφορά το header του πακέτου και μέσω αυτού μπορεί να γίνει έλεγχος εάν το header είναι το σωστό ή εάν κατά την μεταφορά του πακέτου έχει υποστεί φθορά.
- Source Address field (32 bits). Το πεδίο αυτό έχει την τιμή της IP διεύθυνσης του αποστολέα του πακέτου.
- Destination Address field (32 bits). Το πεδίο αυτό έχει την τιμή της IP διεύθυνσης του παραλήπτη του πακέτου.

- Options field. Το πεδίο Options δεν έχει σταθερό μέγεθος αλλά εξαρτάται από τις επιλογές (options) που στέλνονται σε κάθε πακέτο. Επίσης είναι σημαντικό να αναφερθεί ότι το πεδίο αυτό μπορεί να μην χρησιμοποιείται.
- Data field. Το μέγεθος του Data πεδίου μπορεί να διαφέρει ανά πακέτο ανάλογα με την πληροφορία που περιέχει. Η πληροφορία που περιέχει αφορά τα υψηλότερα επίπεδα (upper layers) (TechRepublic - Exploring the anatomy of a data packet, 02/07/2001) (PEARSON IT CERTIFICATION - Anatomy of an IPv4 Packet, 29/02/2012).



Εικόνα 4. Δομή IPv4 πακέτου

2.1.5. Client vs Server

Server είναι ένας φυσικός υπολογιστής που εκτελεί υπηρεσίες που είναι αναγκαίες για άλλους υπολογιστές. Ανάλογα με το είδος της υπηρεσίας που εκτελεί ένας server μπορεί να ανήκει σε κατηγορία όπως file server, database server.

Client είναι ένας φυσικός υπολογιστής ή ένα λογισμικό που χρησιμοποιεί τις υπηρεσίες προσφέρει ένας server (LearnTomato -What is a Client? What is a Server? And What is a Host?, 09/05/2014).

2.2. Είδη κυβερνο-επιθέσεων

Οι κυβερνο-επιθέσεις μπορούν να χωριστούν σε κατηγορίες ανάλογα με τον τομέα εφαρμογής της κάθε επίθεσης. Παρακάτω αναλύονται οι κυριότερες κατηγορίες:

2.2.1. Injection code επιθέσεις:

Αφορά επιθέσεις που γίνονται όταν εισάγεται εκτελέσιμος κώδικας στο θύμα. Είδη injection attacks:

- SQL injection: Συμβαίνει όταν μέσω ενός input πεδίου σε ένα site ή web εφαρμογή εισάγεται ένα SQL query το οποίο περιέχει εντολές για την βάση και έτσι μπορούμε για παράδειγμα να πάρουμε δεδομένα της βάσης (acunetix - What is SQL Injection (SQLi) and How to Prevent It, n.d.) .
- Code injection: Συμβαίνει όταν εισάγεται malicious κώδικας σε μορφή κάποιας γλώσσας προγραμματισμού (acunetix – What is Code Injection,15/04/2019).
- CRLF (Carriage Return and Line Feed) injection: Αρχικά Carriage Return είναι ο χαρακτήρας (“\r”) που χρησιμοποιείται όταν γράφουμε στο πληκτρολόγιο για να κάνουμε reset τον κέρσορα στην αρχή της γραμμής του κειμένου, ενώ Line Feed είναι ο χαρακτήρας (“\n”) που χρησιμοποιείται για να πάμε σε καινούρια γραμμή κειμένου. Γενικά οι χαρακτήρες CR και LF υποδηλώνουν το EOF (End of Line). Επομένως αυτή η επίθεση συμβαίνει όταν χρησιμοποιούνται μια ακολουθία από τους χαρακτήρες CRLF σε ένα query προκειμένου ο επιτιθέμενος να εισάγει δικό του query. Για παράδειγμα σε ένα HTTP response ο επιτιθέμενος θα μπορούσε να εισάγει μία ακολουθία από CRLF χαρακτήρες προκειμένου να προσθέσει ένα νέο header στο response ή να τερματίσει το header και να γράψει στο body του HTTP response (Veracode - CRLF INJECTION TUTORIAL: LEARN ABOUT CRLF INJECTION VULNERABILITIES AND PREVENTION,n.d.) (netsparker - CRLF Injection and HTTP Response Splitting Vulnerability, 23/05/2019).
- Cross-site Scripting (XSS) : Συμβαίνει όταν ο επιτιθέμενος εισάγει σε μια σελίδα ή σε ένα web application malicious κώδικα. Έτσι όταν το θύμα επισκεφτεί την σελίδα ο browser του θα εκτελέσει τον κώδικα του επιτιθέμενου καθώς «πιστεύει» ότι κώδικας προήλθε από έγκυρη πηγή που είναι η σελίδα που επισκέφτηκε. Αυτό μπορεί να έχει ως αποτέλεσμα ο επιτιθέμενος να πάρει πληροφορίες που αφορούν το θύμα όπως για παράδειγμα cookies (acunetix – Cross-site Scripting (XSS), n.d.) .
- Email injection : Πολλές φορές ιστοσελίδες και web εφαρμογές χρησιμοποιούν πλατφόρμα επικοινωνίας (contact form). Σε πολλές από αυτές τις περιπτώσεις το input που θα βάλει ο χρήστης δεν ελέγχεται με αποτέλεσμα κάποιος επιτιθέμενος να μπορεί να το εκμεταλλευτεί και για παράδειγμα να στείλει ανώνυμα mails σε άλλους users για spam ή για phising (acunetix - What Are Email Injection Attacks, 27/06/2019).

- Host Header injection: Αρχικά θα εξηγήσουμε τι είναι Host Header. Όταν ένας web server χρησιμοποιεί την ίδια IP διεύθυνση για διαφορετικές ιστοσελίδες ή web εφαρμογές τότε χρησιμοποιεί το host header για να διευκρινίσει ποια ιστοσελίδα ή web εφαρμογή θα εξυπηρετήσει ένα HTTP request. Όταν όμως ο server δεν αναγνωρίζει το host header τότε περνάει στην πρώτη web εφαρμογή στην λίστα του. Έτσι ο επιτιθέμενος μπορεί να «πειράξει» το πειράξει το host header και να το στείλει στην web εφαρμογή (acunetix - What is a Host Header Attack?, 25/04/2017) (Medium - HOST HEADER INJECTION ATTACK, 12/04/2018).
- LDAP injection : Αρχικά θα εξηγήσουμε τι είναι το LDAP. Το LDAP (Lightweight Directory Access Protocol) είναι ένα client/server πρωτόκολλο που χρησιμοποιείται για την επεξεργασία directories σε ένα IP δίκτυο. Οι web εφαρμογές που χρησιμοποιούν το LDAP αναμένουν κάποιο input από τον χρήστη και στο back-end σχηματίζεται ένα LDAP query για να προβάλει ή να επεξεργαστεί δεδομένα. Για παράδειγμα το LDAP μπορεί να χρησιμοποιηθεί προκειμένου να προσφέρει ένα «μέρος» για να αποθηκεύσουμε usernames και passwords. Έτσι η επίθεση αυτή μπορεί να συμβεί όταν web εφαρμογές που χρησιμοποιούν το LDAP περιμένουν input από τον χρήστη αλλά δεν ελέγχουν επαρκώς το input με αποτέλεσμα ο επιτιθέμενος να εισάγει στο input κάποιο κώδικα που θα εκτελεστεί στο LDAP query που θα σχηματιστεί στο backend. Για παράδειγμα εάν έχουμε μία μπάρα αναζήτησης για usernames ο επιτιθέμενος θα μπορούσε να εισάγει κώδικα(όπως «*») για να του εμφανιστούν όλα τα usernames της βάσης (GeeksforGeeks -LDAP and LDAP Injection/Prevention, 12/07/2019).
- OS Command injection : Η επίθεση αυτή, γνωστή και ως shell injection, συμβαίνει όταν ο επιτιθέμενος εκτελεί εντολές λειτουργικού συστήματος (π.χ. ping, cat, ls) σε μία web εφαρμογή. Για να συμβεί μία τέτοια επίθεση θα πρέπει η εφαρμογή να χρησιμοποιεί κώδικα που επιτρέπει την χρήση system calls, όπου έχουμε input από τον χρήστη. Αξίζει να σημειωθεί ότι η επίθεση αυτή είναι ανεξάρτητη από την γλώσσα προγραμματισμού που χρησιμοποιείται στην web εφαρμογή. Για παράδειγμα, μία τέτοια επίθεση μπορεί να γίνει μέσω ενός URL request που στέλνετε στην εφαρμογή. Έστω ότι αναζητείται ένα προϊόν σε ένα μαγαζί. Για να εμφανιστεί θα σχηματιστεί ένα URL με το id του. Στο URL αυτό μετά το id ο επιτιθέμενος θα μπορούσε να εισάγει εντολές λειτουργικού συστήματος χρησιμοποιώντας τον χαρακτήρα “&” (PortSwigger - OS command injection, n.d.) (acunetix -What Is OS Command Injection, 1/07/2019).
- XPath injection : Αρχικά θα εξηγήσουμε τι είναι το XPath. Όταν χρησιμοποιούμε XML η επεξεργασία των δεδομένων γίνεται μέσω προτάσεων

που αποτελούν το XPath. Επομένως η επίθεση αυτή συμβαίνει όταν ο επιτιθέμενος χρησιμοποιεί το input μιας ιστοσελίδας για σχηματίσει ένα XPath query και να ανακτήσει δεδομένα που ένας απλός χρήστης δεν έχει πρόσβαση (w3schools.com - XML and XPath, n.d.).

(acunetix - What Are Injection Attacks, 18/04/2019).

2.2.2. Phishing επιθέσεις:

Phishing είναι ένα είδος επίθεσης κατά την οποία ο επιτιθέμενος στέλνει στο θύμα ένα μήνυμα, email ή γενικότερα κάποιο link (σύνδεσμο) προκειμένου το θύμα να πατήσει το link αυτό. Μόλις το θύμα πατήσει το link τότε εκτελείται ένα κακόβουλο πρόγραμμα του επιτιθέμενου που έχει ως αποτέλεσμα να κλέψει τα δεδομένα του θύματος. Είδη phishing επιθέσεων:

- Email phishing: Αποτελεί το πιο δημοφιλές είδος phishing επίθεσης. Κατά την διάρκεια της επίθεσης αυτής ο επιτιθέμενος στέλνει mail στο θύμα παρουσιάζοντας τον εαυτό του ως κάποια γνωστή μάρκα ή επιχείρηση προκειμένου να ξεγελάσει το θύμα και να πατήσει στο κακόβουλο link (phoenxNAP - Preventing a Phishing Attack : How to Identify Types of Phishing, 11/01/2019).
- Spear phishing: Η επίθεση αυτή στέλνει mails σε συγκεκριμένα θύματα σε αντίθεση με το email phishing όπου στέλνονται μαζικά mails σε θύματα χωρίς κάποιο συγκεκριμένο χαρακτηριστικό. Η επίθεση spear phishing λοιπόν στέλνει mails τα οποία φαίνονται ότι έρχονται από κάποιο οικείο πρόσωπο του θύματος. Για να επιτευχθεί κάτι τέτοιο ο επιτιθέμενος πρέπει να αποκτήσει πληροφορίες για την προσωπική ζωή του θύματος. Η επίθεση spear phishing είναι σύνηθες να χρησιμοποιείται για να στοχεύσει υψηλόβαθμα στελέχη μίας επιχείρησης τροποποιώντας το mail έτσι ώστε να φαίνεται ότι στάλθηκε από κάποιον υπάλληλο της εταιρείας (Digital Guardian - What is Spear-phishing? Defining and Differentiating Spear-phishing from Phishing, 06/10/2020).
- Whaling: Η επίθεση αυτή στοχεύει σε υψηλόβαθμα στελέχη εταιρειών. Λόγω του κύρους του θύματος η επίθεση αυτή ονομάστηκε whaling. Κατά την διάρκεια της επίθεσης αυτής ο επιτιθέμενος στέλνει mail το οποίο έχει σχεδιαστεί συγκεκριμένα για το θύμα με βάση τις εμπειρίες και τις συνδέσεις του θύματος. Επομένως γίνεται αντιληπτό ότι μία τέτοιου είδους επίθεση απαιτεί χρόνο και γνώσεις για το θύμα (Digital Guardian - What is a Whaling Attack? Defining and Identifying Whaling Attacks, 27/07/2017).
- Clone Phishing: Κατά την επίθεση αυτή ο επιτιθέμενος κατασκευάζει μία ιστοσελίδα η οποία παρουσιάζει αρκετές ομοιότητες με κάποια ιστοσελίδα

που επισκέπτεται συχνά το θύμα και με τον τρόπο αυτό προσπαθεί να ξεγελάσει το θύμα ότι mail που στάλθηκε ήρθε από την πραγματική ιστοσελίδα. Αν το θύμα πιστέψει ότι το mail στάλθηκε από την πραγματική ιστοσελίδα και επισκεφθεί εν τέλει την ιστοσελίδα που αναγράφεται στο mail τότε ο επιτιθέμενος μπορεί να κλέψει τα προσωπικά δεδομένα του χρήστη (phoenixNAP - Preventing a Phishing Attack : How to Identify Types of Phishing, 11/01/2019).

- Phone and Text Phishing: Μερικές επιχειρήσεις αντί για mails χρησιμοποιούν το κινητό προκειμένου να ενημερώσουν έναν χρήστη για ένα γεγονός. Κατά την επίθεση Phone and Text Phishing ένας επιτιθέμενος εκμεταλλεύεται τον παραπάνω τρόπο λειτουργίας των επιχειρήσεων αυτών και στέλνει παραπλανητικά μηνύματα με κακόβουλους συνδέσμους (links) στον χρήστη με σκοπό να κλέψει προσωπικά δεδομένα του χρήστη (phoenixNAP - Preventing a Phishing Attack : How to Identify Types of Phishing, 11/01/2019).
- Social Media Phishing: Κατά την επίθεση αυτή ο επιτιθέμενος στοχεύει τα social media όπως Facebook προκειμένου να αποκτήσει προσωπικές πληροφορίες για τους χρήστες. Για παράδειγμα μπορεί να στείλει αίτημα φιλίας σε κάποιον χρήστη και να του ζητήσει να συμπληρώσει κάποιο quiz το οποίο να περιέχει ερωτήσεις σχετικά με την προσωπική του ζωή (phoenixNAP - Preventing a Phishing Attack : How to Identify Types of Phishing, 11/01/2019).
- False or Fake Advertisements: Οι ιστοσελίδες χρησιμοποιούν πολλές διαφημίσεις προκειμένου να κερδίσουν λεφτά. Ένας επιτιθέμενος μπορεί να εκμεταλλευτεί το γεγονός αυτό δημιουργώντας μία ψεύτικη διαφήμιση η οποία περιέχει κακόβουλο κώδικα. Έτσι όταν χρήστης πατήσει πάνω στην ψεύτικη διαφήμιση ο κακόβουλος κώδικας θα εκτελεστεί στον υπολογιστή του χρήστη και ο επιτιθέμενος θα αποκτήσει είτε έλεγχο του υπολογιστή είτε προσωπικά δεδομένα του χρήστη (phoenixNAP - Preventing a Phishing Attack : How to Identify Types of Phishing, 11/01/2019).

2.2.3. Malware επιθέσεις:

Malware είναι ένας κακόβουλος κώδικας ο οποίος έχει σχεδιαστεί για να εισέλθει και να αλληλοεπιδράσει με το σύστημα του υπολογιστή κάποιου χρήστη χωρίς την συναίνεση του χρήστη. Είδη Malware επιθέσεων:

- Ransomware: Κατά την διάρκεια της επίθεσης αυτής ένας κακόβουλος κώδικας μπλοκάρει πρόσβαση τα δεδομένα του χρήστη που βρίσκονται στον υπολογιστή του και ζητάει λίτρα προκειμένου ο χρήστης να αποκτήσει και πάλι πρόσβαση στα δεδομένα του.

- Rootkit: Rootkit είναι ένα κακόβουλο πρόγραμμα το οποίο εκτελείται στον υπολογιστή του θύματος και επιτρέπει στον επιτιθέμενο να εκτελεί άλλα αρχεία στον υπολογιστή του θύματος από απόσταση (remote).
- Spyware: Το spyware είναι ένας κακόβουλος κώδικας ο οποίος όταν εκτελεστεί στον υπολογιστή του θύματος δίνει την δυνατότητα στον επιτιθέμενο να κατασκοπεύει τα δεδομένα του χρήστη.
- Trojan Horse: Το Trojan Horse είναι ένα κακόβουλο αρχείο το οποίο παρουσιάζεται στον χρήστη ως κάποιο έγκυρο αρχείο και προσπαθεί να τον ξεγελάσει προκειμένου ο χρήστης να το εγκαταστήσει στον υπολογιστή του. Μόλις εγκατασταθεί επιτρέπει στον επιτιθέμενο να αποκτήσει πρόσβαση στον υπολογιστή του θύματος με αποτέλεσμα να εγκαταστήσει και άλλα κακόβολα προγράμματα στον υπολογιστή ή να κλέψει δεδομένα του χρήστη που βρίσκονται στον υπολογιστή.
- Virus: Είναι η πιο συνηθισμένη επίθεση malware. Το κακόβουλο πρόγραμμα με το όνομα Virus μόλις εγκατασταθεί σε έναν υπολογιστεί, δημιουργεί αντίγραφα του εαυτού του και διαδίδεται σε άλλους υπολογιστές.

(Comodo CyberSecurity - What is a Malware attack? | Different Types of Malware Attacks)

2.2.4. Distributed Denial of Service (DDoS) επιθέσεις:

DDos επίθεση είναι μία επίθεση κατά την οποία ο επιτιθέμενος προσπαθεί να σταματήσει την λειτουργία μίας υπηρεσίας που προσφέρεται σε χρήστες. Μία τέτοιου είδους επίθεση συνήθως εκτελείται από περισσότερες από μία συσκευές λόγω της ανάγκης να σταλθεί μεγάλος αριθμός πακέτων μέσα σε λίγο χρονικό διάστημα. Συνηθισμένα είδη DDos επιθέσεων:

- UDP flood: Κατά την επίθεση αυτή το θύμα δέχεται μεγάλο αριθμό UDP πακέτων σε διαφορετικές θύρες προορισμού κάθε φορά με αποτέλεσμα το θύμα να περιμένει δεδομένα στην κάθε θύρα και άρα η θύρα αυτή να μην μπορεί να χρησιμοποιηθεί για άλλον χρήστη.
- ICMP (Ping) flood: Κατά την επίθεση αυτή το θύμα δέχεται μεγάλο αριθμό ICMP Echo Request (ping) πακέτα σε μικρό χρονικό διάστημα. Το θύμα προσπαθεί να απαντήσει στα πακέτα αυτά με ICMP Echo Reply πακέτα με αποτέλεσμα η απόδοση του συστήματος του θύματος να μειώνεται δραματικά.
- TCP SYN flood: Αρχικά είναι σημαντικό να ορισθεί η έννοια “three-way handshake”. Η έννοια αφορά την διαδικασία δημιουργίας TCP σύνδεσης μεταξύ του αποστολέα και του παραλήπτη των πακέτων που θα σταλθούν. Κατά την παραπάνω διαδικασία ο αποστολέας στέλνει ένα TCP SYN request

πακέτο, ο παραλήπτης απαντάει με ένα TCP SYN-ACK πακέτο και τέλος ο αποστολέας στέλνει ένα TCP ACK πακέτο που επιβεβαιώνει ότι έλαβε το TCP SYN-ACK πακέτο. Μόλις ο παραλήπτης λάβει το TCP ACK πακέτο δημιουργείται η σύνδεση. Μέχρι όμως να λάβει το πακέτο αυτό ο παραλήπτης περιμένει την απάντηση. Κατά την επίθεση TCP SYN flood λοιπόν το θύμα δέχεται μεγάλο αριθμό TCP SYN request πακέτα σε μικρό χρονικό διάστημα. Αυτό έχει ως αποτέλεσμα το θύμα να προσπαθεί να απαντήσει με πακέτα TCP SYN-ACK περιμένοντας στην συνέχεια απάντηση από τον επιτιθέμενο. Ωστόσο ο επιτιθέμενος δεν στέλνει ποτέ πακέτο TCP ACK με αποτέλεσμα το θύμα να περιμένει πακέτα τα οποία δεν θα φτάσουν ποτέ. Όσο πιο πολλά TCP SYN πακέτα σταλθούν στο θύμα τόσο περισσότερα TCP ACK πακέτα θα περιμένει να λάβει ως απάντηση με αποτέλεσμα να οδηγείται σε διακοπή παροχής των υπηρεσιών του σε άλλου χρήστες (denial of service) (imperva – TCP SYN Flood, n.d.).

- Ping of Death: Κατά την διάρκεια αποστολής ενός μεγάλου πακέτου το πακέτο αυτό διασπάται σε μικρότερα πακέτα τα οποία όταν όλα φτάσουν στον προορισμό ενώνονται και σχηματίζουν το αρχικό πακέτο. Είναι σημαντικό επίσης να αναφερθεί ότι το μέγιστο μέγεθος ενός IP πακέτου είναι 65,535 bytes. Κατά την διάρκεια λοιπόν της επίθεσης Ping of Death στέλνονται πακέτα τα οποία χωρίζονται σε μικρότερα πακέτα. Τα μικρότερα όμως αυτά πακέτα περιέχουν κακόβουλα δεδομένα με σκοπό όταν ενωθούν στον προορισμό να προκύψει πακέτο μεγαλύτερο από 65,535 bytes. Αυτό έχει ως αποτέλεσμα η μνήμη που διατίθεται για τα πακέτα να μην είναι αρκετή (overflow) και άρα να σταματάει η παροχή της υπηρεσίας για πακέτα που στέλνονται από έγκυρους χρήστες.
- Slowloris επίθεση: Κατά την επίθεση αυτή ο επιτιθέμενος στέλνει μεγάλο αριθμό HTTP πακέτων τα οποία όμως δεν είναι ολοκληρωμένα. Αυτό έχει ως αποτέλεσμα να δημιουργούνται πολλές συνδέσεις οι οποίες παραμένουν ανοιχτές επειδή το θύμα περιμένει να σταλθεί η υπόλοιπη πληροφορία για HTTP πακέτα. Επομένως το σύστημα του θύματος φτάνει το όριο για ταυτόχρονες ανοιχτές συνδέσεις και άρα αρνείται να δημιουργήσει νέες συνδέσεις οι οποίες ζητούνται από έγκυρους χρήστες.

(imperva - DDos Attack Types & Mitigation Methods, n.d.)

2.2.5. Man-in-the-Middle (MITM) επίθεση:

Κατά τις επιθέσεις Man-in-the-Middle ο επιτιθέμενος «κρυφακούει» την επικοινωνία μεταξύ ενός χρήστη και του συστήματος με αποτέλεσμα να μπορεί να αποκτήσει πληροφορίες για είτε τον χρήστη είτε το σύστημα.

Τα παραπάνω είδη επιθέσεων αφορούν τις πιο συνηθισμένες κυβερνο-επιθέσεις που πραγματοποιούνται και για τον λόγο αυτό είναι σημαντικό να υπάρχουν τρόποι αντιμετώπισης αυτών.

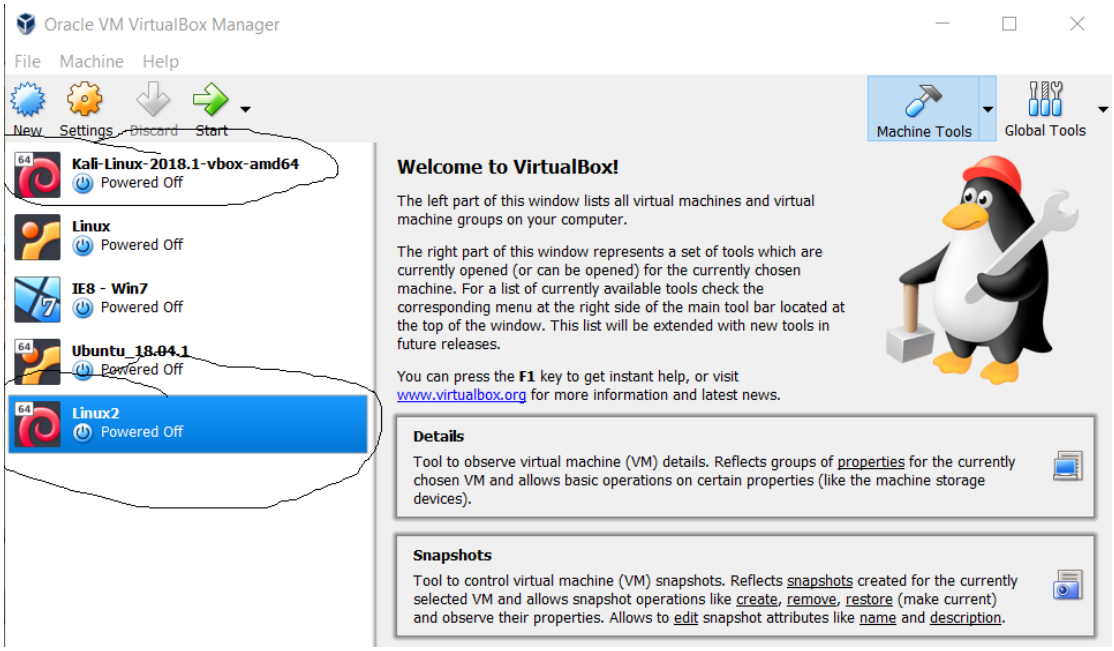
(phoenixNap - 17 Types of Cyber Attacks To Secure Your Company From in 2020, 21/02/2019)

Κεφάλαιο 3. Υλοποίηση Συστήματος

3.1. Υλοποίηση του Συστήματος

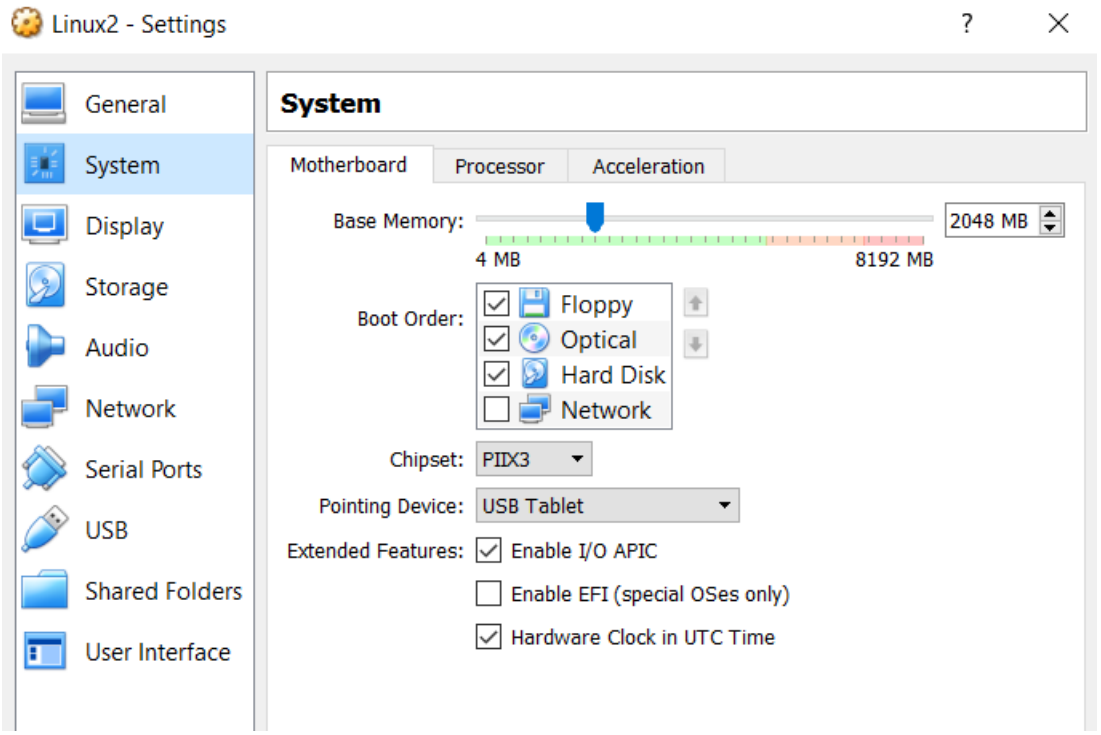
3.1.1. Περιβάλλον Προσομοίωσης

Για την αναπαραγωγή κυβερνο-επιθέσεων είναι απαραίτητη η ύπαρξη ενός ασφαλούς περιβάλλοντος το οποίο δεν περιέχει δεδομένα πραγματικών χρηστών και είναι αποκομμένο από το διαδίκτυο. Για τον λόγο αυτό χρησιμοποιήθηκε η τεχνολογία του εικονικού υπολογιστή ή αλλιώς Virtual Machine. Στην παρούσα διπλωματική χρησιμοποιήθηκε η εφαρμογή Oracle VM VirtualBox. Μέσω της εφαρμογής αυτής δημιουργήθηκαν δύο εικονικοί ηλεκτρονικοί υπολογιστές:

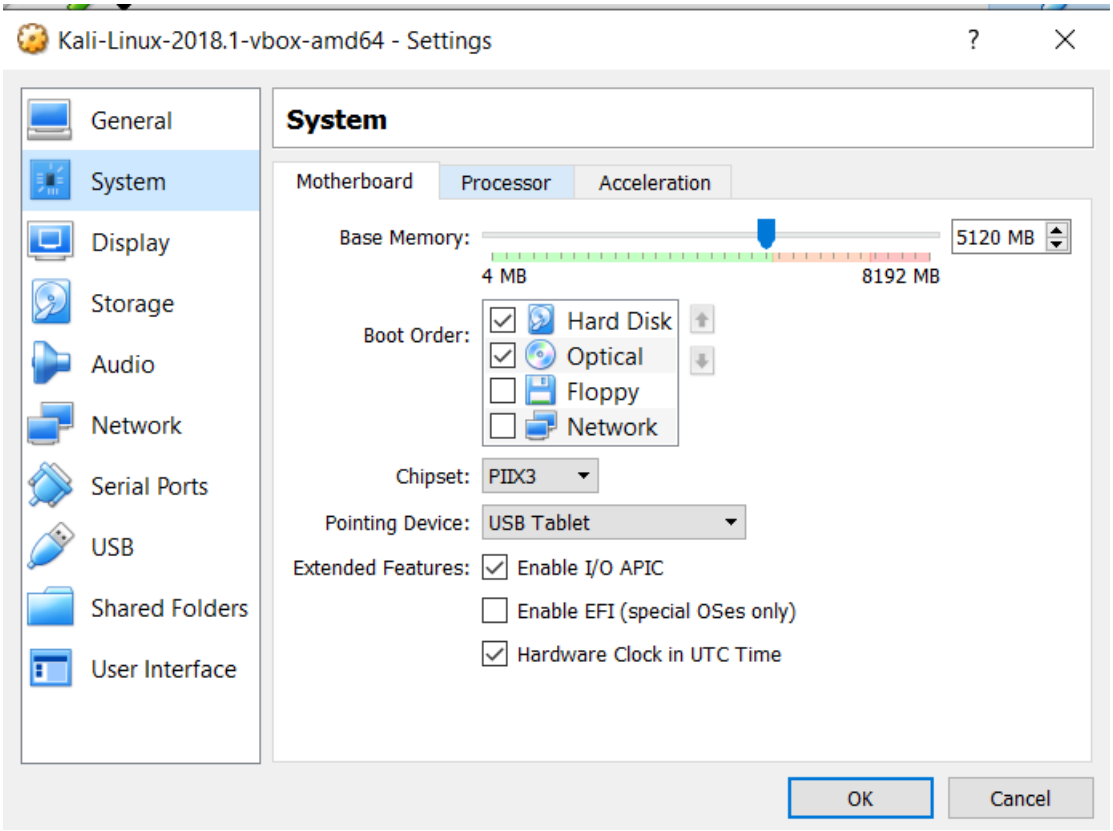


Εικόνα 5. User Interface της εφαρμογής Oracle VM VirtualBox

Στην παραπάνω εικόνα φαίνεται η αρχική σελίδα της εφαρμογής Oracle VM VirtualBox, στην οποία εμφανίζονται όλα τα εικονικά μηχανήματα που έχουν δημιουργηθεί. Τα δύο μηχανήματα που περιβάλλονται από κύκλο είναι αυτά τα οποία έχουν χρησιμοποιηθεί για την υλοποίηση του πειράματος. Το πρώτο μηχάνημα είναι το μηχάνημα που αναπαριστά τον επιτιθέμενο, ενώ το δεύτερο αυτό που αναπαριστά το θύμα μίας κυβερνο-επίθεσης. Είναι σημαντικό να σημειωθεί ότι τα μηχανήματα αυτά δεν έχουν απεριόριστη μνήμη που σημαίνει ότι είναι αδύνατον να καταγράφουν κυβερνο-επιθέσεις με μεγάλο αριθμό δεδομένων για μεγάλο χρονικό διάστημα.



Εικόνα 6. Κατανομή μνήμης του εικονικού μηχανήματος του θύματος



Εικόνα 7. Κατανομή μνήμης του εικονικού μηχανήματος του επιτιθέμενου

Η πρώτη εικόνα αφορά το εικονικό μηχάνημα που αναπαριστά το θύμα μίας κυβερνο-επίθεσης, ενώ η δεύτερη εικόνα αφορά τον επιτιθέμενο. Από τις παραπάνω εικόνες γίνεται αντιληπτό ότι το μέγιστο όριο της μνήμης που είναι δυνατόν να διατεθεί σε ένα εικονικό μηχάνημα είναι 8 Giga Bytes. Η τιμή αυτή διαφέρει ανάλογα με τα χαρακτηριστικά του ηλεκτρονικού υπολογιστή στον οποίο είναι εγκαταστημένη η εφαρμογή που προσφέρει τα εικονικά μηχανήματα. Παρατηρείται ότι το θύμα έχει μνήμη ίση με 2 Giga Bytes και ο επιτιθέμενος 5 Giga Bytes. Άρα συνολικά στα δύο εικονικά μηχανήματα διατίθενται 7 Giga Bytes, τιμή που δεν ξεπερνά το ανώτατο όριο. Τα τιμή αυτή δικαιολογείται από το γεγονός ότι τα δύο εικονικά μηχανήματα λειτουργούν ταυτόχρονα για τις ανάγκες του πειράματος, οπότε δεν είναι επιθυμητό το άθροισμα των χαρακτηριστικών των δύο μηχανημάτων να ξεπερνάει το ανώτατο όριο.

Ένα ακόμα χαρακτηριστικό που πρέπει να επιλεχθεί για την σωστή λειτουργία των δύο εικονικών μηχανημάτων είναι ο τρόπος που διασύνδεσης του δικτύου στο οποίο ανήκουν. Οι επιλογές είναι οι εξής:

- **Not attached**. Με την επιλογή αυτή το εικονικό μηχάνημα δεν έχει πρόσβαση στο δίκτυο.
- **NAT**. Το εικονικό μηχάνημα που έχει την επιλογή αυτή στις ρυθμίσεις του δικτύου του, μπορεί να συνδεθεί με εξωτερικά δίκτυα όπως το διαδίκτυο αλλά δεν μπορεί να επικοινωνήσει με άλλα εικονικά μηχανήματα.
- **NAT Network**. Με την επιλογή NAT Network το εικονικό μηχάνημα μπορεί να επικοινωνήσει με άλλα εικονικά μηχανήματα στο δίκτυο αλλά και με εξωτερικά δίκτυα όπως το Διαδίκτυο. Ωστόσο τόσο συσκευές που ανήκουν σε κάποιο εξωτερικό δίκτυο όσο και η συσκευή στην οποία εκτελείται η εφαρμογή Oracle VM VirtualBox και προσφέρει μέρος των χαρακτηριστικών στα εικονικά μηχανήματα, αλλιώς λεγόμενη και ως host, δεν μπορούν να επικοινωνήσουν με τα εικονικά μηχανήματα μέσω του δικτύου NAT Network.
- **Bridged Adapter**. Με την επιλογή αυτή το εικονικό μηχάνημα μπορεί να επικοινωνήσει με τον host αλλά και με συσκευές, εικονικές ή φυσικές, που βρίσκονται στο φυσικό δίκτυο στο οποίο βρίσκεται και το host μηχάνημα. Ακόμα το εικονικά μηχάνημα αποκτά πρόσβαση και σε εξωτερικά δίκτυα. Αξίζει να σημειωθεί ότι με τον όρο φυσικό δίκτυο εννοείται το δίκτυο υπολογιστών που βρίσκονται στον χώρο που είναι το host μηχάνημα και συνδέονται μεταξύ τους με κάποιο καλώδιο ή με ασύρματη σύνδεση με την βοήθεια ενός δρομολογητή (host).
- **Internal Network**. Εικονικά μηχανήματα ενός host που έχουν επιλογή δικτύου Internal Network μπορούν μόνο να επικοινωνήσουν μεταξύ τους. Δηλαδή δεν είναι δυνατό να συνδεθούν με το host μηχάνημα ή με μηχανήματα που ανήκουν είτε στο φυσικό δίκτυο είτε σε εξωτερικό δίκτυο.

- **Host-only Adapter**. Η επιλογή αυτή επιτρέπει την επικοινωνία του host με τα εικονικά μηχανήματα, καθώς και την επικοινωνία μεταξύ εικονικών μηχανημάτων που έχουν σαν επιλογή δικτύου Host-only Adapter. Ωστόσο τα εικονικά μηχανήματα τα οποία ανήκουν σε τέτοιου είδους δικτύου, δεν μπορούν να επικοινωνήσουν με συσκευές ενός εξωτερικού δικτύου.

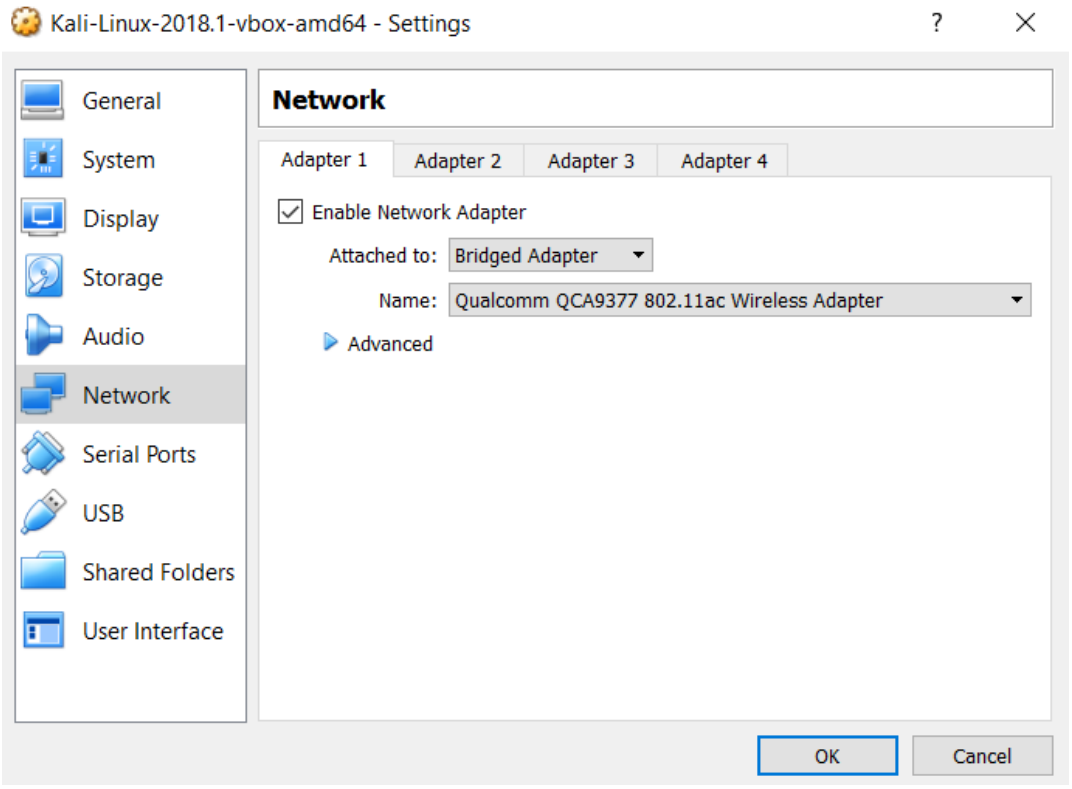
(Nakivo - VirtualBox Network Settings: Complete Guide, 16/07/2019)

Παρακάτω φαίνεται μία σύντομη περιγραφή για την επικοινωνία των εικονικών μηχανημάτων ανάλογα με το είδος δικτύου που έχει επιλεχθεί:

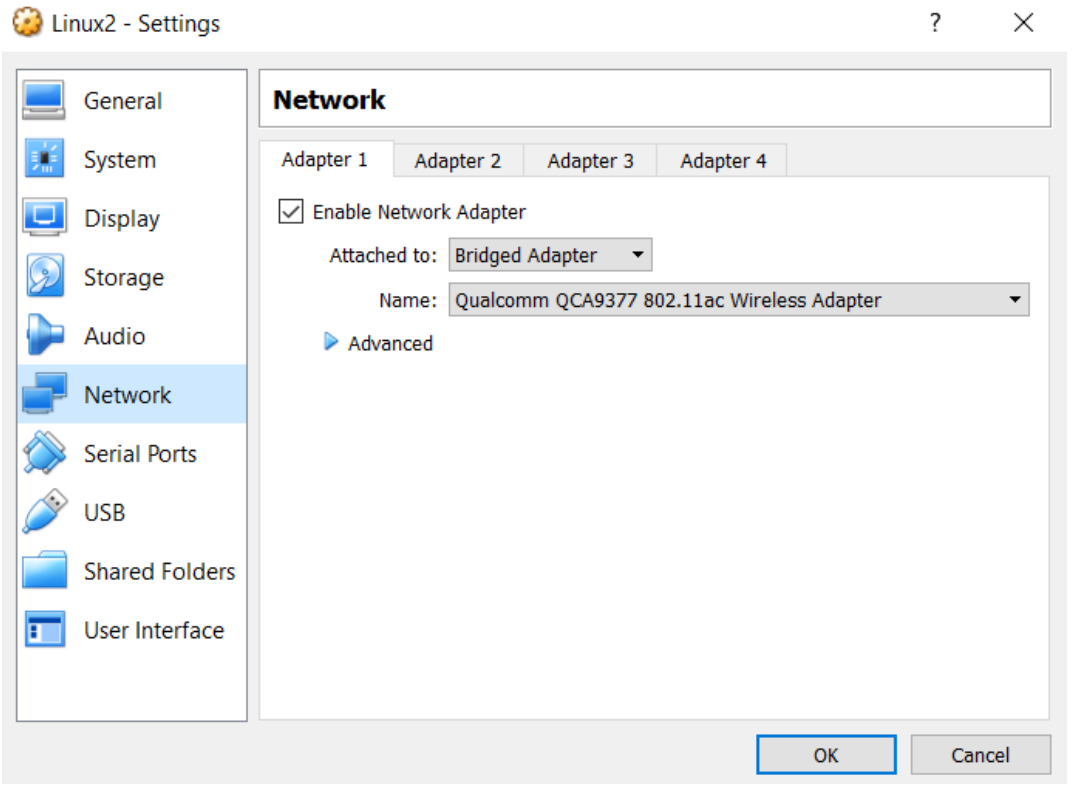
	VM ↔ VM	VM → Host	VM ← Host	VM → LAN	VM ← LAN
Not attached	–	–	–	–	–
NAT	–	+	Port Forward	+	Port Forward
NAT Network	+	+	Port Forward	+	Port Forward
Bridged	+	+	+	+	+
Internal Network	+	–	–	–	–
Host-only	+	+	+	–	–

Εικόνα 8. VirtualBox Network settings comparison

Στο πείραμα για την παρούσα διπλωματική εργασία έχει επιλεχθεί το Bridged Adapter δίκτυο γιατί θεωρήθηκε ότι είναι αυτό που αναπαριστά καλύτερα την πραγματικότητα μιας και δεν έχει κάποιο περιορισμό στην επικοινωνία μεταξύ συσκευών, όπως μπορεί να γίνει αντιληπτό και από την παραπάνω εικόνα.



Εικόνα 9. VirtualBox Kali Linux Network Configuration

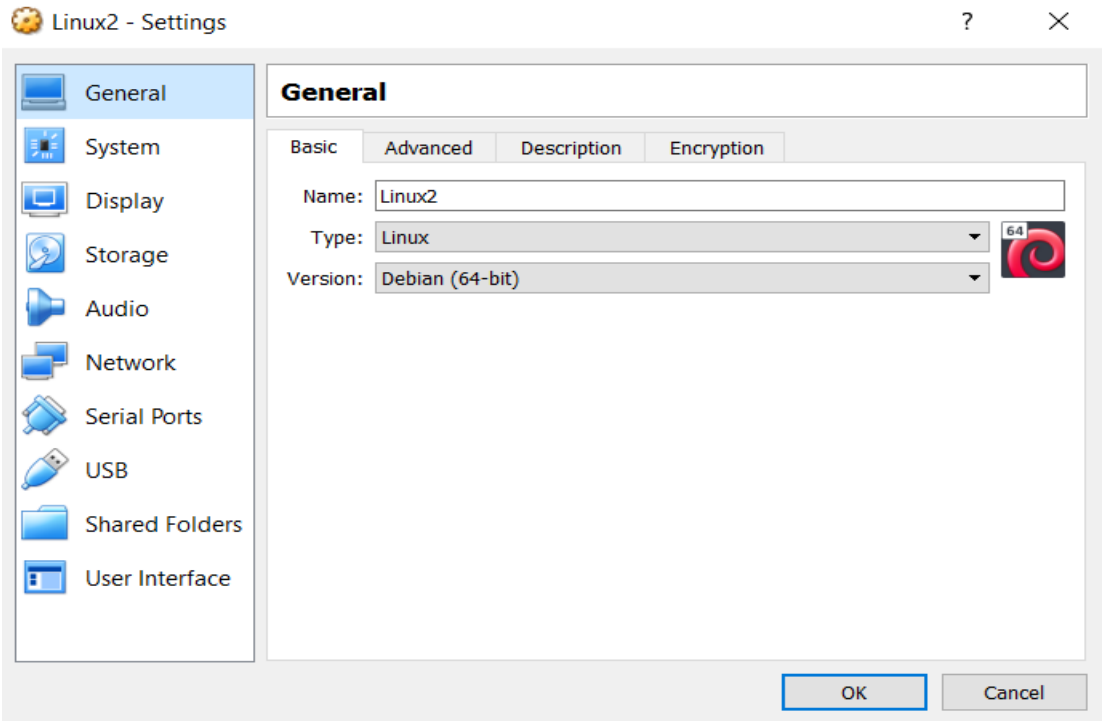


Εικόνα 10. Virtual Box Linux Network configuration

Η πρώτη εικόνα (9) αφορά τις ρυθμίσεις δικτύου του εικονικού μηχανήματος που θα λειτουργήσει ως επιτιθέμενος , ενώ η δεύτερη εικόνα (10) αφορά τις ρυθμίσεις δικτύου που θα λειτουργήσεις ως θύμα των κυβερνο-επιθέσεων που θα προσομοιωθούν. Και στις δύο εικόνες φαίνεται ότι έχει επιλεχθεί το Bridged Adapter δίκτυο. Επισημαίνεται ότι αν τα δύο εικονικά μηχανήματα δεν είχαν τις ίδιες ρυθμίσεις δικτύου τότε δεν θα ήταν δυνατή η επικοινωνία μεταξύ τους.

Το επόμενο βήμα για την δημιουργία ενός περιβάλλοντος προσομοίωσης για την αναπαραγωγή κυβερνο-επιθέσεων είναι η εγκατάσταση λειτουργικού συστήματος στα εικονικά μηχανήματα που δημιουργήθηκαν. Το λειτουργικό σύστημα είναι ένα είδος λογισμικού που είναι υπεύθυνο για την διαχείριση των χαρακτηριστικών της συσκευής μέσω της οποίας προσφέρει διάφορες υπηρεσίες στον χρήστη. Επομένως κάθε ηλεκτρονικός υπολογιστής χρειάζεται το λειτουργικό του σύστημα και το ίδιο ισχύει και για τα εικονικά μηχανήματα. Οι τρεις πιο γνωστές κατηγορίες λειτουργικών συστημάτων για ηλεκτρονικούς υπολογιστές είναι τα Windows, macOS και Linux.

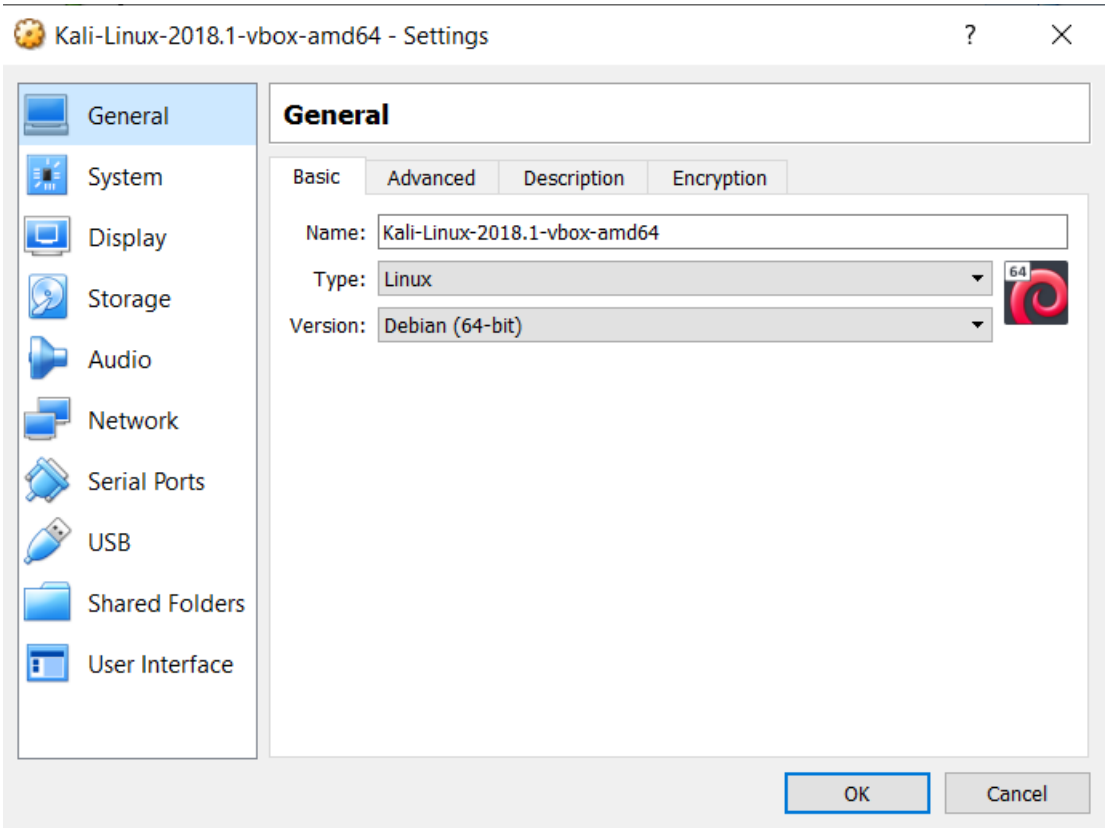
Για το πείραμα επιλέχθηκε το λειτουργικό σύστημα Linux για το θύμα διότι σε περιβάλλον Linux είναι πιο εύκολο να γίνει η ανάλυση των επιθέσεων λόγω της προγραμματιστικής γλώσσας που χρησιμοποιήθηκε. Παρακάτω φαίνεται το λειτουργικό σύστημα Linux που εγκαταστάθηκε στο εικονικό μηχάνημα του θύματος:



Εικόνα 11. VirtualBox Linux Operating system settings

Για το εικονικό μηχάνημα του επιτιθέμενου χρησιμοποιήθηκε μία ειδική κατηγορία του λειτουργικού συστήματος Linux , η οποία λέγεται Kali Linux και προσφέρει εργαλεία για την αναπαραγωγή κυβερνο-επιθέσεων. Παρακάτω φαίνεται το

λειτουργικό σύστημα Kali Linux το οποίο εγκαταστάθηκε στο εικονικό μηχάνημα του επιτιθέμενου:

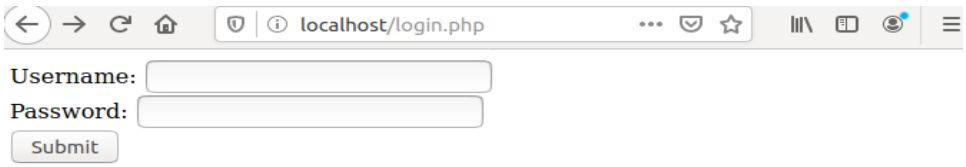


Εικόνα 12. VirtualBox Kali Linux Operating system settings

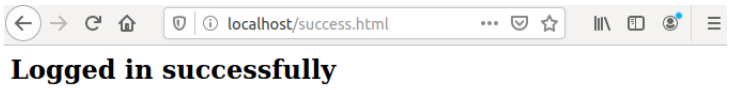
Για την υλοποίηση ορισμένων από τις επιθέσεις που επιλέχθηκαν ήταν απαραίτητη η δημιουργία μίας login φόρμας στην οποία ο χρήστης εισάγει το όνομα (username) και τον κωδικό (password) του. Εάν το username και το password είναι ο σωστός συνδυασμός τότε ο χρήστης μεταφέρεται σε μία άλλη σελίδα (success.html) που δηλώνει ότι επιβεβαιώθηκε ότι υπάρχει ο συγκεκριμένος χρήστης στην βάση δεδομένων. Ωστόσο εάν είναι λάθος είτε το username είτε το password τότε ο χρήστης μεταφέρεται σε μία σελίδα (fail.html) που δηλώνει ότι ο χρήστης δεν υπάρχει στην βάση δεδομένων.

Για την χρήση της login φόρμας χρησιμοποιήθηκε η προγραμματιστική γλώσσα php και HTML. Η γλώσσα HTML είναι υπεύθυνη για την δομή κάθε ιστοσελίδας και εμφανίζει οντότητες που είναι απαραίτητες για την λειτουργία της ιστοσελίδας όπως για παράδειγμα την εμφάνιση και την λειτουργία ενός κουμπιού. Η γλώσσα php είναι υπεύθυνη για την προσπέλαση των δεδομένων που εισάγει ο χρήστης και καθορίζει την συμπεριφορά της ιστοσελίδας ανάλογα με το αποτέλεσμα της προσπέλασης των δεδομένων. Για παράδειγμα , στο πείραμα, καθορίζει εάν ο χρήστης είναι έγκυρος ή όχι. Επίσης συνδέει την εφαρμογή με την βάση δεδομένων. Για την βάση δεδομένων του πειράματος χρησιμοποιήθηκε ο MySQL Server ο οποίος προσφέρει την

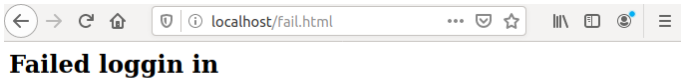
δυνατότητα διαχείρισης των δεδομένων μίας εφαρμογής. Η διαχείριση των δεδομένων γίνεται μέσω της γλώσσας MySQL που ανήκει στην κατηγορία των SQL (Structured Query Language) γλωσσών. Παρακάτω παρουσιάζονται η login φόρμα και οι σελίδες success.html και fail.html :



Εικόνα 13. PHP login form

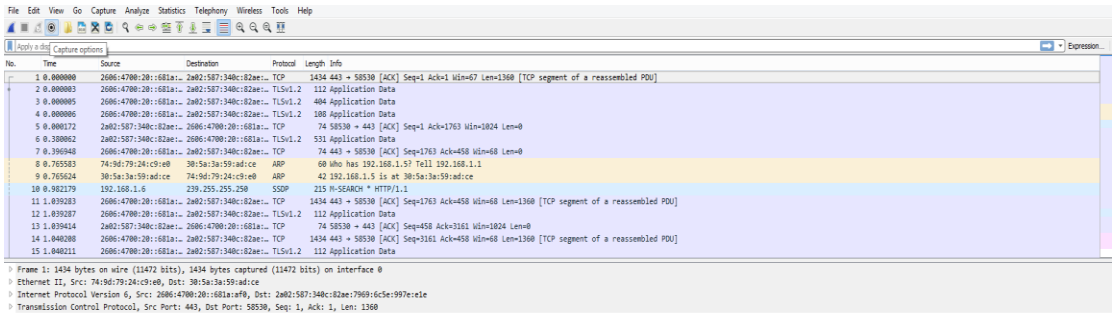


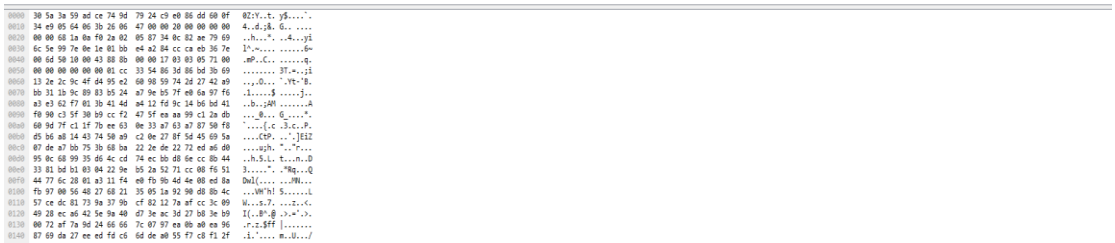
Εικόνα 14. Success login from PHP login form



Εικόνα 15. Failed login from PHP login form

Μετά την δημιουργία του περιβάλλοντος προσομοίωσης είναι αναγκαία η εύρεση ενός εργαλείου για την καταγραφή των πακέτων των κυβερνο-επιθέσεων προκειμένου να γίνει η ανάλυση αυτών. Το εργαλείο που χρησιμοποιήθηκε για τον σκοπό αυτό είναι το Wireshark. Το Wireshark προσφέρει την δυνατότητα να καταγράφει πακέτα τα οποία έχουν είτε ως αποστολέα είτε ως παραλήπτη το τρέχων μηχάνημα. Επομένως την στιγμή που αρχίζει να εκτελείται η επίθεση το Wireshark καταγράφει όλα τα πακέτα που σχετίζονται με την επίθεση αυτή. Παρακάτω παρουσιάζεται ένα παράδειγμα καταγραφής στο Wireshark:



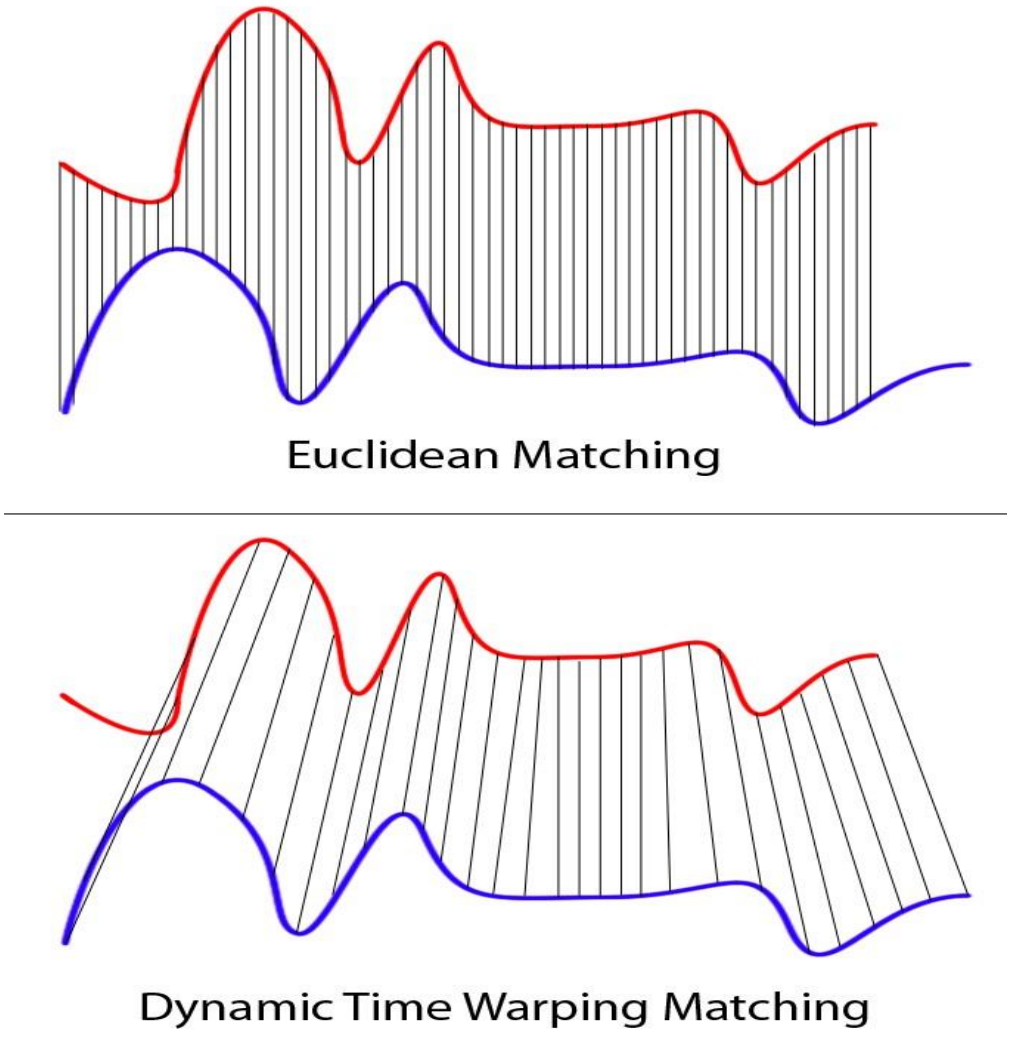


Εικόνα 16. Example of a capture of traffic in Wireshark

Το Wireshark προσφέρει επίσης την δυνατότητα εισαγωγής φίλτρων με σκοπό την εμφάνιση πακέτων τα οποία εκπληρώνουν κάποια συνθήκη, όπως για παράδειγμα πακέτα τα οποία έχουν ως αποστολέα μία ορισμένη IP διεύθυνση.

Ωστόσο το Wireshark χρησιμοποιείται απλά για την καταγραφή των πακέτων και όχι για την ανάλυση αυτών. Επομένως προκύπτει η ανάγκη εύρεσης ενός τρόπου ανάλυσης των πληροφοριών που προσφέρουν τα πακέτα τα οποία έχουν καταγραφεί μέσω του Wireshark. Για τον σκοπό αυτό χρησιμοποιήθηκε η μέθοδος Dynamic Time Warping (DTW).

Dynamic Time Warping είναι ένας αλγόριθμος ο οποίος χρησιμοποιείται για την μέτρηση της ομοιότητας μεταξύ δύο ακολουθιών διαφορετικού μεγέθους. Ο αλγόριθμος αυτός χρησιμοποιείται συνήθως για ακολουθίες που σχετίζονται με τον χρόνο. Παρακάτω φαίνονται δύο συναρτήσεις για τις οποίες εξετάζεται η ομοιότητά τους:



Εικόνα 17. Comparison between Euclidean and DTW

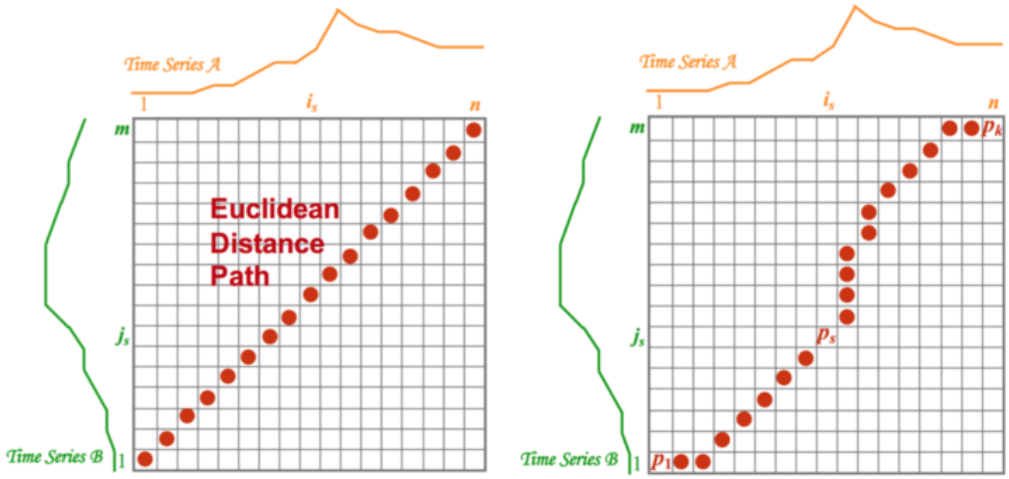
Γίνεται αντιληπτό ότι οι δύο συναρτήσεις ακολουθούν την ίδια ακολουθία αλλά η μπλε συνάρτηση έχει μεγαλύτερο μήκος από την κόκκινη. Αν εφαρμοστεί ένα προς ένα αντιστοίχιση, όπως στο πρώτο μέρος της εικόνας (Euclidean Matching) , τότε η αντιστοίχιση δεν είναι σωστή , καθώς το τελευταίο κομμάτι της μπλε συνάρτησης δεν αντιστοιχεί με κάποιο κομμάτι της κόκκινης συνάρτηση.

Με τον αλγόριθμο Dynamic Time Warping τα μέγιστα και τα ελάχιστα μίας συνάρτησης συνδέονται με τα αντίστοιχα σημεία της άλλης συνάρτησης. Με τον τρόπο αυτό ο υπολογισμός της ομοιότητας των δύο συναρτήσεων είναι πιο ακριβής.

Ο τρόπος με τον οποίο ο αλγόριθμος DTW υπολογίζει την ομοιότητα μεταξύ δύο ακολουθιών που σχετίζονται με τον χρόνο, διαφέρει από την ευκλείδεια απόσταση. Η ευκλείδεια απόσταση υπολογίζει την ελάχιστη απόσταση , και κατά συνέπεια την ομοιότητα, δύο συναρτήσεων στις οποίες τα παρόμοια σημεία , όπως οι κορυφές ίδιου ύψους, βρίσκονται στην ίδια χρονική στιγμή. Τι συμβαίνει, όμως, όταν τα σημεία αυτά είναι διασκορπισμένα σε διαφορετικές χρονικές στιγμές; Τότε η ευκλείδεια απόσταση θα δώσει λάθος αποτέλεσμα καθώς δεν θα θεωρήσει ότι οι δύο συναρτήσεις είναι όμοιες. Λύση στο πρόβλημα αυτό δίνει ο αλγόριθμος Dynamic Time Warping, ο οποίος αντιστοιχίζει σημεία, όπως οι κορυφές, της μίας συνάρτησης με τα όμοια σημεία της άλλης συνάρτησης ανεξάρτητα εάν βρίσκονται στην ίδια χρονική στιγμή ή όχι. Η απόσταση και συνεπώς η ομοιότητα, των δύο συναρτήσεων υπολογίζεται με την παρακάτω μέθοδο:

- Οι δύο ακολουθίες διαιρούνται σε ίσο αριθμό σημείων
- Για κάθε σημείο της πρώτης ακολουθίας υπολογίζεται η ευκλείδεια απόσταση του σημείου αυτού με όλα τα σημεία της δεύτερης ακολουθίας και αποθηκεύεται η μικρότερη απόσταση για το σημείο αυτό. Η ίδια διαδικασία πραγματοποιείται και για κάθε σημείο της δεύτερης ακολουθίας.
- Προστίθενται όλες οι μικρότερες αποστάσεις που έχουν αποθηκευτεί με αποτέλεσμα να προκύπτει η ομοιότητα των δύο ακολουθιών

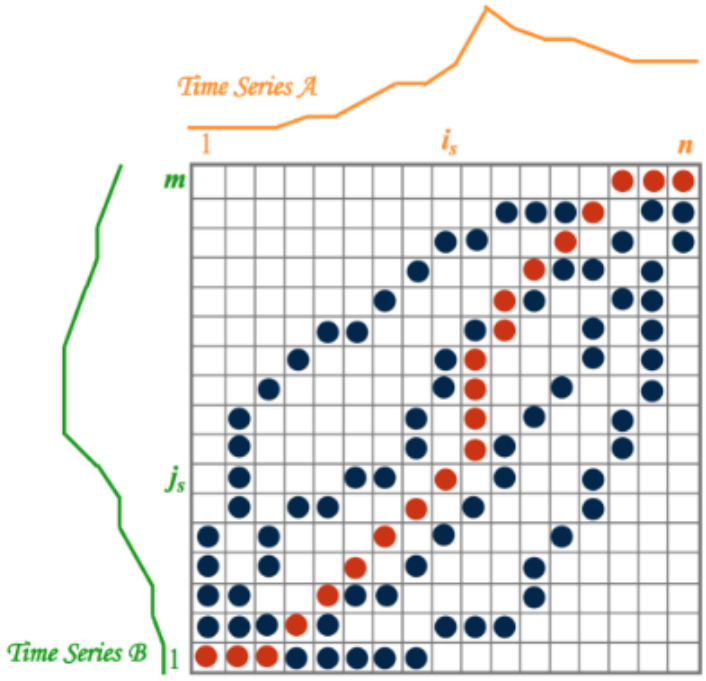
Παρακάτω φαίνονται οι αποστάσεις των δύο μεθόδων:



Εικόνα 18. Distance paths of Euclidean and DTW

Αριστερά έχει εφαρμοστεί η Ευκλείδεια απόσταση μεταξύ των ακολουθιών , ενώ δεξιά έχει εφαρμοστεί ο αλγόριθμος DTW.

Με τον αλγόριθμο DTW πέρα από την ομοιότητα δύο ακολουθιών προκύπτει και το βέλτιστο μονοπάτι (optimal path ή warping path) και αναπαριστά την χαμηλότερου κόστους συνάρτηση η οποία όταν εφαρμοστεί στις δύο ακολουθίες τις ενώνει σε μία νέα ακολουθία. Εάν ενώσουμε τις κόκκινες κουκίδες της δεξιάς εικόνας προκύπτει το βέλτιστο μονοπάτι. Κατά την διαδικασία υπολογισμού του βέλτιστου μονοπατιού προκύπτουν και άλλα μονοπάτια τα οποία όμως δεν είναι βέλτιστα καθώς αντιστοιχούν σε συναρτήσεις με υψηλότερο κόστος από αυτό της συνάρτησης του βέλτιστου μονοπατιού. Παρακάτω φαίνονται τα μονοπάτια:

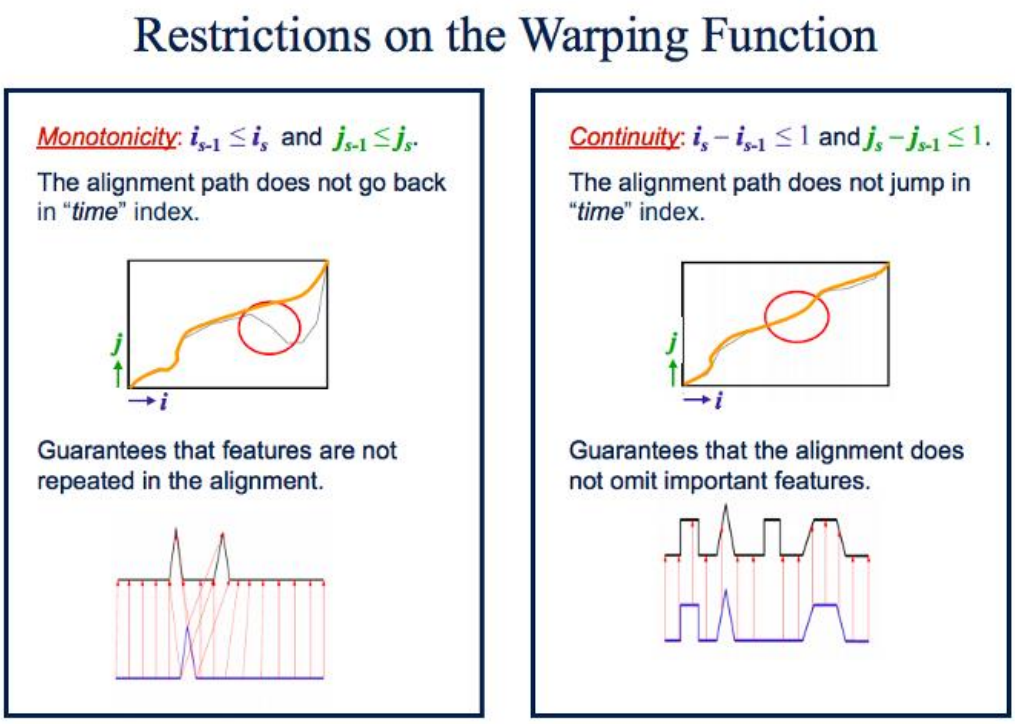


Εικόνα 19. Computed Paths of DTW

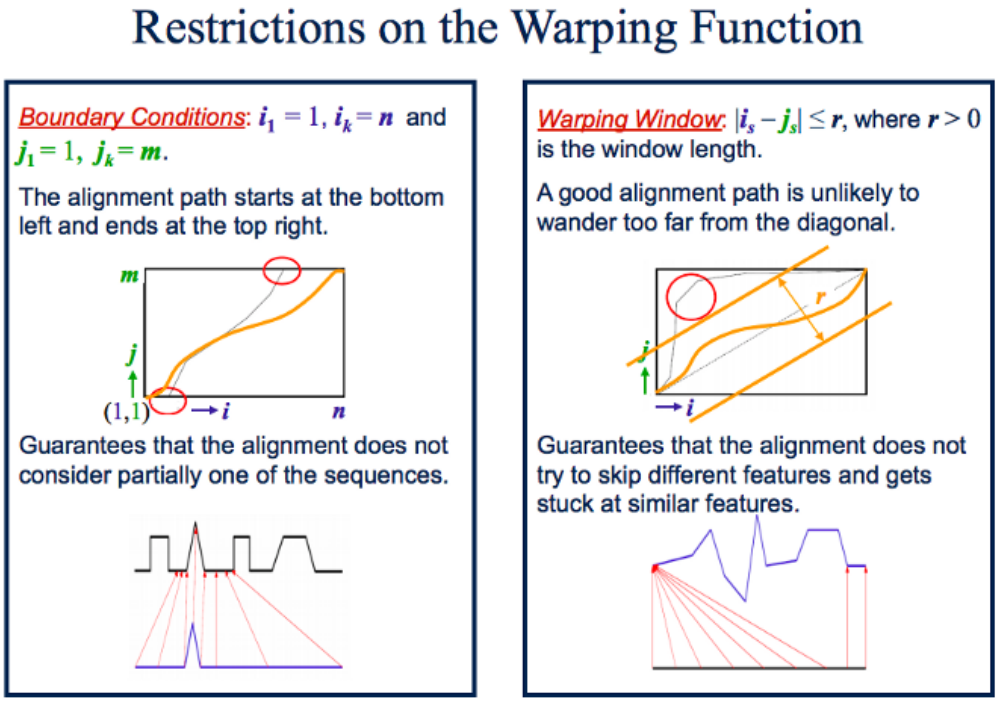
Τα μονοπάτια που υπολογίζονται από τον αλγόριθμο DTW , επομένως και οι συναρτήσεις στις οποίες αντιστοιχίζονται τα μονοπάτια, έχουν ορισμένους περιορισμούς:

- Το μονοπάτι δεν μπορεί να ταξιδεύει πίσω στο χρόνο. (Monotonicity)
- Το μονοπάτι πρέπει να είναι συνεχόμενο. (Continuity)
- Το μονοπάτι πρέπει να ξεκινάει από κάποιο σημείο που βρίσκεται κάτω αριστερά και να καταλήγει σε κάποιο σημείο που είναι μετατοπισμένο πιο δεξιά και ψηλά σε σχέση με το σημείο εκκίνησης. (Boundary conditions)
- Το μονοπάτι δεν πρέπει να αποκλίνει σημαντικά από την διαγώνιο. (Warping windows)
- Το μονοπάτι δεν πρέπει να είναι πολύ απότομο, δηλαδή δεν πρέπει να αυξάνεται απότομα , αλλά και ούτε πολύ ρηχό , δηλαδή δεν πρέπει να αυξάνεται με πολύ αργούς ρυθμούς. (Slop constraint)

Παρακάτω φαίνονται οι περιορισμοί:



Εικόνα 20. Restrictions on the Warping function



Εικόνα 21. More restrictions on the Warping function

Στο πείραμα έχουν χρησιμοποιηθεί δύο είδη προγραμμάτων για τον αλγόριθμο Dynamic Time Warping με διαφορετικό αποτέλεσμα το καθένα. Το πρώτο πρόγραμμα δέχεται ως είσοδο δύο πίνακες, για τους οποίους ερευνάται η ομοιότητά τους και δημιουργεί τις γραφικές παραστάσεις αυτών, σε συνδυασμό με την αντιστοίχιση των σημείων των παραστάσεων. Η αντιστοίχιση των σημείων είναι ένας τρόπος για να αναπαρασταθεί η ομοιότητα των δύο γραφικών παραστάσεων. Το δεύτερο πρόγραμμα δέχεται ως είσοδο δύο πίνακες και υπολογίζει το βέλτιστο μονοπάτι και το ελάχιστο κόστος το οποίο όσο πιο κοντά στο μηδέν είναι τόσο αυξάνεται η ομοιότητα των δύο ακολουθιών που βρίσκονται στους πίνακες. Οι πίνακες των ακολουθιών προκύπτουν από τα δεδομένα τα οποία καταγράφηκαν μέσω του Wireshark. Το Wireshark , όμως, δεν δημιουργεί από μόνο του πίνακες για τα δεδομένα που κατέγραψε αλλά εξάγει τα δεδομένα από μία καταγραφή σε αρχείο. Επομένως προκειμένου τα προγράμματα, που αφορούν τον αλγόριθμο Dynamic Time Warping , να δεχθούν έγκυρη είσοδο κρίθηκε αναγκαίο να δημιουργηθεί ένα ακόμα πρόγραμμα το οποίο διαβάζει τα δεδομένα από ένα από τα αρχεία στα οποία εξάγει δεδομένα το Wireshark , ώστε να σχηματίσει έναν πίνακα από τα δεδομένα αυτά (towards data science – Dynamic Time Warping, n.d.) (Medium – Dynamic Time Warping with Time Series , 07/09/2018) (RIP Tutorial – Introduction To Dynamic Time Warping, n.d.).

Τα προγράμματα που σχετίζονται με την ανάλυση των κυβερνο-επιθέσεων εκτελέστηκαν στο περιβάλλον προσομοίωσης που δημιουργήθηκε για το θύμα των επιθέσεων. Είναι σημαντικό να σημειωθεί ότι η επιλογή αυτή δεν σχετίζεται με τον

ρόλο του χρήστη του περιβάλλοντος προσομοίωσης αλλά με το λειτουργικό σύστημα του περιβάλλοντος αυτού. Για όλα τα προγράμματα που δημιουργήθηκαν για το πείραμα χρησιμοποιήθηκε η προγραμματιστική γλώσσα Python. Ο κύριος λόγος που χρησιμοποιήθηκε η συγκεκριμένη προγραμματιστική γλώσσα είναι η ευκολία εκτέλεσης προγραμμάτων της γλώσσας αυτής στο λειτουργικό σύστημα που επιλέχθηκε για το περιβάλλον προσομοίωσης του θύματος (Linux).

Κεφάλαιο 4. Διεξαγωγή πειράματος

4.1. Εισαγωγή

Για το πείραμα έχουν επιλεχθεί τέσσερις κυβερνο-επιθέσεις. Για κάθε επίθεση τα βήματα που πραγματοποιήθηκαν είναι τα εξής:

- Προσομοίωση κυβερνο-επίθεσης
- Καταγραφή κυβερνο-επίθεσης κατά την διάρκεια προσομοίωσής της
- Καταγραφή φυσιολογικής κίνησης του δικτύου προκειμένου να συγκριθεί με την καταγραφή της επίθεσης
- Ανάλυση των δύο καταγραφών μέσω του αλγορίθμου Dynamic Time Warping

Η ανάλυση των καταγραφών έγινε με βάση τρεις διαφορετικές μετρικές οι οποίες είναι ο χρόνος που έφτασαν τα πακέτα, το μέγεθος των πακέτων και το χρονικό διάστημα των χρόνων μεταξύ δύο συνεχόμενων πακέτων. Η επιλογή τριών μετρικών και όχι μίας οδηγεί σε ένα πιο τεκμηριωμένο αποτέλεσμα σχετικά με την ικανότητα του συστήματος να διαχωρίσει μία επίθεση από μία φυσιολογική κίνηση του δικτύου.

Αξίζει να σημειωθεί ότι ο χρόνος που φτάνουν τα πακέτα στο Wireshark είναι σε nanoseconds και αρχίζει να μετράει από την στιγμή που φτάσει το πρώτο πακέτο.

Οι τέσσερις κυβερνο-επιθέσεις που επιλέχθηκαν είναι οι εξής:

- Brute force attack
- SQL injection attack
- Slowloris attack
- TCP SYN flood attack

Οι επιθέσεις ανήκουν σε δύο διαφορετικές κατηγορίες κυβερνο-επιθέσεων. Οι επιθέσεις brute force και SQL injection ανήκουν στις Web based attacks και χρησιμοποιούνται απέναντι σε web εφαρμογές ή ιστοσελίδες. Από την άλλη μεριά οι επιθέσεις Slowloris και TCP SYN flood έχουν στόχο να διαταράσσουν την ομαλή λειτουργία των servers και κατά συνέπεια των συστημάτων που χρησιμοποιούν τους servers αυτούς.

4.2. SQL injection Attack

4.2.1. Προσομοίωση επίθεσης

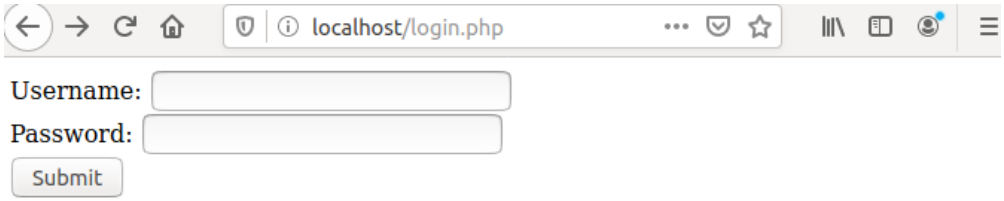
Η επίθεση SQL injection στοχεύει είτε ιστοσελίδες είτε web εφαρμογές και εισάγει κακόβουλο κώδικα από την προγραμματιστική γλώσσα SQL σε φόρμες εισαγωγής πληροφοριών. Προκειμένου λοιπόν να πραγματοποιηθεί η επίθεση αυτή χρειάστηκε

χρήση της login φόρμας που δημιουργήθηκε με την βοήθεια των τεχνολογιών HTML, SQL , PHP. Η φόρμα αυτή χρειάζεται να εκτελεστεί σε έναν υπολογιστή προκειμένου να χρησιμοποιηθεί από έναν χρήστη και με αποτέλεσμα να αναπαραστήσει επιτυχώς τις φόρμες πληροφοριών που παρέχουν οι ιστοσελίδες και οι web εφαρμογές.

Στην περίπτωση του πειράματος ο υπολογιστής που θα εκτελέσει την φόρμα και θα την παρέχει στους χρήστες, είναι ο εικονικός υπολογιστής ο οποίος αντιπροσωπεύει το θύμα. Προκειμένου να αποκτήσουμε πρόσβαση στην φόρμα και να γίνει δυνατή η εκτέλεση της επίθεσης, χρειάστηκε αρχικά να συνδεθούμε στον εικονικό υπολογιστή που αντιπροσωπεύει τον επιτιθέμενο. Για να συνδεθούμε στην φόρμα από οποιονδήποτε υπολογιστή θα πρέπει μέσω ενός web browser να εισάγουμε το URL που αντιστοιχεί στην φόρμα. Το URL που αντιστοιχεί στην φόρμα θα είναι http://IP-address/path-to-form/login.php. Το URL στον υπολογιστή που εκτελείται η φόρμα μπορεί επίσης να είναι http://localhost/path-to-form/login.php. Τα δύο αυτά URLs διαφέρουν από τα συνηθισμένα URLs τα οποία βλέπει ένας χρήστης όταν επισκέπτεται μία ιστοσελίδα. Τα URLs των ιστοσελίδων δεν αρχίζουν με μία IP διεύθυνση αλλά με το όνομα της ιστοσελίδας. Ωστόσο αυτό το όνομα αντιστοιχεί σε μία IP διεύθυνση. Ο λόγος που στο διαδίκτυο χρησιμοποιούνται ονόματα αντί για IP διευθύνσεις είναι επειδή είναι αρκετά πιο εύκολο για έναν χρήστη να θυμάται ένα όνομα παρά μία IP διεύθυνση. Επομένως το URL του πειράματος και τα URLs των ιστοσελίδων δεν έχουν καμία διαφορά όσον αφορά την λειτουργικότητα τους.

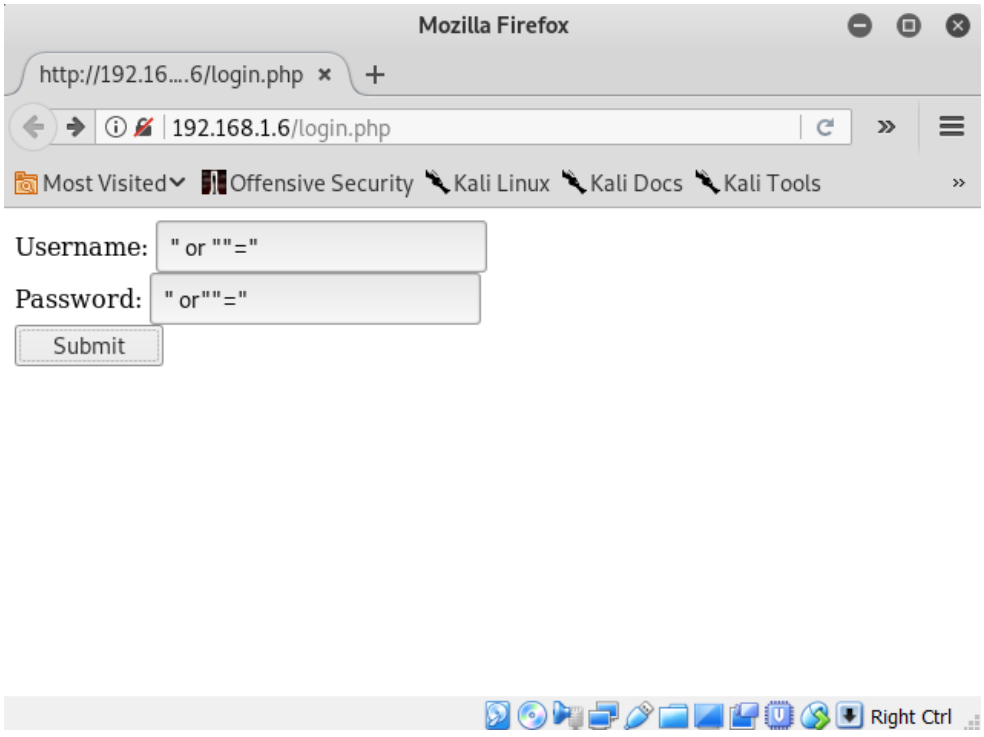
Συνεπώς για να συνδεθούμε στην φόρμα θα πρέπει να μάθουμε την IP διεύθυνση του εικονικού μηχανήματος του θύματος στο οποίο εκτελείται η φόρμα. Αυτό επιτυγχάνεται μέσω της γραμμής εντολής του λειτουργικού συστήματος Linux εκτελώντας την εντολή ifconfig. Με τον τρόπο αυτόν γίνεται διαθέσιμη στο εικονικό μηχάνημα του επιτιθέμενου η φόρμα και κατά συνέπεια μπορεί να πραγματοποιηθεί η επίθεση.

Η φόρμα έχει δύο πεδία εισαγωγής δεδομένων τα οποία μπορεί να χρησιμοποιήσει ο χρήστης. Αυτά είναι τα username και password, όπως φαίνεται και παρακάτω:



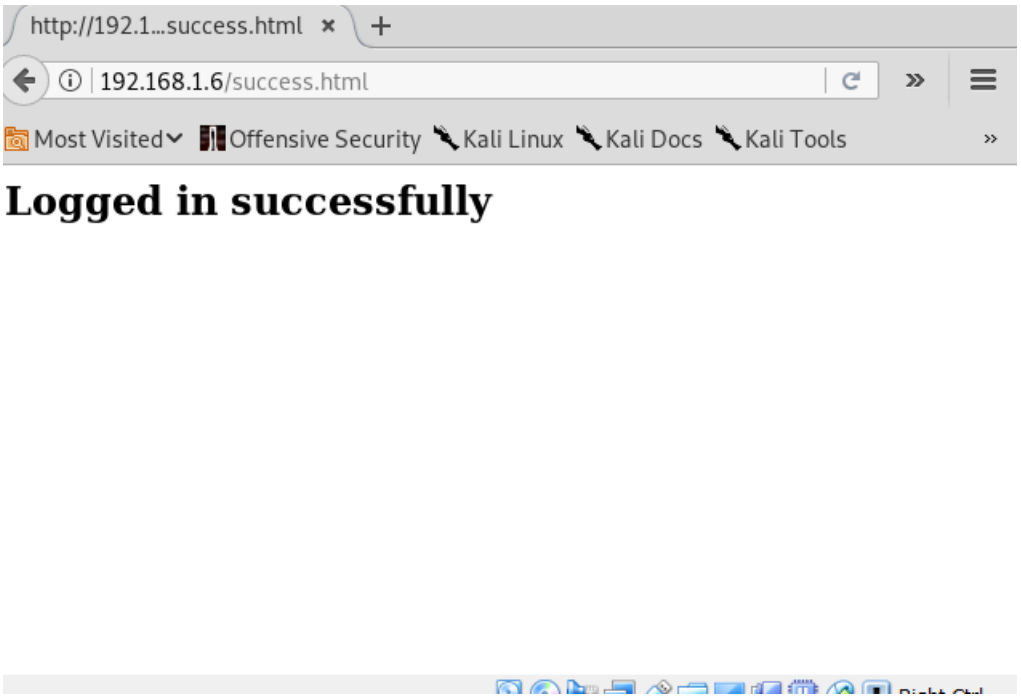
Εικόνα 22. PHP login form used for SQL injection attack

Επομένως ο επιτιθέμενος μπορεί να χρησιμοποιήσει κάποιο αυτά προκειμένου να επιτεθεί. Στο πείραμα χρησιμοποιήθηκαν και τα δύο πεδία. Όπως αναφέρθηκε, για να εκτελέσουμε μία SQL injection επίθεση πρέπει να εισάγουμε SQL κώδικα στα πεδία. Παρακάτω φαίνεται η επίθεση:



Εικόνα 23. SQL injection code in the PHP login form

Γίνεται αντιληπτό ότι ο κακόβουλος SQL κώδικας είναι ο : «” or “”=”». Επίσης στο URL φαίνεται μία IP διεύθυνση (192.168.1.6) μαζί με το αρχείο που εκτελεί τον κώδικα για την φόρμα (login.php). Η IP διεύθυνση αυτή είναι η IP διεύθυνση του εικονικού μηχανήματος του θύματος. Με το πάτημα του κουμπιού Submit καταφέρνουμε να εισέλθουμε επιτυχώς στην εφαρμογή.

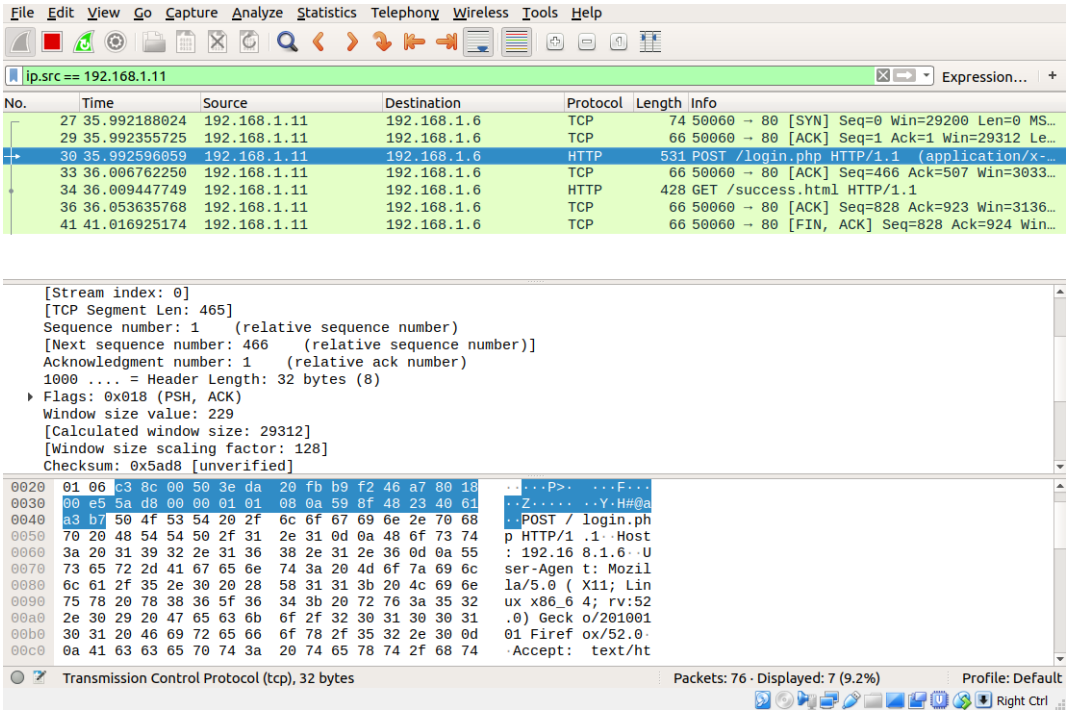


Εικόνα 24.Success login after SQL injection

Το αποτέλεσμα της συγκεκριμένης επίθεσης είναι ότι καταφέραμε να αποκτήσουμε πρόσβαση σε μία εφαρμογή χωρίς να είμαστε έγκυροι χρήστες. Αυτό σημαίνει μη εξουσιοδοτημένη χρήση της εφαρμογής και πιθανή απόκτηση πληροφοριών σχετικά με την εφαρμογή. Με τροποποίηση του κακόβουλου SQL κώδικα είναι δυνατή η απόκτηση δεδομένων για όλους τους χρήστες, αλλά και η καταστροφή της βάσης δεδομένων της εφαρμογής.

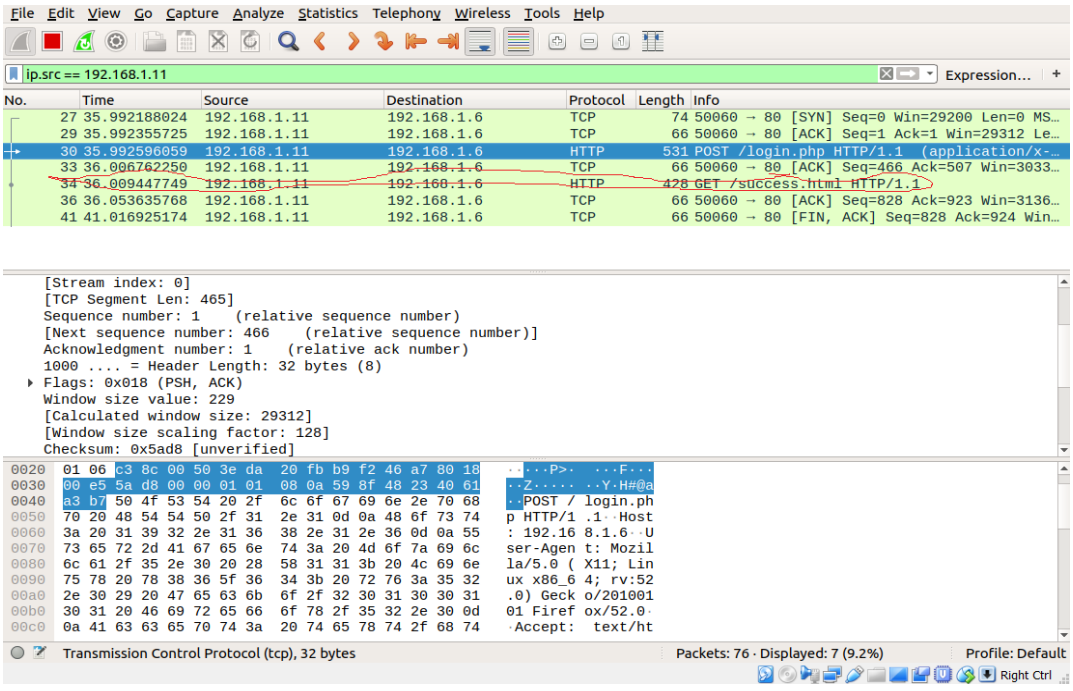
4.2.2. Καταγραφή και ανάλυση της επίθεσης

Προκειμένου να αναλυθεί η επίθεση επιτυχώς, κρίθηκε απαραίτητο η καταγραφή της κίνησης που δημιουργείται όταν ο χρήστης εισάγει στην φόρμα έγκυρα στοιχεία. Παρακάτω φαίνεται η καταγραφή μίας τέτοιας κίνησης:



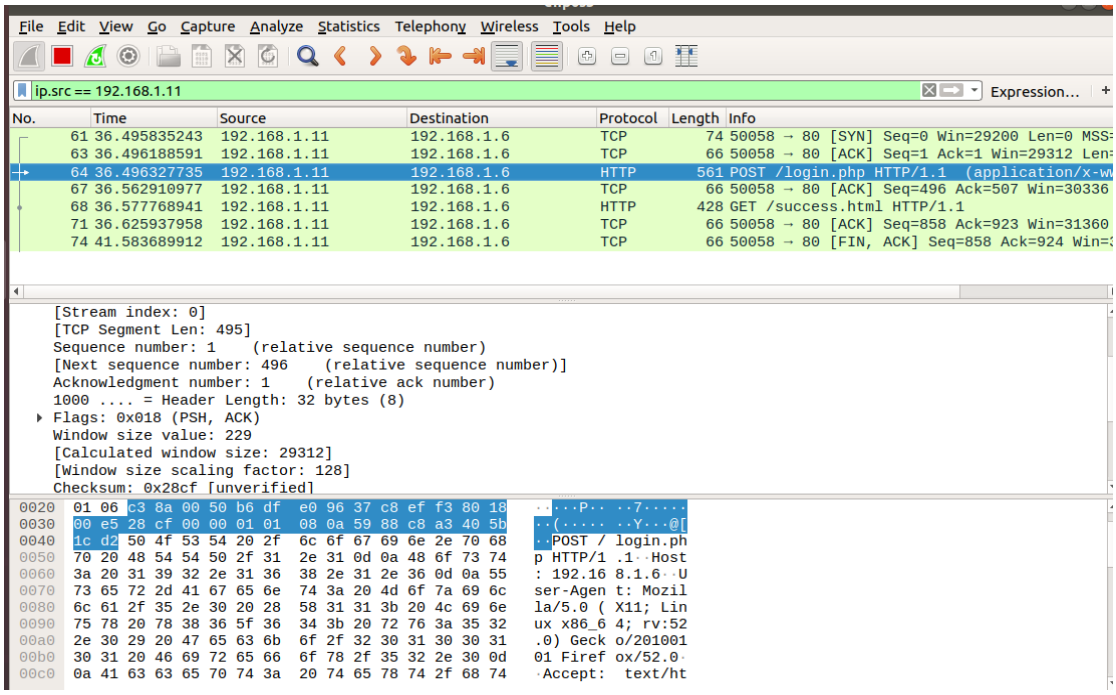
Εικόνα 25. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της SQL injection επίθεσης

Στην παραπάνω εικόνα , λοιπόν , φαίνεται η καταγραφή, μέσω του Wireshark , μίας κίνησης κατά την οποία ο χρήστης εισήγαγε έγκυρο username και password και συνδέθηκε επιτυχώς. Το γεγονός ότι ο χρήστης συνδέθηκε επιτυχώς φαίνεται από την γραμμή της καταγραφής στην οποία η στήλη “Info” έχει την πληροφορία “GET /success.html”.

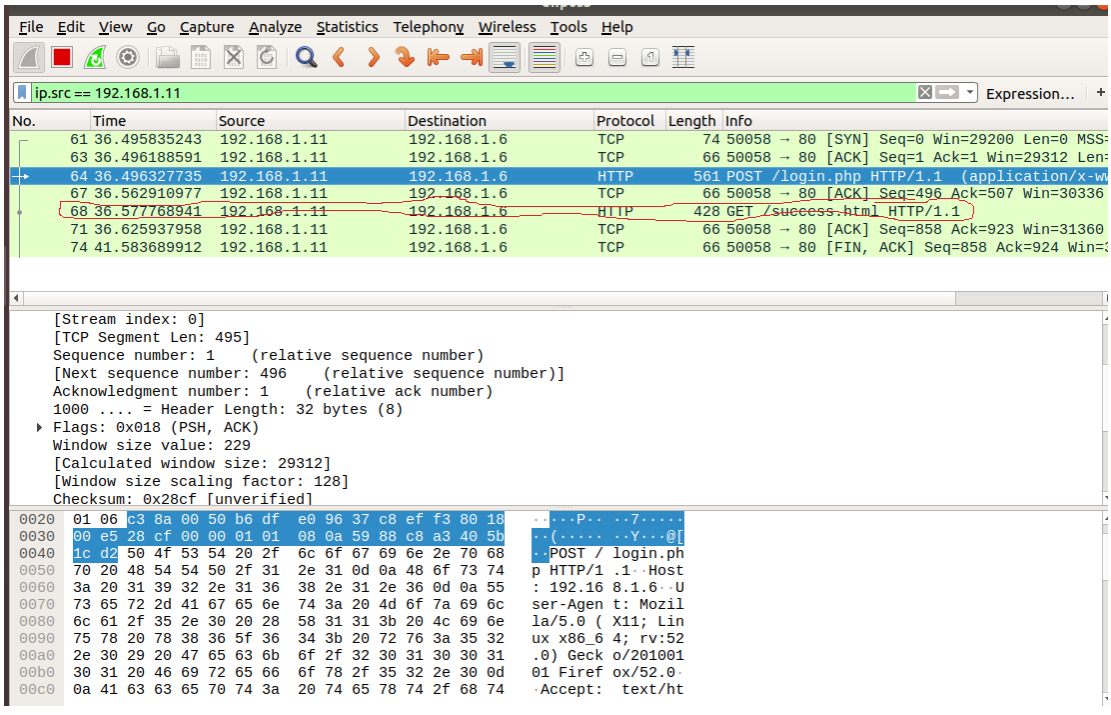


Εικόνα 26. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της SQL injection επίθεσης με σημειωμένο το πακέτο που δείχνει το επιτυχημένο login

Μέσω της καταγραφής της κίνησης έγκυρων δεδομένων και της κίνησης της επίθεσης μπορούμε να βγάλουμε μία τεκμηριωμένη απάντηση σχετικά με το εάν είναι δυνατόν να ξεχωρίσουμε μία επίθεση από μία απλή κίνηση με την βοήθεια της καταγραφής. Παρακάτω φαίνεται η καταγραφή της επίθεσης:



Εικόνα 27. Καταγραφή Wireshark για την επίθεση SQL injection



Εικόνα 28. Καταγραφή Wireshark για την επίθεση SQL injection με σημειωμένο το πακέτο που δείχνει το επιτυχημένο login

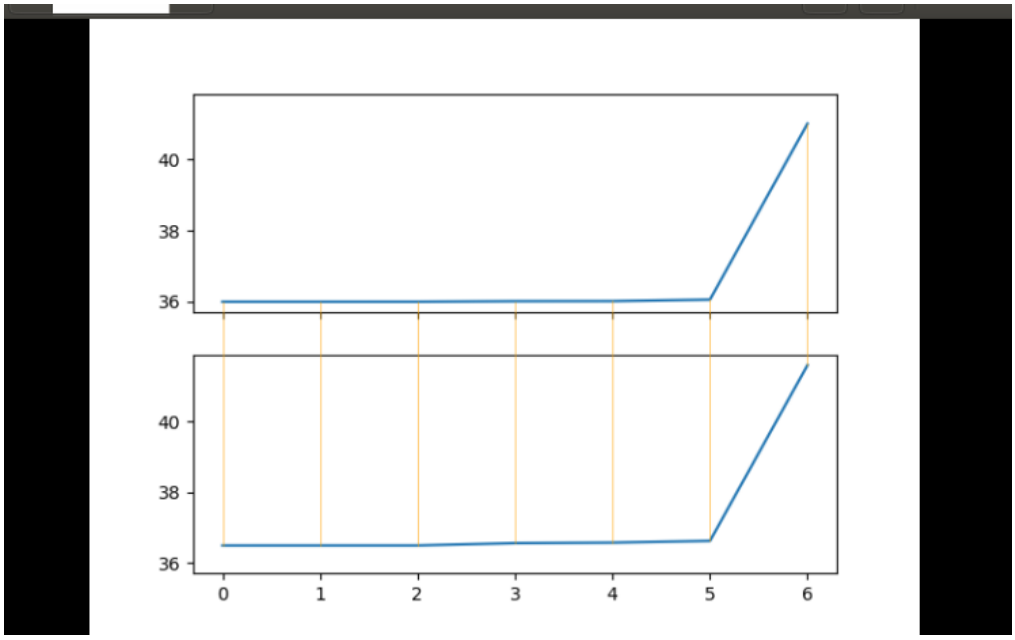
Παρατηρείται ότι οι δύο καταγραφές είναι αρκετά όμοιες καθώς έχουν τον ίδιο αριθμό πακέτων, παρόμοια είδη πακέτων και τέλος περιέχουν το πακέτο επιτυχής σύνδεσης (success.html). Προκειμένου όμως να προκύψει ένα έγκυρο αποτέλεσμα σχετικά με την επίθεση, είναι αναγκαία η περαιτέρω σύγκριση των δύο καταγραφών. Για τον λόγο αυτό χρησιμοποιήθηκε η μέθοδος Dynamic Time Warping (DTW) για τρεις διαφορετικές παραμέτρους που είτε αποτελούν στήλες είτε προκύπτουν από στήλες στην καταγραφή του Wireshark. Οι παράμετροι αυτοί είναι:

- Time
- Interval
- Size

4.2.3. Ανάλυση των καταγραφών με βάση την παράμετρο Time

Η στήλη Time στην καταγραφή του Wireshark αντιπροσωπεύει την χρονική διαφορά μεταξύ του πακέτου που εξετάζεται και του πρώτου frame, όπου frame ένα σύνολο δεδομένων του επιπέδου Data Link Layer στο OSI μοντέλο. Η χρονική διαφορά υπολογίζεται σε nanoseconds.

Προκειμένου λοιπόν να πραγματοποιηθεί η ανάλυση των καταγραφών του Wireshark για την επίθεση και την απλή κίνηση, τα δεδομένα από την στήλη των δύο καταγραφών μετατράπηκαν σε πίνακες. Στην συνέχεια μέσω ενός προγράμματος το οποίο χρησιμοποιεί τον αλγόριθμο Dynamic Time Warping (DTW) προέκυψε η ομοιότητα των δύο καταγραφών. Παρακάτω φαίνεται η ομοιότητα:



Εικόνα 29. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Time

Η πρώτη γραφική παράσταση αφορά την καταγραφή της απλής κίνησης όπου ο χρήστης πληκτρολόγησε έγκυρα δεδομένα στην φόρμα. Κατά συνέπεια η δεύτερη γραφική παράσταση αφορά την καταγραφή της επίθεσης. Με μία πρώτη ματιά φαίνεται ότι οι δύο γραφικές παραστάσεις είναι αρκετά όμοιες. Υπάρχει όμως ένας ακόμη τρόπος ανάλυσης των καταγραφών μέσω του οποίου προκύπτει το βέλτιστο μονοπάτι και το ελάχιστο κόστος ομοιότητας των δύο καταγραφών με βάση πάντα τον αλγόριθμο Dynamic Time Warping (DTW).

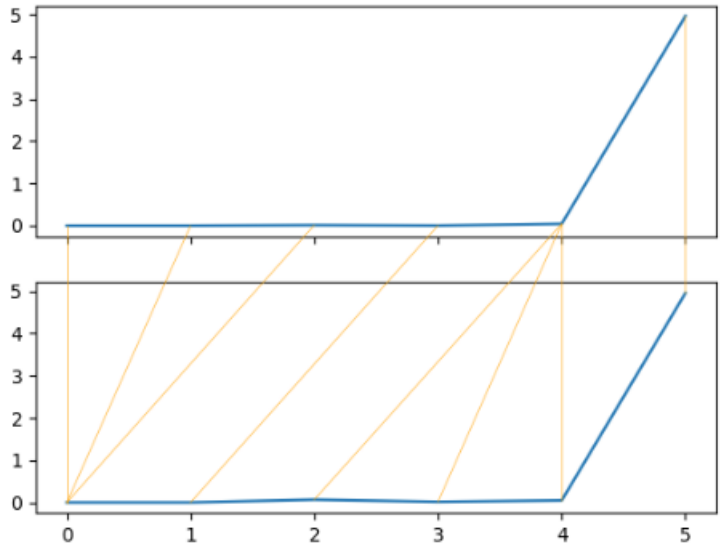
Εικόνα 30. Βέλτιστο μονοπάτι και ελάχιστο κόστος των καταγραφών κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Time

Στα αριστερά της εικόνας φαίνεται το βέλτιστο μονοπάτι και στα δεξιά το ελάχιστο κόστος που είναι 3.7747 περίπου. Αξίζει να σημειωθεί ότι ο πίνακας στον οποίο απεικονίζεται το βέλτιστο μονοπάτι έχει ως άξονα x την κίνηση του χρήστη όταν εισήγαγε έγκυρα δεδομένα και ως άξονα y την κίνηση της επίθεσης. Επίσης όταν το βέλτιστο μονοπάτι περιέχει διαγώνιες κινήσεις, τότε σημαίνει ότι οι γραφικές παραστάσεις των δύο κινήσεων ταυτίζονται στο κομμάτι του πίνακα στο οποίο έχουμε την διαγώνια κίνηση. Στην περίπτωση μας βλέπουμε ότι ολόκληρο το βέλτιστο μονοπάτι αποτελείται από διαγώνιες κινήσεις καθώς και ότι το ελάχιστο κόστος δεν απέχει πολύ από το μηδέν. Επομένως μπορούμε να θεωρήσουμε τις δύο κινήσεις όμοιες ως προς τον χρόνο των πακέτων.

Ωστόσο ο χρόνος των πακέτων μπορεί να μην αποτελεί μία έγκυρη μετρική. Επομένως είναι αναγκαίο να εξεταστούν και άλλοι παράμετροι προκειμένου να κρίνουμε εάν οι δύο κινήσεις μπορούν να θεωρηθούν όμοιες.

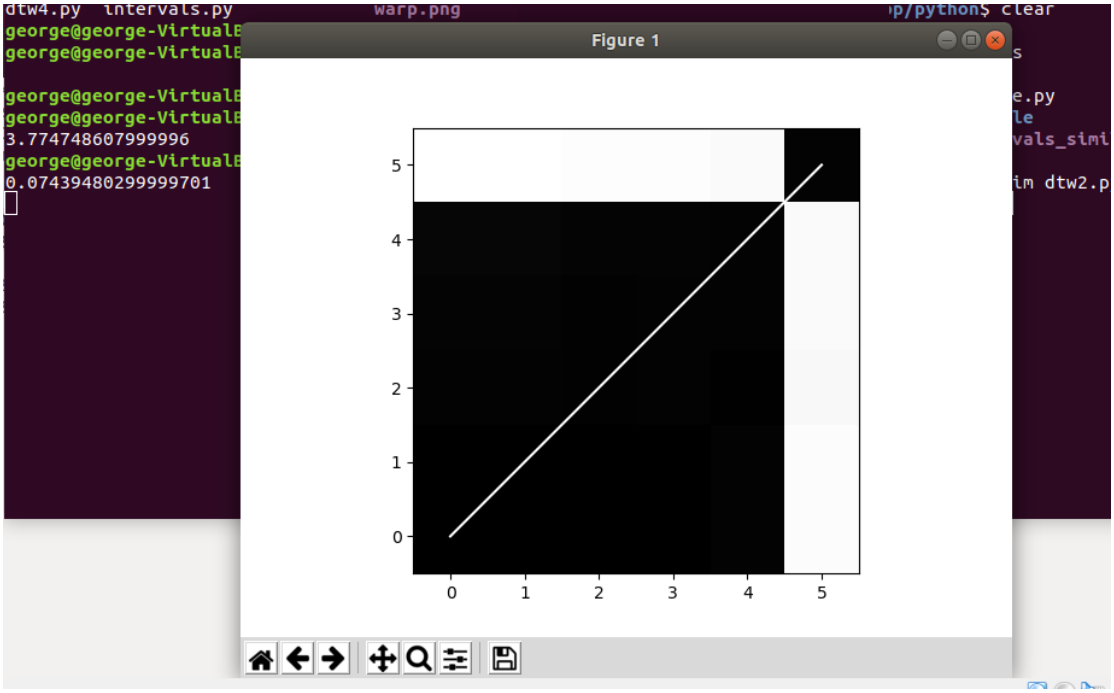
4.2.4. Ανάλυση των καταγραφών με βάση την παράμετρο Interval

Η επόμενη παράμετρος με βάση την οποία θα γίνει η ανάλυση της επίθεσης SQL injection είναι τα intervals, δηλαδή η χρονική διαφορά μεταξύ δύο συνεχόμενων πακέτων. Τα intervals δεν αποτελούν κάποια παράμετρο του Wireshark. Επομένως κατασκευάσθηκε ένα πρόγραμμα το οποίο δέχεται ως είσοδο τις τιμές της παραμέτρου Time του Wireshark και εμφανίζει ως έξοδο τα intervals. Όπως συνέβη για την παράμετρο Time , οι τιμές των intervals μετατράπηκαν σε πίνακες μέσω ενός προγράμματος. Οι πίνακες λειτούργησαν ως είσοδοι για το πρόγραμμα που υλοποιεί τον αλγόριθμο Dynamic Time Warping (DTW) με αποτέλεσμα να προκύψει η παρακάτω ομοιότητα:



Εικόνα 31. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Interval

Γίνεται αντιληπτό ότι η αντιστοίχιση των δύο γραφικών παραστάσεων είναι διαφορετική από αυτή των γραφικών παραστάσεων της παραμέτρου Time καθώς στην περίπτωση αυτή ένα σημείο της δεύτερης γραφικής παράσταση μπορεί να αντιστοιχεί σε παραπάνω από ένα σημεία της πρώτης γραφικής παράσταση. Προκειμένου όμως να είμαστε σε θέση να συγκρίνουμε πλήρως τις ομοιότητες των γραφικών παραστάσεων των παραμέτρων Time και Interval πρέπει να υπολογίσουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος των γραφικών παραστάσεων της παραμέτρου Interval:



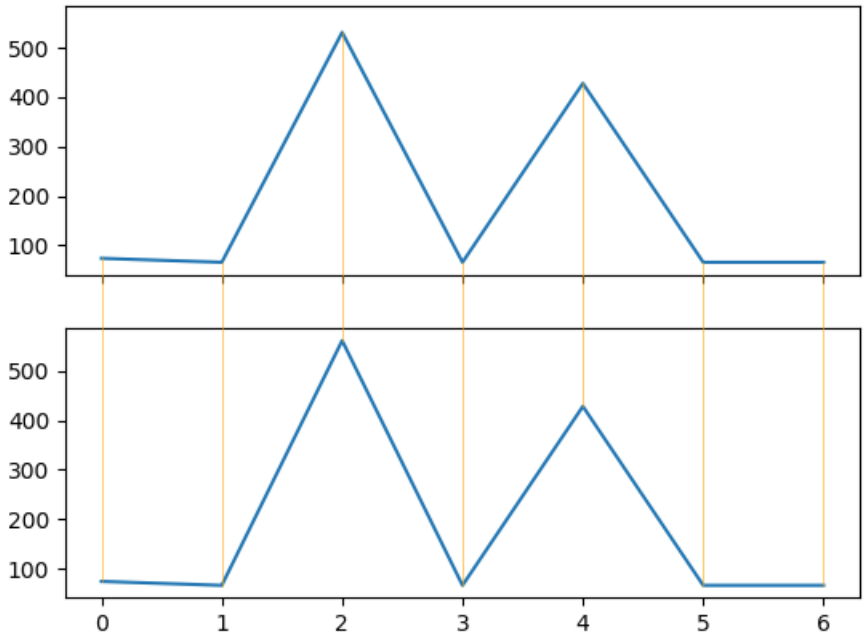
Εικόνα 32. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Interval

Βλέπουμε ότι το βέλτιστο μονοπάτι αποτελείται μόνο από διαγώνιες κινήσεις. Αυτό σημαίνει ότι οι δύο γραφικές παραστάσεις είναι αρκετά όμοιες. Στο συμπέρασμα αυτό καταλήγουμε επίσης εάν παρατηρήσουμε την τιμή του ελάχιστου κόστους που είναι ίση περίπου με 0.074, τιμή που βρίσκεται αρκετά κοντά στο μηδέν.

Με βάση λοιπόν τις δύο παραπάνω εικόνες γίνεται κατανοητό ότι η ομοιότητα των δύο καταγραφών με βάση την παράμετρο Interval είναι αρκετά μεγάλη.

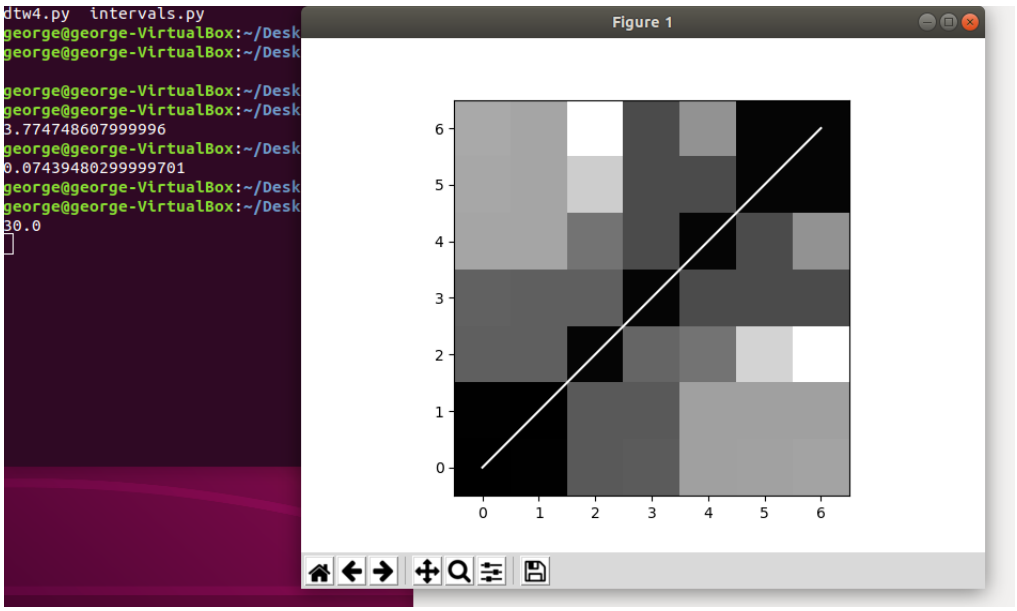
4.2.5. Ανάλυση των καταγραφών με βάση την παράμετρο Size

Η τελευταία παράμετρος για την οποία θα εξετάσουμε την ομοιότητα των δύο καταγραφών είναι η παράμετρος Size η οποία αποτελεί στήλη της καταγραφής του Wireshark. Με βάση λοιπόν των προγραμμάτων προκύπτει η ομοιότητα των καταγραφών:



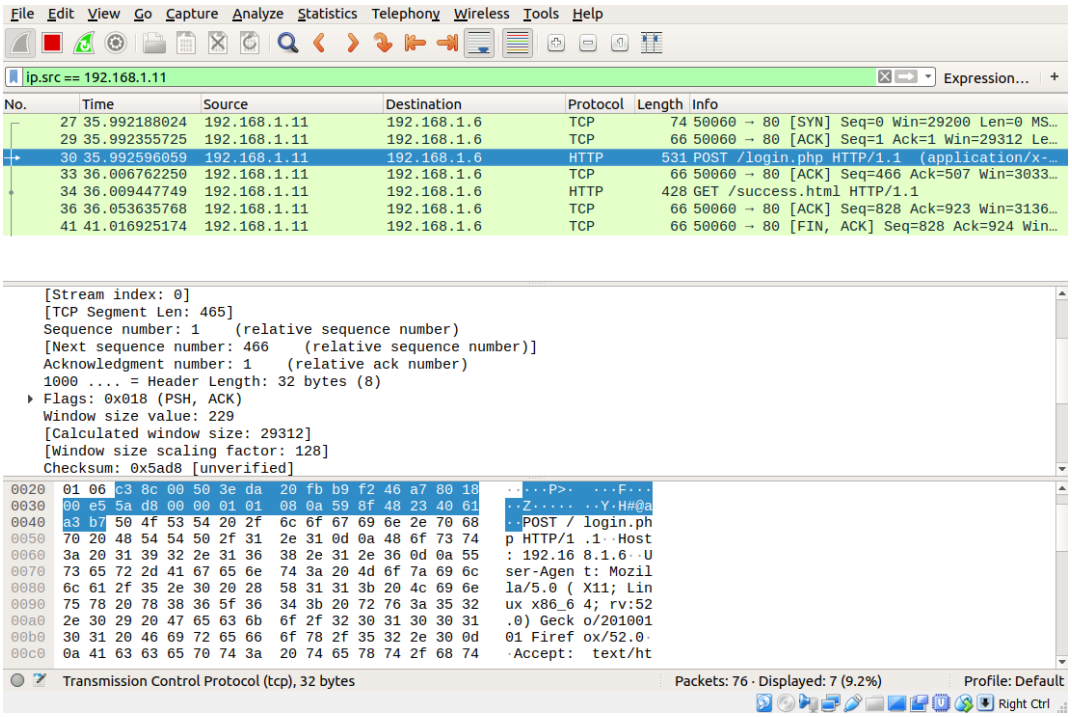
Εικόνα 33. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Size

Βλέπουμε ότι η μόνη διαφορά των γραφικών παραστάσεων είναι ότι το πρώτο peak της δεύτερης γραφικής παράστασης, η οποία αντιπροσωπεύει την επίθεση SQL injection, είναι μεγαλύτερο από το αντίστοιχο peak της πρώτης γραφικής παράστασης. Πέρα από την ομοιότητα εξετάζουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος:

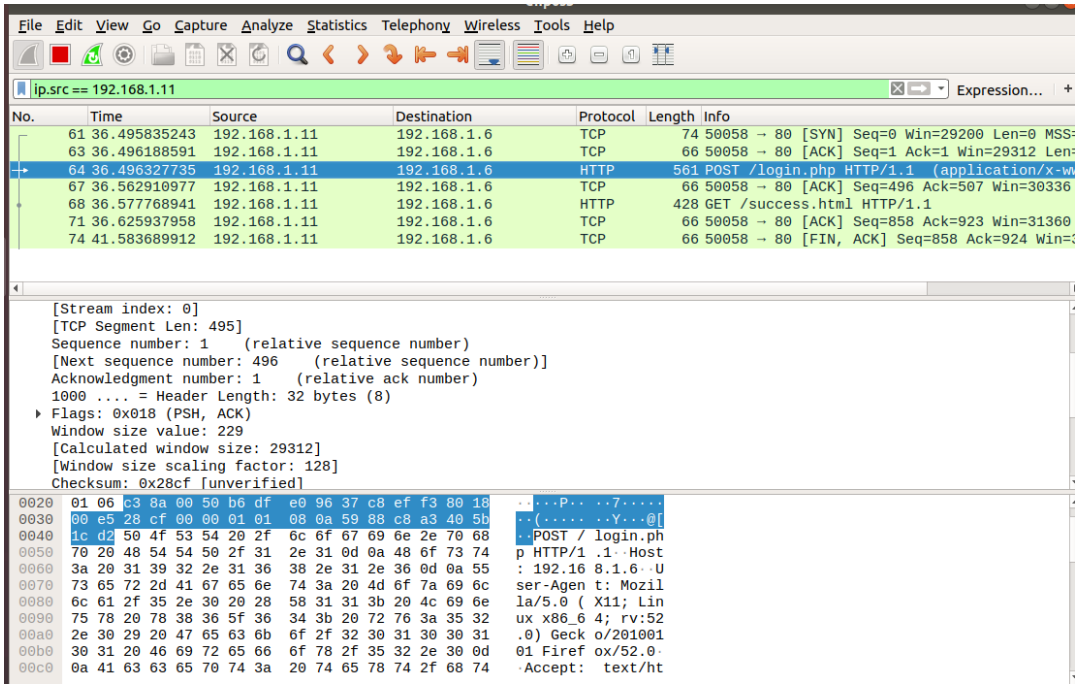


Εικόνα 34. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης SQL injection με βάση την παράμετρο Size

Βλέπουμε ότι το βέλτιστο μονοπάτι αποτελείται από διαγώνιες κινήσεις. Ωστόσο το ελάχιστο κόστος είναι ίσο με 30 το οποίο απέχει αρκετά από το 0. Επομένως η ομοιότητα των δύο καταγραφών με βάση την παράμετρο Size δεν είναι μεγάλη. Η διαφορά αυτή στην ομοιότητα προκύπτει από το πακέτο το οποίο περιέχει τα δεδομένα που εισήγαγε ο χρήστης προκειμένου να συνδεθεί επιτυχώς στην φόρμα:



Εικόνα 35. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης SQL injection με σημειωμένο το πακέτο που περιέχει τα δεδομένα που έστειλε ο χρήστης στην login φόρμα



Εικόνα 36. Καταγραφή Wireshark για την επίθεση SQL injection με σημειωμένο το πακέτο που περιέχει τα δεδομένα που έστειλε ο χρήστης στην login φόρμα

Στις δύο παραπάνω εικόνες είναι υπογραμμισμένα τα πακέτα που περιέχουν τα δεδομένα που εισήγαγε ο χρήστης στην φόρμα και με βάση τα οποία συνδέθηκε επιτυχώς. Η πρώτη εικόνα αφορά την κίνηση που προκλήθηκε από τα έγκυρα δεδομένα του χρήστη, ενώ η δεύτερη εικόνα αφορά την κίνηση που προκλήθηκε από την επίθεση SQL injection. Αν παρατηρήσουμε στην πρώτη εικόνα το υπογραμμισμένο πακέτο έχει μέγεθος (Length) ίσο με 531 ενώ στην δεύτερη εικόνα έχει μέγεθος ίσο με 561. Τα υπόλοιπα πακέτα έχουν ίδιο μέγεθος για τις δύο καταγραφές. Επομένως το υπογραμμισμένο πακέτο είναι αυτό που μειώνει την ομοιότητα των δύο καταγραφών. Το συμπέρασμα αυτό δικαιολογείται και από το γεγονός ότι τα δύο πακέτα περιέχουν διαφορετικό είδος δεδομένων τα οποία όμως οδηγούν σε επιτυχής σύνδεση και στις δύο περιπτώσεις. Στην πρώτη καταγραφή το πακέτο περιέχει την τιμή test για το πεδίο username και την τιμή 123 για το πεδίο password της φόρμας ενώ στην δεύτερη καταγραφή το πακέτο περιέχει τόσο για το username όσο και για το password τον SQL κώδικα: «” or “”=”» . Ωστόσο εάν τα έγκυρα δεδομένα είχαν μεγαλύτερο μέγεθος, δηλαδή εάν για παράδειγμα το πεδίο username της φόρμας είχε σαν έγκυρη τιμή μία μεγαλύτερη σειρά χαρακτήρων, τότε είναι πιθανό το μέγεθος των πακέτων να μην διέφερε. Για τον λόγο αυτό η μείωση της ομοιότητας που παρατηρήθηκε στην περίπτωσή μας μπορεί να μην παρατηρηθεί σε άλλες περιπτώσεις.

4.2.6. Συμπέρασμα των αναλύσεων

Συνεπώς, η ομοιότητα των καταγραφών της κανονικής κίνησης ενός χρήστη με έγκυρα δεδομένα και της κίνησης που προκλήθηκε από την επίθεση SQL injection είναι αρκετά όμοια όσων αφορά τις παραμέτρους Time και Interval σε αντίθεση με την ομοιότητα των καταγραφών με βάση την παράμετρο Size, γεγονός όμως που δεν συμβαίνει σε όλες τις περιπτώσεις. Επομένως είναι αρκετά δύσκολο ένα σύστημα να διαχωρίσει μεταξύ μίας απλής κίνησης και μίας επίθεσης SQL injection μέσω των καταγραφών των κινήσεων.

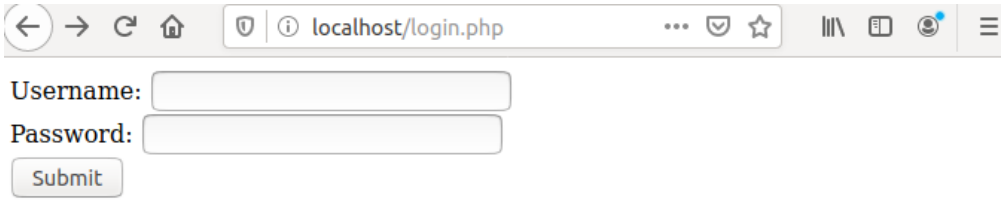
4.3. Bruteforce attack

4.3.1. Προσομοίωση επίθεσης

Η επίθεση bruteforce στοχεύει φόρμες στις οποίες ο χρήστης εισάγει δεδομένα προκειμένου να συνδεθεί επιτυχώς σε μία web εφαρμογή. Κατά την επίθεση αυτή ο κακόβουλος χρήστης προσπαθεί να βρει το κατάλληλο συνδυασμό των πεδίων της φόρμας προκειμένου να συνδεθεί στην εφαρμογή ως έγκυρος χρήστης. Δηλαδή ο χρήστης επιδιώκει να βρει τα διαπιστευτήρια (credentials) ενός άλλου χρήστη. Για να το καταφέρει αυτό ο επιτιθέμενος συνήθως χρησιμοποιεί κάποιο πρόγραμμα το

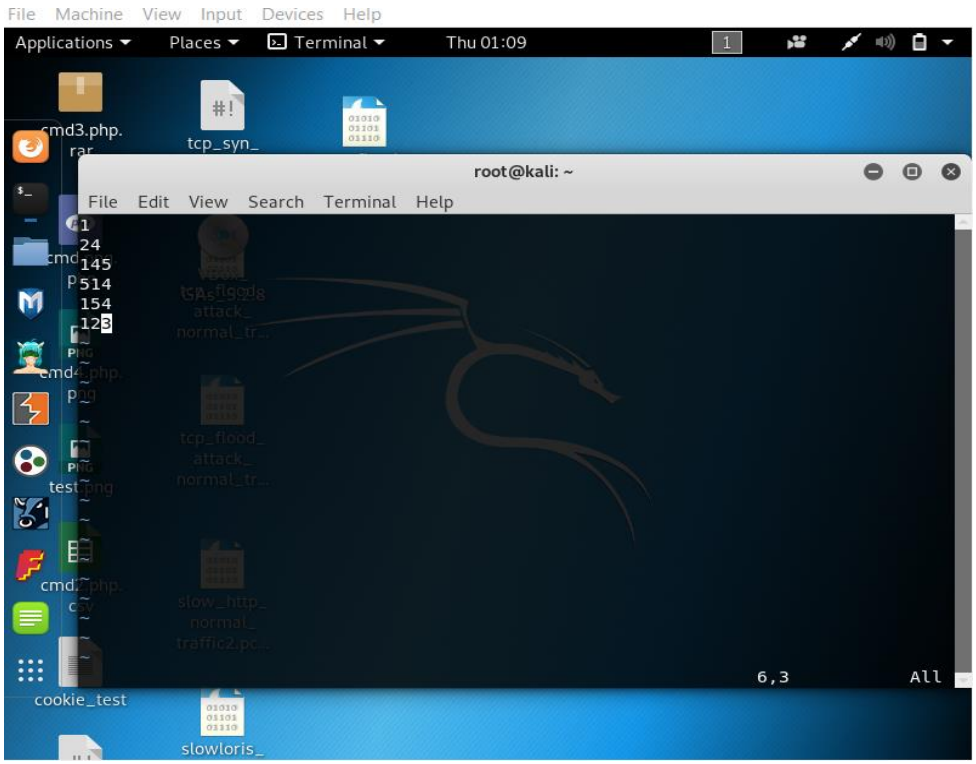
οποίο «διαβάζει» από μία λίστα με τυχαίους συνδυασμούς πεδίων, τους οποίους στέλνει στην φόρμα προκειμένου συνδεθεί επιτυχώς. Η αποστολή των συνδυασμών στην φόρμα γίνεται μέσω requests. Οι web εφαρμογές προκειμένου να επικοινωνούν με τον χρήστη χρησιμοποιούν requests, δηλαδή πακέτα που περιέχουν την πληροφορία που ζητάει ή στέλνει ο χρήστης. Δύο συνηθισμένα requests είναι τα GET και POST.

Επομένως προκειμένου να προσομοιωθεί η επίθεση χρησιμοποιήθηκε η φόρμα:



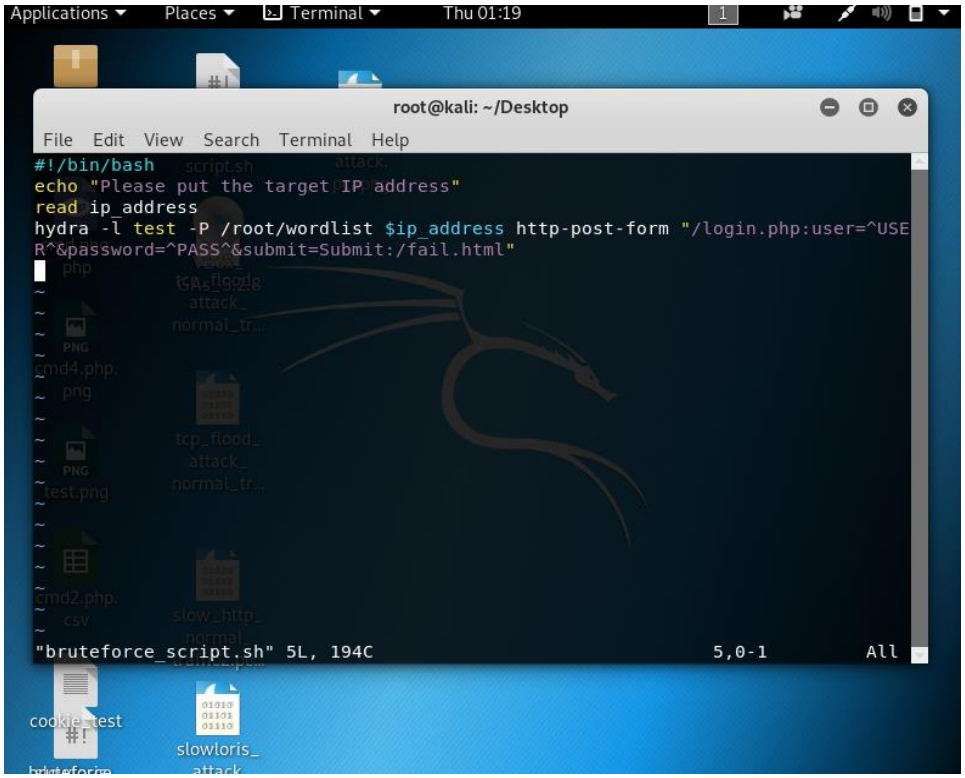
Εικόνα 37. PHP login form used for brute force attack

Επιπλέον δημιουργήθηκε μία λίστα από τυχαία passwords μέσα στην οποία προστέθηκε επιπλέον το έγκυρο password:



Εικόνα 38. Password list for brute force attack

Στην συνέχεια δημιουργήθηκε το πρόγραμμα το οποίο εκτελέστηκε προκειμένου να προσομοιωθεί η επίθεση. Το πρόγραμμα αυτό χρησιμοποιεί ως username το έγκυρο username (test) αλλά το password προσπαθεί να το βρει από την λίστα των passwords που κατασκευάσθηκε. Παρακάτω φαίνεται το πρόγραμμα:



Εικόνα 39. Script για την εκτέλεση της επίθεσης brute force

Αρχικά το πρόγραμμα αυτό ρωτάει τον χρήστη την IP διεύθυνση που θέλει να επιτεθεί και στην συνέχεια μέσω ενός εργαλείου που λέγεται hydra εκτελείται η επίθεση. Το εργαλείο hydra προκειμένου να εκτελεστεί χρειάζεται κάποιες ρυθμίσεις, μερικές από τις οποίες είναι αναγκαίες για να δουλέψει το εργαλείο, ενώ παρέχεται η επιπρόσθετη παραμετροποίηση μέσω επιπλέον προαιρετικών επιλογών. Στην περίπτωση μας προσδιορίσθηκαν οι αναγκαίες ρυθμίσεις, ως εξής:

- Η τιμή του username μετά την ρύθμιση “-l”.
  Στην περίπτωσή μας έχει δοθεί η έγκυρη τιμή του username για λόγους ευκολίας αλλά είναι δυνατό να υπάρχει στην ρύθμιση αυτή μία τυχαία λίστα από usernames. Στην περίπτωση που γίνεται χρήση λίστας από usernames τότε το “-l” μετατρέπεται σε “-L”.
- Η τιμή του Password μετά την ρύθμιση “-P”.
  Στο πείραμα χρησιμοποιήθηκε λίστα από passwords. Εάν γίνει χρήση μίας μοναδικής τιμής password τότε το “-P” μετατρέπεται σε “-p”.
- Στην συνέχεια προσδιορίζεται η IP διεύθυνση στην οποία θα γίνει η επίθεση.
  Στην περίπτωσή μας η μεταβλητή , η οποία περιέχει την τιμή που πληκτρολόγησε ο χρήστης, αναπαριστά την IP διεύθυνση
- Τέλος προσδιορίζεται το είδος των requests που θα σταλθούν στο θύμα.
  Στην εντολή της εικόνας το “http-post-form” αναπαριστά POST requests σε μία φόρμα. Μετά το “http-post-form” δίνονται πληροφορίες για την φόρμα, όπως το όνομα του αρχείου στο οποίο εκτελείται η φόρμα (“login.php”), το όνομα του πεδίου της φόρμας για το username (“user”) , το όνομα του πεδίου της φόρμας για το password (“password”), το όνομα του κουμπιού που πρέπει να πατηθεί για να σταλθούν τα δεδομένα του χρήστη και το όνομα του αρχείου το οποίο βλέπει ο χρήστης σε περίπτωση λάθους συνδυασμού username-password. Όλα τα παραπάνω είναι αρκετά εύκολα να βρεθούν είτε παρατηρώντας την φόρμα είτε με την λειτουργία inspect element που προσφέρει κάθε browser.

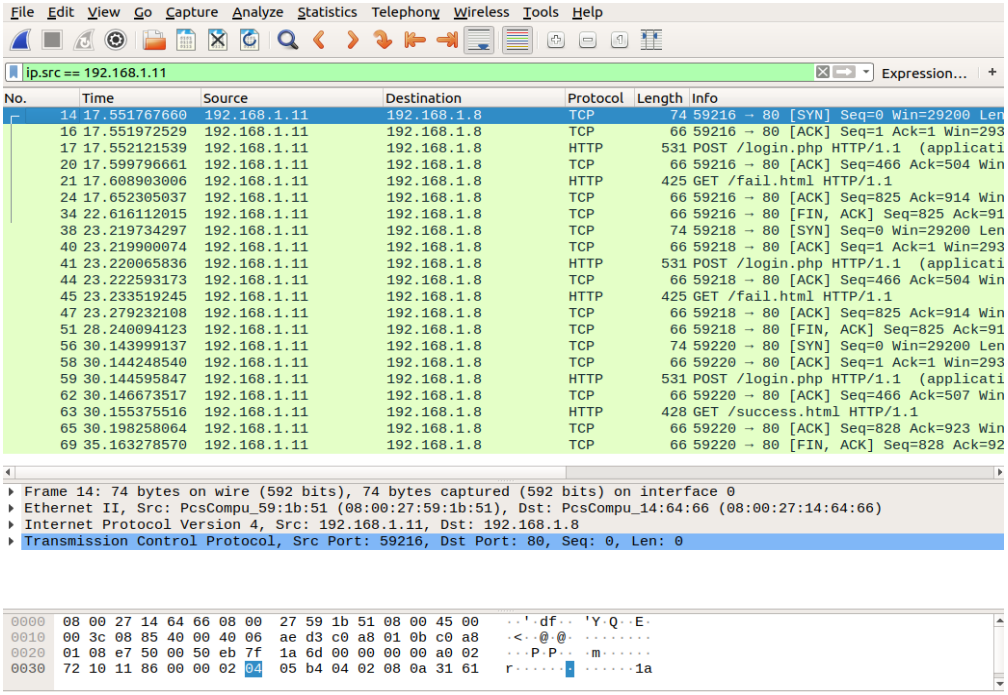
Από την εικόνα που περιέχει την λίστα των passwords την οποία θα χρησιμοποιήσει το πρόγραμμα για την επίθεση bruteforce, γίνεται αντιληπτό ότι υπάρχουν 6 διαφορετική συνδυασμοί username-password. Επομένως θα αποσταλούν στο θύμα κατά μέγιστο 6 POST requests προκειμένου να βρεθεί ο σωστός συνδυασμός για την επιτυχής σύνδεση στην φόρμα.

Πέρα από την επίθεση bruteforce προσομοιώθηκε η κίνηση ενός έγκυρου χρήστη που προσπαθεί να συνδεθεί στην φόρμα. Στην πραγματικότητα είναι αρκετά σύνηθες ένας χρήστης να αποτύχει να συνδεθεί επιτυχώς στην φόρμα στην πρώτη του προσπάθεια επειδή πληκτρολόγηση λανθασμένα τα στοιχεία του. Για την καλύτερη αναπαράσταση της πραγματικότητας λοιπόν προσομοιώθηκε η κίνηση ενός χρήστη που συνδέθηκε επιτυχώς στην τρίτη προσπάθειά του. Επομένως η καταγραφή της

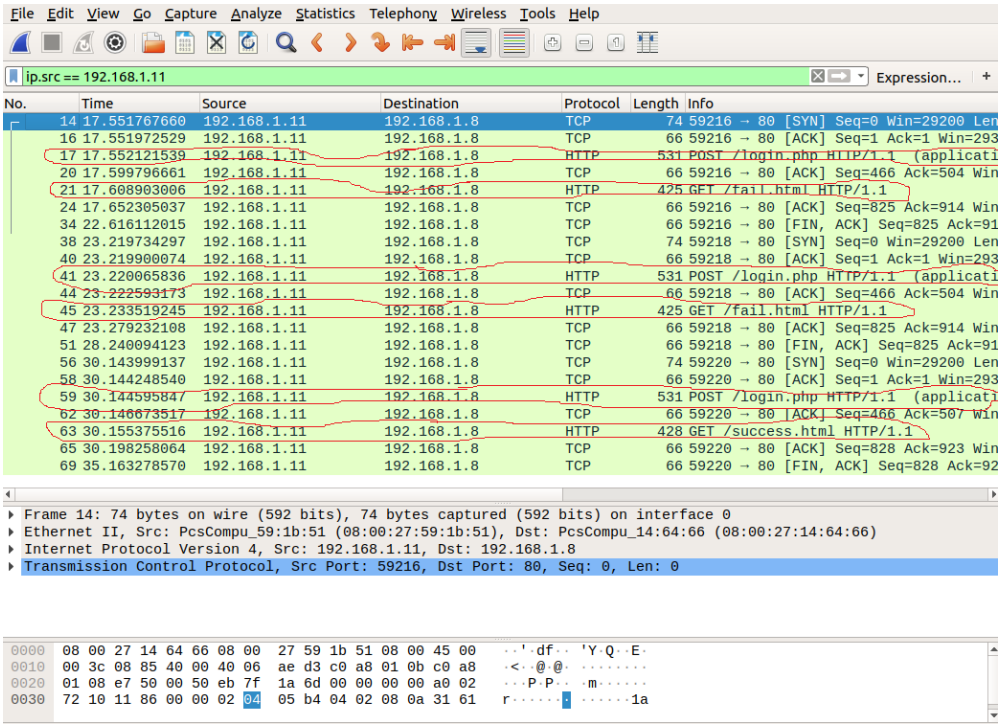
κίνησης ενός έγκυρου χρήστη θα περιέχει δύο αποτυχημένες προσπάθειες και μία επιτυχημένη προσπάθεια.

4.3.2. Καταγραφή και ανάλυση της επίθεσης

Παρακάτω φαίνεται η καταγραφή της κίνησης ενός χρήστη που προσπαθεί να συνδεθεί στην φόρμα με τα έγκυρα στοιχεία του:



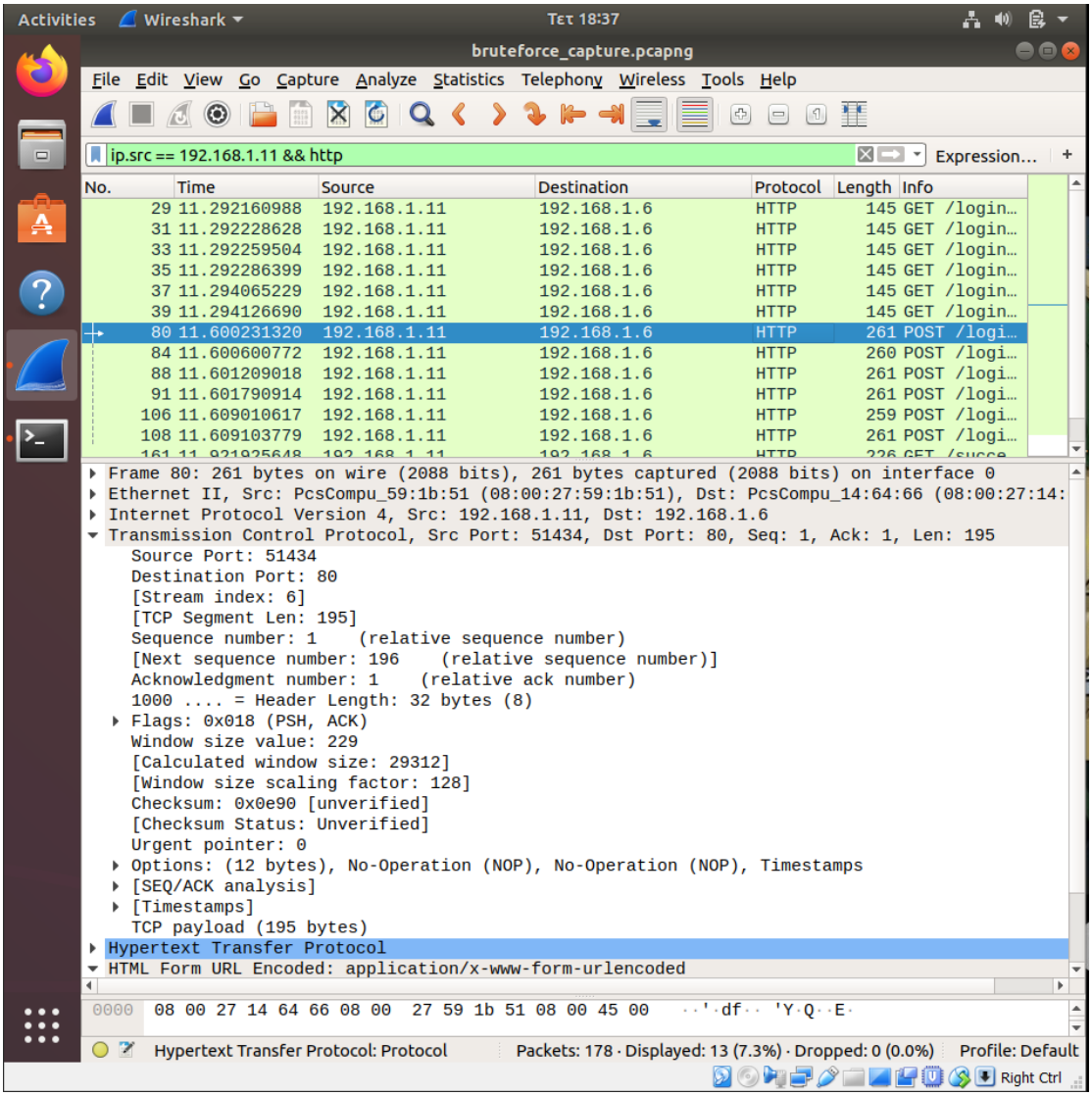
Εικόνα 40. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης brute force



Εικόνα 41. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης brute force με σημειωμένα τα HTTP πακέτα

Στην παραπάνω καταγραφή έχει χρησιμοποιηθεί φίλτρο προκειμένου να περιορίσει τα πακέτα σε αυτά που στέλνονται από την IP διεύθυνση του επιτιθέμενη. Στην δεύτερη εικόνα τα πακέτα που αφορούν τις προσπάθειες του χρήστη να συνδεθεί στην φόρμα είναι αυτά τα οποία είναι σημειωμένα με κόκκινο. Τα τέσσερα πρώτα πακέτα είναι οι δύο αποτυχημένες προσπάθειες και αυτό γίνεται αντιληπτό από το δεύτερο πακέτο της κάθε προσπάθειας, το οποίο είναι ένα GET request και στην στήλη Info του Wireshark αναφέρεται η σελίδα “fail.html”. Τα δύο τελευταία σημειωμένα πακέτα αφορούν την επιτυχημένη προσπάθεια του χρήστη να συνδεθεί, γεγονός που γίνεται αντιληπτό από την πληροφορία “success.html” στην στήλη Info του τελευταίου σημειωμένου πακέτου.

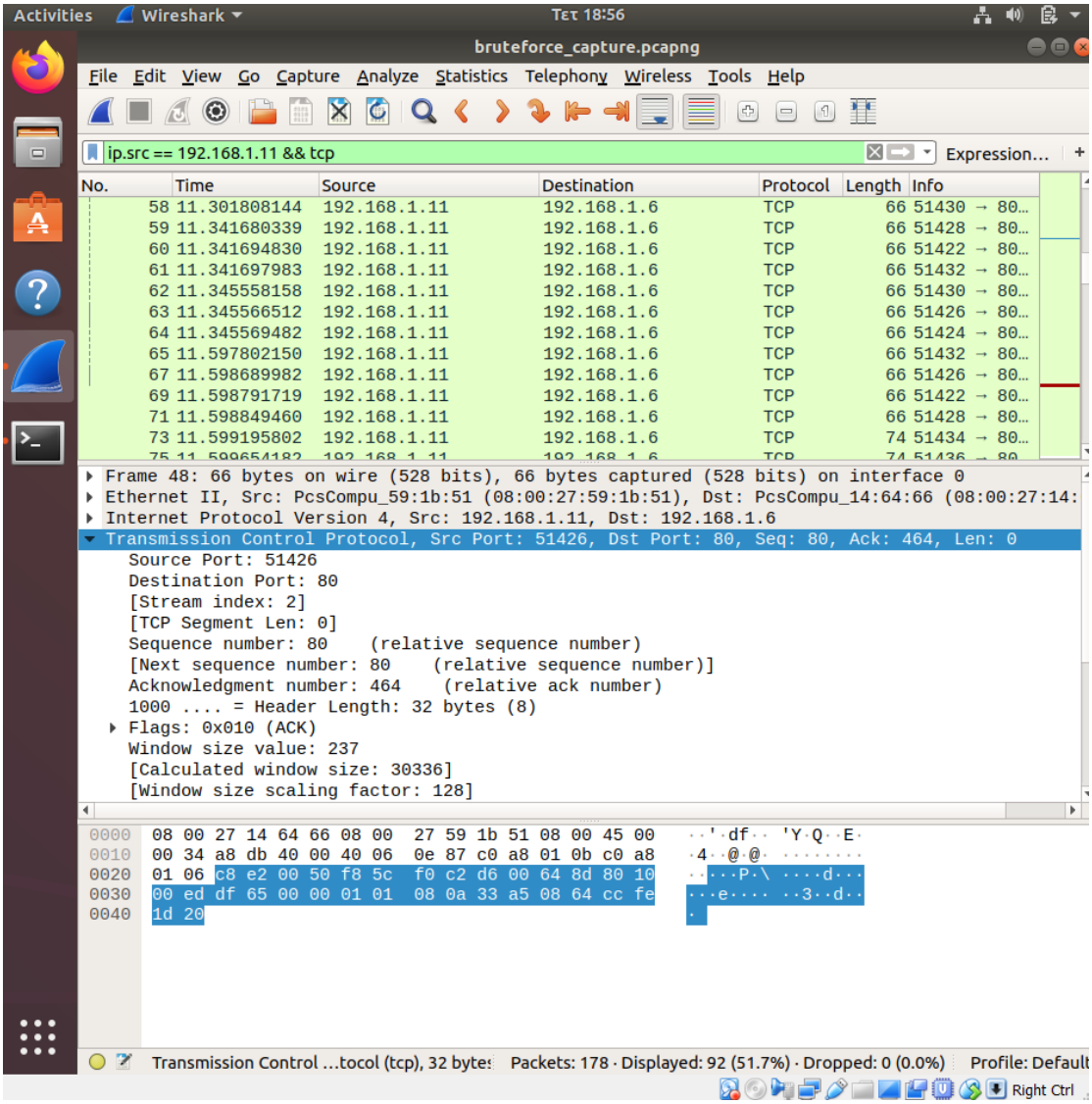
Στην συνέχεια παρουσιάζεται η καταγραφή της κίνησης που αφορά την επίθεση brute force:



Εικόνα 42. Καταγραφή Wireshark για HTTP πακέτα για την επίθεση brute force

Αρχικά παρατηρείται ότι η καταγραφή τις κίνησης που αφορά την επίθεση είναι μεγαλύτερη από αυτή της απλής κίνησης. Για τον λόγο αυτό πέρα από το φίλτρο για την IP διεύθυνση χρησιμοποιήθηκε φίλτρο για να παρουσιαστούν μόνο τα HTTP πακέτα καθώς δεν είναι δυνατή η παρουσίαση ολόκληρης της καταγραφής μέσω μίας εικόνα. Επομένως επιλέχθηκε στην εικόνα να παρουσιασθεί το κομμάτι της καταγραφής που αφορά τα POST requests που στάλθηκαν στο θύμα.

Παρακάτω φαίνεται η καταγραφή της κίνησης μόνο για τα TCP πακέτα:



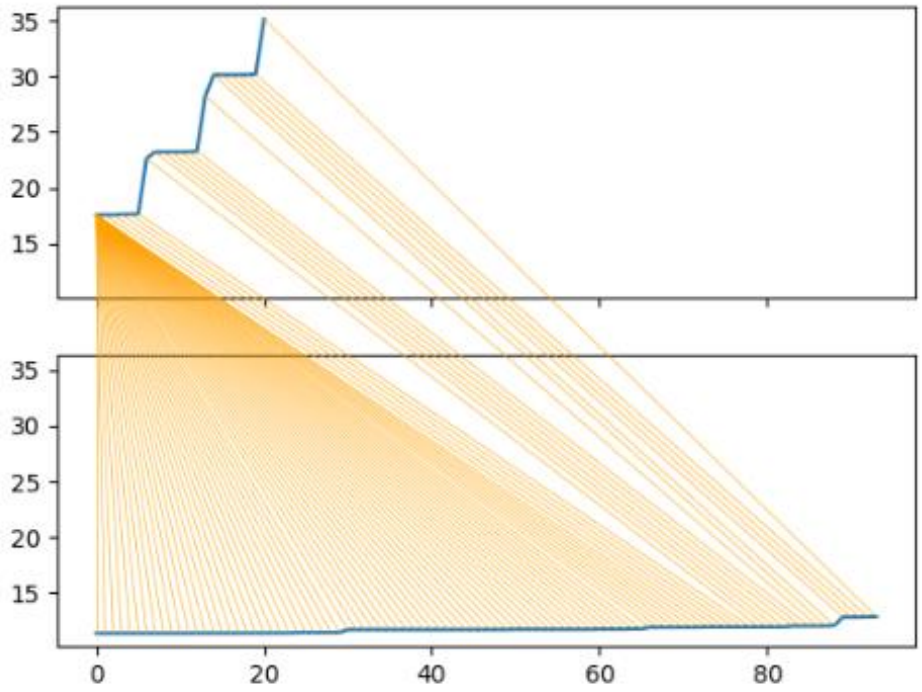
Εικόνα 43. Καταγραφή Wireshark για TCP πακέτα για την επίθεση brute force

Αν παρατηρήσουμε τις καταγραφές των δύο διαφορετικών κινήσεων, δηλαδή της απλής κίνησης και της κίνησης της επίθεσης, θα δούμε ότι στην πρώτη καταγραφή ανάμεσα στα POST requests υπάρχουν και άλλα HTTP πακέτα που έχουν προηγηθεί σε αντίθεση με την δεύτερη καταγραφή όπου τα POST requests διαδέχονται το ένα το άλλο χωρίς να εμφανίζεται κάποιο άλλο HTTP πακέτο ανάμεσα. Επιπλέον στην πρώτη καταγραφή σε κάθε αποτυχημένη προσπάθεια εμφανίζονται πακέτα που είναι GET requests για την σελίδα “fail.html”. Αντίθετα στην δεύτερη καταγραφή δεν εμφανίζονται καθόλου τέτοιου είδους πακέτα.

Συνεπώς μέσω παρατήρησης των καταγραφών προκύπτουν ορισμένες διαφορές μεταξύ των δύο κινήσεων. Ωστόσο προκειμένου να βγάλουμε έγκυρα συμπεράσματα απαιτείται η χρήση του αλγορίθμου Dynamic Time Warping (DTW) σε συνάρτηση με τις τρεις παραμέτρους: Time, Size, Interval.

4.3.3. Ανάλυση των καταγραφών με βάση την παράμετρο Time

Παρακάτω φαίνεται η ομοιότητα των δύο κινήσεων με βάση την παράμετρο Time με την βοήθεια του προγράμματος που χρησιμοποιεί τον αλγόριθμο Dynamic Time Warping:

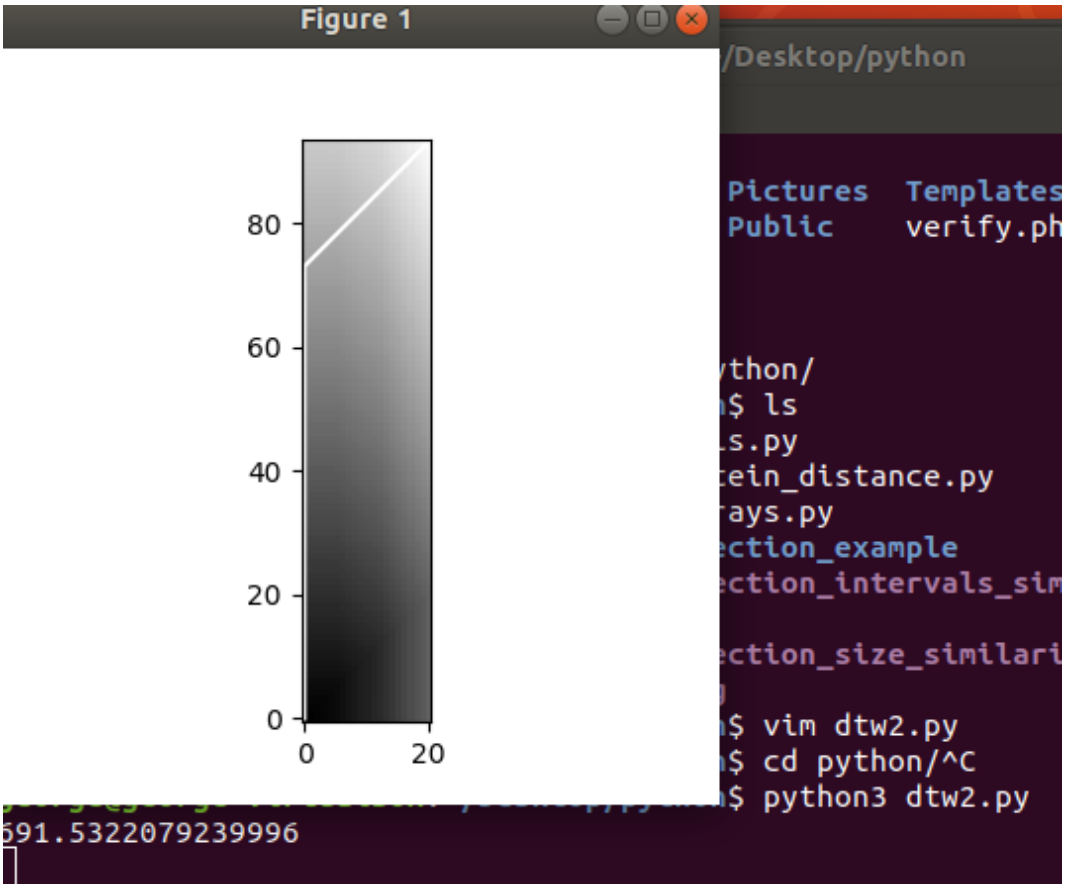


Εικόνα 44. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Time

Η πρώτη γραφική παράσταση αφορά την κίνηση ενός έγκυρου χρήστη, ενώ η δεύτερη γραφική παράσταση αφορά την κίνηση της επίθεσης bruteforce. Βλέπουμε ότι τα πακέτα στην πρώτη γραφική παράσταση τα πακέτα απέχουν σημαντικό χρονικό διάστημα μεταξύ τους σε αντίθεση με την δεύτερη γραφική παράσταση η οποίο δεν διαφέρει πολύ από μία οριζόντια ευθεία, γεγονός που σημαίνει ότι τα πακέτα στην δεύτερη γραφική παράσταση έχουν ελάχιστη χρονική διαφορά μεταξύ τους. Το παραπάνω συμπέρασμα δικαιολογείται από το γεγονός ότι στην πρώτη η κίνηση οι ενέργειες του χρήστη είναι αυτές που στέλνουν τα πακέτα το οποίο οδηγεί στην χρονική διαφορά των πακέτων. Αντίθετα στην δεύτερη κίνηση τα πακέτα στέλνονται από υπολογιστή και επομένως δεν υπάρχει καθυστέρηση μεταξύ των πακέτων.

Επιπλέον παρατηρείται ότι η δεύτερη γραφική παράσταση παρουσιάζει ομοιότητα ως προς συγκεκριμένα τμήματα της πρώτης γραφικής παράστασης, τα περισσότερα από τα οποία είναι μικρά ορίζονται ευθύγραμμα τμήματα.

Πέρα όμως από την ομοιότητα πρέπει να εξετάσουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



Εικόνα 45. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Time

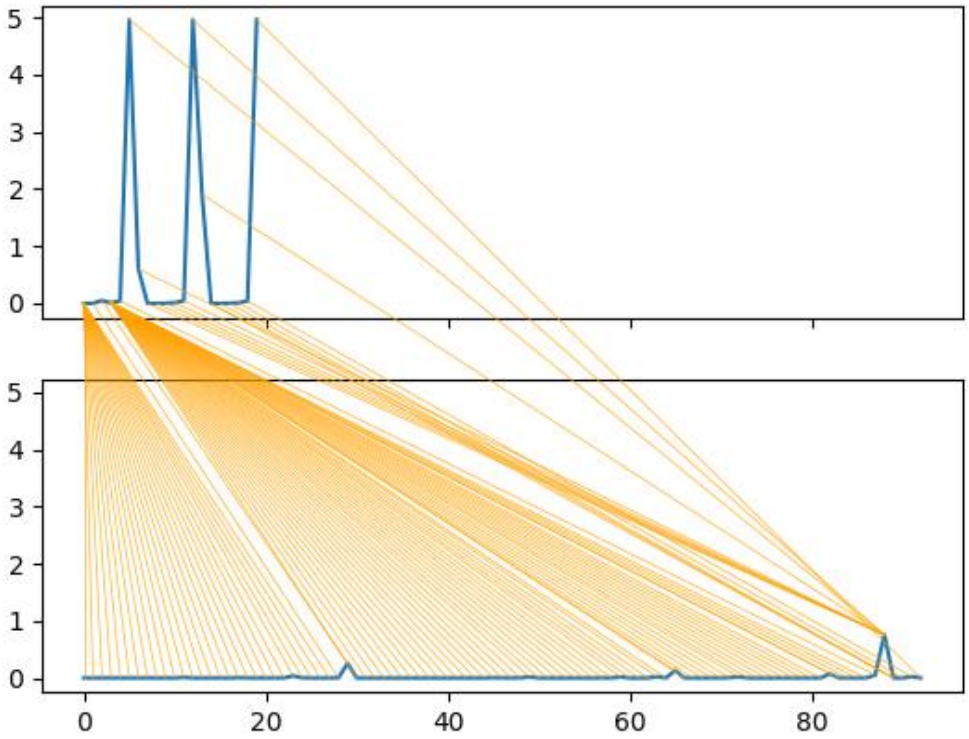
Βλέπουμε ότι το μεγαλύτερο μέρος του βέλτιστου μονοπατιού δεν αποτελείται από διαγώνιες κινήσεις, γεγονός που υποδηλώνει ότι οι δύο κινήσεις στα περισσότερα τμήματα δεν είναι όμοιες. Επίσης το ελάχιστο κόστος είναι περίπου ίσο με 691.5 το οποίο απέχει αρκετά από το μηδέν. Υπενθυμίζεται ότι όσο πιο κοντά στο μηδέν είναι το ελάχιστο κόστος τόσο μεγαλύτερη είναι ομοιότητα των δύο κινήσεων.

Συνεπώς με βάση την παράμετρο Time οι δύο καταγραφές παρουσιάζουν μικρή ομοιότητα. Ωστόσο δεν μπορούμε να βγάλουμε συμπέρασμα μόνο με βάση την παράμετρο Time, καθώς η τιμή της παραμέτρου Time αναπαριστά την χρονική διαφορά του πακέτου από το πρώτο frame. Επομένως στην διαφορά μεταξύ των δύο καταγραφών είναι πιθανό να συνέλαβε και το γεγονός ότι στην περίπτωση της

κίνησης που αφορά την επίθεση bruteforce ο υπολογιστής άρχισε να στέλνει νωρίτερα πακέτα σε σχέση με την κίνηση όπου ο χρήστης πληκτρολογούσε δεδομένα.

4.3.4. Ανάλυση των καταγραφών με βάση την παράμετρο Interval

Η επόμενη παράμετρος με βάση την οποία έγινε η ανάλυση των καταγραφών είναι τα intervals, που είναι η χρονική διαφορά μεταξύ δύο συνεχόμενων πακέτων. Παρακάτω φαίνεται η ομοιότητα των καταγράφων με βάση την παράμετρο αυτή:

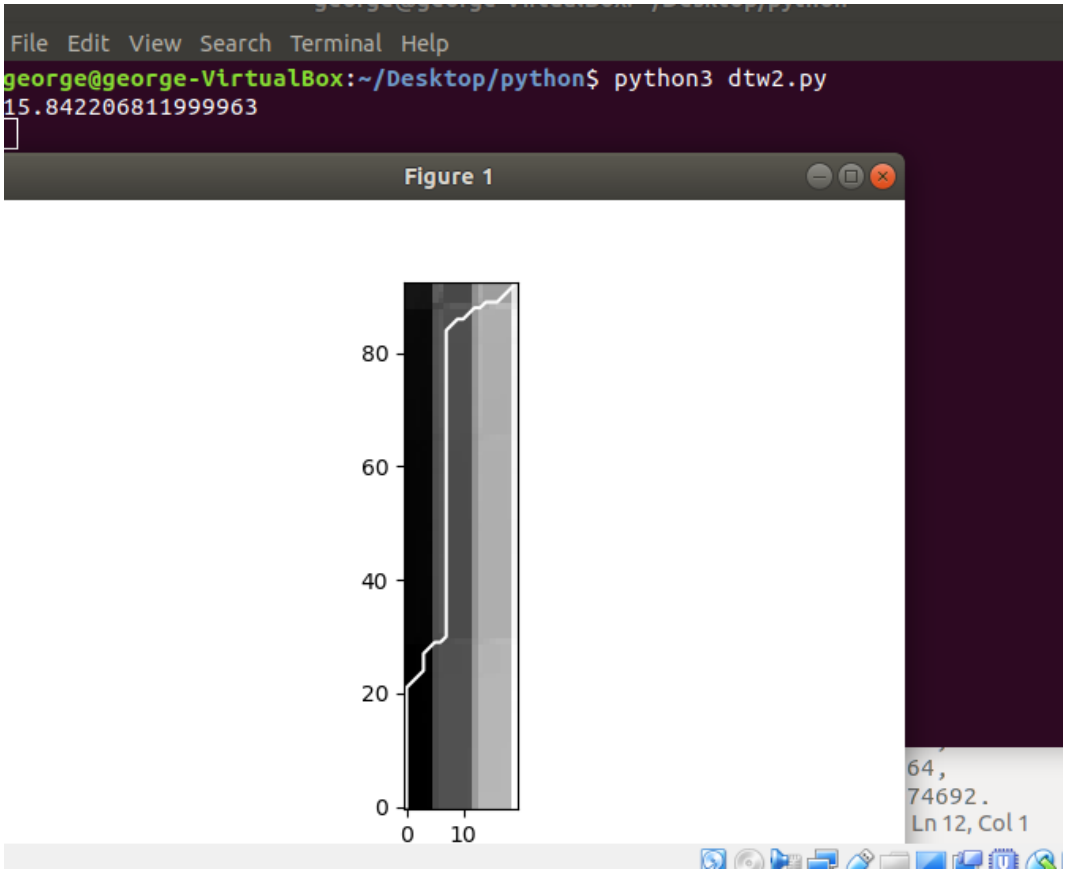


Εικόνα 46. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Interval

Η πρώτη γραφική παράσταση αφορά την κίνηση του έγκυρου χρήστη και η δεύτερη την κίνηση της επίθεσης.

Η δεύτερη γραφική παράσταση είναι όμοια μόνο ως προς συγκεκριμένα τμήματα της πρώτης γραφικής παράστασης. Επίσης η πρώτη κίνηση παρουσιάζει high peaks από intervals για κάθε προσπάθεια του χρήστη να συνδεθεί επιτυχώς σε αντίθεση με την δεύτερη κίνηση όπου έχουμε μόνο ένα peak το οποίο είναι αρκετά μικρότερο από τα peaks της πρώτης κίνησης. Αυτό οφείλεται στο γεγονός ότι ο υπολογιστής μπορεί να στέλνει μεγάλο αριθμό πακέτων λίγα μόλις δευτερόλεπτα.

Το παραπάνω συμπέρασμα μας προϊδεάζει για μία πιθανή επίθεση από υπολογιστή. Για να καταλήξουμε σε κάποιο συμπέρασμα όμως πρέπει να λάβουμε υπόψιν το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



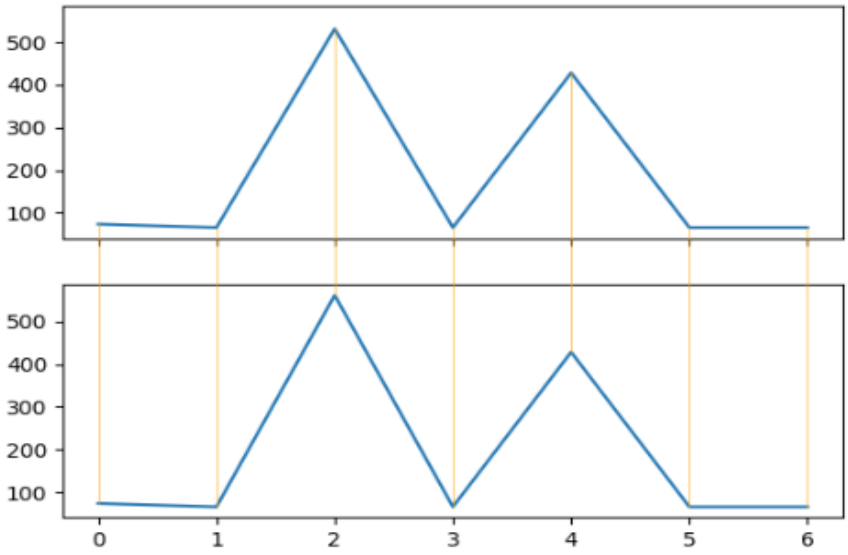
Εικόνα 47. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Interval

Βλέπουμε ότι το βέλτιστο μονοπάτι αποτελείται κυρίως από κάθετες κινήσεις και όχι από διαγώνιες κινήσεις γεγονός που δείχνει ότι οι δύο κινήσεις δεν είναι όμοιες στα περισσότερα σημεία. Το ελάχιστο κόστος είναι ίσο περίπου με 15.8 το οποίο δεν απέχει ούτε πολύ ούτε λίγο από το μηδέν.

Με βάση όλα τα παραπάνω μπορούμε να βγάλουμε το συμπέρασμα ότι οι δύο κινήσεις διαφέρουν με βάση την παράμετρο Interval.

4.3.5. Ανάλυση των καταγραφών με βάση την παράμετρο Size

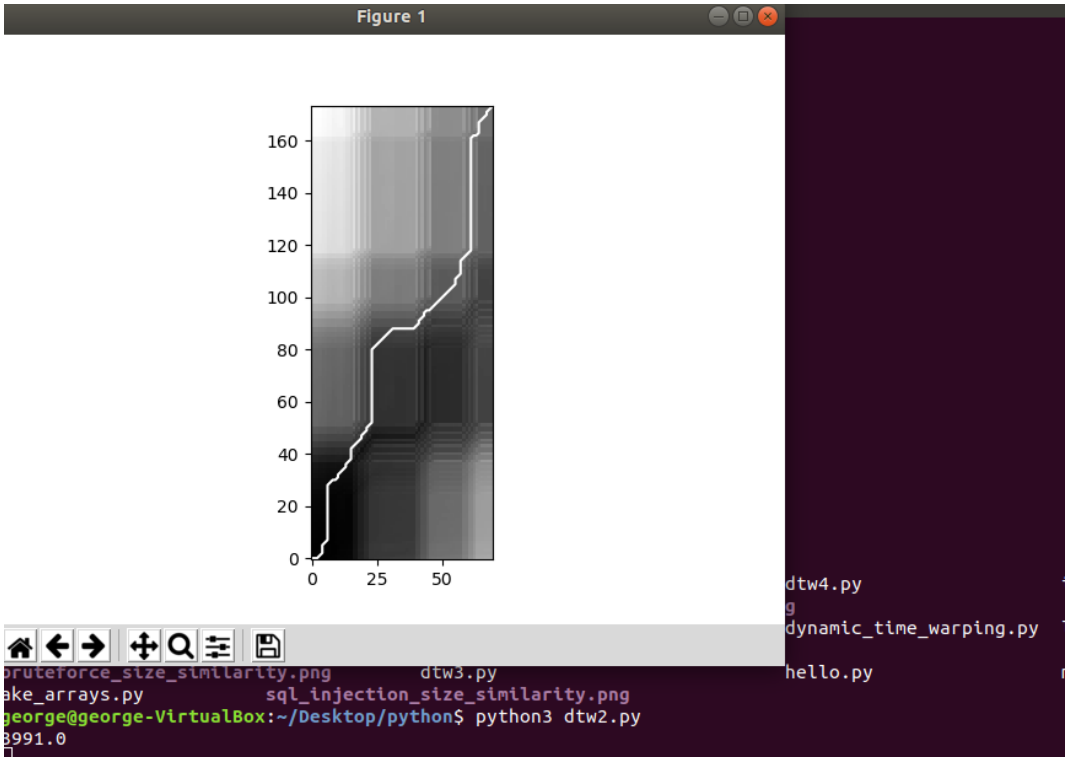
Η τελευταία παράμετρος που θα εξετάσουμε είναι η παράμετρος Size. Η ομοιότητα λοιπόν των κινήσεων με βάση την παράμετρο Size είναι:



Εικόνα 48. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Size

Βλέπουμε ότι οι δύο κινήσεις παρουσιάζουν μεγάλη ομοιότητα ως προς την παράμετρο Size. Αυτή η ομοιότητα οφείλεται στο γεγονός ότι και στις δύο κινήσεις τα πακέτα που στάλθηκαν είχαν παρόμοιο περιεχόμενο και κατά συνέπεια παρόμοιο μέγεθος.

Επίσης υπολογίζεται το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



Εικόνα 49. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης brute force με βάση την παράμετρο Size

Βλέπουμε ότι οι διαγώνιες κινήσεις έχουν αυξηθεί σε σχέση με τις προηγούμενες παραμέτρους χωρίς όμως να έχουν εξαφανισθεί οι κάθετες κινήσεις. Επομένως τα σημεία στα οποία οι καταγραφές είναι όμοιες είναι πιο πολλά. Όμως το ελάχιστο κόστος απέχει αρκετά από το μηδέν γεγονός που δείχνει ότι οι δύο κινήσεις δεν είναι αρκετά όμοιες.

4.3.6. Συμπέρασμα των αναλύσεων

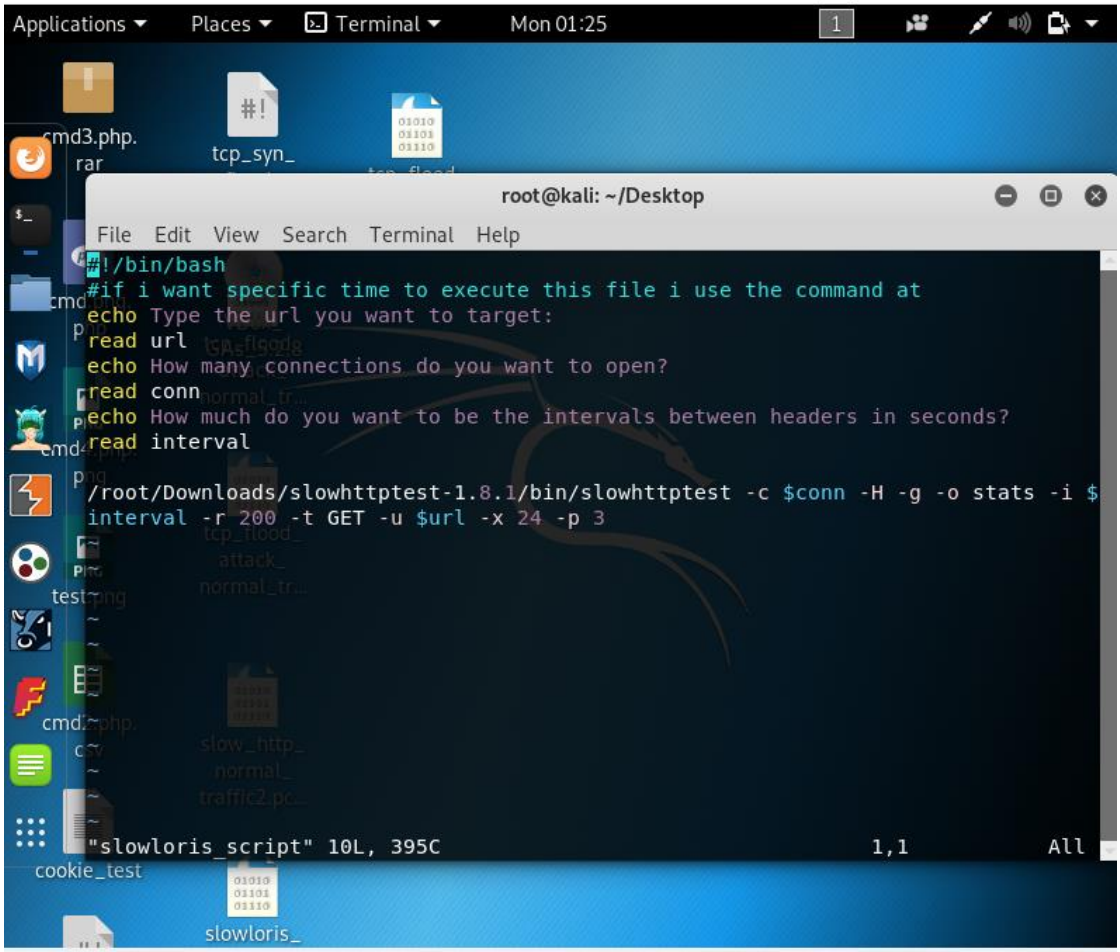
Με βάση τις παραπάνω αναλύσεις βλέπουμε ότι η μεγαλύτερη διαφορά μεταξύ των δύο κινήσεων παρατηρείται στην παράμετρο Time. Όλες όμως οι παράμετροι καταλήγουν σε διαφορές μεταξύ των δύο κινήσεων. Επομένως ένα σύστημα μπορεί να ξεχωρίσει μία επίθεση bruteforce από μία απλή κίνηση που προκλήθηκε από έναν χρήστη.

4.4. Slowloris attack

4.4.1. Προσομοίωση της επίθεσης

Η επίθεση slowloris ανήκει στην κατηγορία denial-of-service attacks (DDOS) και επιτίθεται σε έναν server ανοίγοντας και στην συνέχεια κρατώντας ανοιχτές πολλές HTTP συνδέσεις μεταξύ του επιτιθέμενου και του θύματος. Με τον τρόπο αυτό ένας μόνο υπολογιστής μπορεί να σταματήσει την λειτουργία ενός server (Cloudflare - Slowloris DDos Attack ,n.d.).

Στο πείραμα το θύμα διαθέτει server Apache ο οποίος χρησιμοποιείται για την λειτουργία της φόρμας. Για την προσομοίωση της επίθεσης slowloris δημιουργήθηκε ένα πρόγραμμα το οποίο χρησιμοποιεί ένα εργαλείο με το όνομα slowhttptest. Το εργαλείο αυτό είναι υπεύθυνο για την αποστολή των πακέτων στο θύμα. Παρακάτω φαίνεται πρόγραμμα που θα εκτελέσει την slowloris επίθεση:



Εικόνα 50. Script για την εκτέλεση της επίθεσης slowloris

Στην αρχή το πρόγραμμα ρωτάει τον χρήστη ποιο είναι το url του θύματος, πόσες συνδέσεις να ανοίξει ταυτόχρονα και πόση χρονική διάρκεια να διαφέρουν δύο συνεχόμενα πακέτα μεταξύ τους. Στην συνέχεια χρησιμοποιείται το εργαλείο

slowhttptest για το οποίο πρέπει να ρυθμιστούν ορισμένες επιλογές προκειμένου να λειτουργήσει. Οι επιλογές που χρησιμοποιήθηκαν στο πείραμα είναι:

- -c. Η επιλογή αυτή ορίζει τον αριθμό των συνδέσεων που θα ανοίξει το εργαλείο μεταξύ του θύματος και του επιτιθέμενου.
- -H. Η επιλογή αυτή ορίζει ότι θα πραγματοποιηθεί η επίθεση slowloris.
- -g και -o. Οι επιλογές αυτές χρησιμοποιούνται μαζί για να παράγουν στατιστικά σχετικά με το αποτέλεσμα εκτέλεσης του εργαλείου. Τα στατιστικά αποθηκεύονται στο όνομα του αρχείου που ακολουθεί μετά την επιλογή -o. Στην περίπτωσή μας το όνομα του αρχείου είναι το stats.
- -i. Η επιλογή αυτή ορίζει την χρονική διάρκεια σε δευτερόλεπτα που θα απέχουν τα πακέτα μεταξύ τους.
- -r. Πόσες συνδέσεις ανά δευτερόλεπτο πρέπει να ανοίξουν.
- -t. Τι είδος requests θα σταλθούν στο θύμα.
- -x. Το μέγιστο μήκος των πακέτων που θα σταλθούν.

- -p. Η επιλογή αυτή ορίζει πόσα δευτερόλεπτα να περιμένει το πρόγραμμα για απάντηση από το θύμα μέχρι να θεωρήσει τον server απρόσιτο (inaccessible).

(KaliTools – SlowHTTPTest Package Description, n.d.)

Στο πείραμα επιλέχθηκε το πρόγραμμα να ανοίξει 20 διαφορετικές συνδέσεις και να υπάρχει 10 δευτερόλεπτα χρονική διαφορά μεταξύ των πακέτων.

Πέρα από την επίθεση προσομοιώθηκε μία συνηθισμένη κίνηση πακέτων. Η κίνηση αυτή πραγματοποιήθηκε από το εικονικό μηχάνημα του επιτιθέμενου και αποτελείται από τέσσερα GET requests για την σελίδα που βρίσκεται η φόρμα, δηλαδή αποτελείται από τέσσερις προσπάθειες του χρήστη να συνδεθεί με την σελίδα στην οποία βρίσκεται η φόρμα (login.php).

4.4.2. Καταγραφή και ανάλυση της επίθεσης

Μετά την προσομοίωση των δύο κινήσεων ακολουθεί η καταγραφή αυτών. Παρακάτω φαίνεται η καταγραφή της συνηθισμένης κίνησης:

ip.src == 192.168.1.8 — Expression…

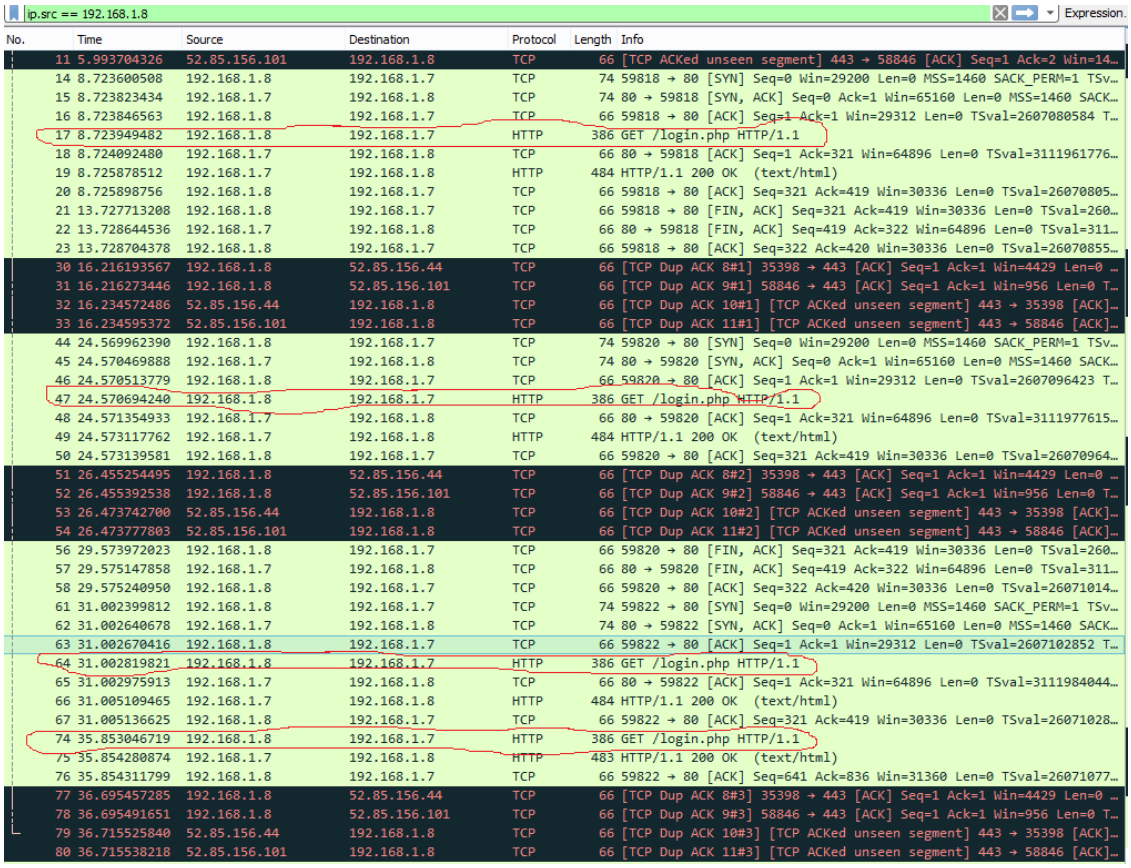
No.	Time	Source	Destination	Protocol	Length	Info
11	5.993704326	52.85.156.101	192.168.1.8	TCP	66	[TCP ACKed unseen segment] 443 → 58846 [ACK] Seq=1 Ack=2 Win=14…
14	8.723600508	192.168.1.8	192.168.1.7	TCP	74	59818 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
15	8.723823434	192.168.1.7	192.168.1.8	TCP	74	80 → 59818 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK…
16	8.723846563	192.168.1.8	192.168.1.7	TCP	66	59818 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607080584 T…
17	8.723949482	192.168.1.8	192.168.1.7	HTTP	386	GET /login.php HTTP/1.1
18	8.724092480	192.168.1.7	192.168.1.8	TCP	66	80 → 59818 [ACK] Seq=1 Ack=321 Win=64896 Len=0 TSval=3111961776…
19	8.725878512	192.168.1.7	192.168.1.8	HTTP	484	HTTP/1.1 200 OK (text/html)
20	8.725898756	192.168.1.8	192.168.1.7	TCP	66	59818 → 80 [ACK] Seq=321 Ack=419 Win=30336 Len=0 TSval=26070805…
21	13.727713208	192.168.1.8	192.168.1.7	TCP	66	59818 → 80 [FIN, ACK] Seq=321 Ack=419 Win=30336 Len=0 TSval=260…
22	13.728644536	192.168.1.7	192.168.1.8	TCP	66	80 → 59818 [FIN, ACK] Seq=419 Ack=322 Win=64896 Len=0 TSval=311…
23	13.728704378	192.168.1.8	192.168.1.7	TCP	66	59818 → 80 [ACK] Seq=322 Ack=420 Win=30336 Len=0 TSval=26070855…
30	16.216193567	192.168.1.8	52.85.156.44	TCP	66	[TCP Dup ACK 8#1] 35398 → 443 [ACK] Seq=1 Ack=1 Win=4429 Len=0 …
31	16.216273446	192.168.1.8	52.85.156.101	TCP	66	[TCP Dup ACK 9#1] 58846 → 443 [ACK] Seq=1 Ack=1 Win=956 Len=0 T…
32	16.234572486	52.85.156.44	192.168.1.8	TCP	66	[TCP Dup ACK 10#1] [TCP ACKed unseen segment] 443 → 35398 [ACK]…
33	16.234595372	52.85.156.101	192.168.1.8	TCP	66	[TCP Dup ACK 11#1] [TCP ACKed unseen segment] 443 → 58846 [ACK]…
44	24.569962390	192.168.1.8	192.168.1.7	TCP	74	59820 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
45	24.570469888	192.168.1.7	192.168.1.8	TCP	74	80 → 59820 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK…
46	24.570513779	192.168.1.8	192.168.1.7	TCP	66	59820 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607096423 T…
47	24.570694240	192.168.1.8	192.168.1.7	HTTP	386	GET /login.php HTTP/1.1
48	24.571354933	192.168.1.7	192.168.1.8	TCP	66	80 → 59820 [ACK] Seq=1 Ack=321 Win=64896 Len=0 TSval=3111977615…
49	24.573117762	192.168.1.7	192.168.1.8	HTTP	484	HTTP/1.1 200 OK (text/html)
50	24.573139581	192.168.1.8	192.168.1.7	TCP	66	59820 → 80 [ACK] Seq=321 Ack=419 Win=30336 Len=0 TSval=26070964…
51	26.455254495	192.168.1.8	52.85.156.44	TCP	66	[TCP Dup ACK 8#2] 35398 → 443 [ACK] Seq=1 Ack=1 Win=4429 Len=0 …
52	26.455392538	192.168.1.8	52.85.156.101	TCP	66	[TCP Dup ACK 9#2] 58846 → 443 [ACK] Seq=1 Ack=1 Win=956 Len=0 T…
53	26.473742700	52.85.156.44	192.168.1.8	TCP	66	[TCP Dup ACK 10#2] [TCP ACKed unseen segment] 443 → 35398 [ACK]…
54	26.473777803	52.85.156.101	192.168.1.8	TCP	66	[TCP Dup ACK 11#2] [TCP ACKed unseen segment] 443 → 58846 [ACK]…
56	29.573972023	192.168.1.8	192.168.1.7	TCP	66	59820 → 80 [FIN, ACK] Seq=321 Ack=419 Win=30336 Len=0 TSval=260…
57	29.575147858	192.168.1.7	192.168.1.8	TCP	66	80 → 59820 [FIN, ACK] Seq=419 Ack=322 Win=64896 Len=0 TSval=311…
58	29.575240950	192.168.1.8	192.168.1.7	TCP	66	59820 → 80 [ACK] Seq=322 Ack=420 Win=30336 Len=0 TSval=26071014…
61	31.002399812	192.168.1.8	192.168.1.7	TCP	74	59822 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
62	31.002640678	192.168.1.7	192.168.1.8	TCP	74	80 → 59822 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK…
63	31.002670416	192.168.1.8	192.168.1.7	TCP	66	59822 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607102852 T…
64	31.002819821	192.168.1.8	192.168.1.7	HTTP	386	GET /login.php HTTP/1.1
65	31.002975913	192.168.1.7	192.168.1.8	TCP	66	80 → 59822 [ACK] Seq=1 Ack=321 Win=64896 Len=0 TSval=3111984044…
66	31.005109465	192.168.1.7	192.168.1.8	HTTP	484	HTTP/1.1 200 OK (text/html)
67	31.005136625	192.168.1.8	192.168.1.7	TCP	66	59822 → 80 [ACK] Seq=321 Ack=419 Win=30336 Len=0 TSval=26071028…
74	35.853046719	192.168.1.8	192.168.1.7	HTTP	386	GET /login.php HTTP/1.1
75	35.854280874	192.168.1.7	192.168.1.8	HTTP	483	HTTP/1.1 200 OK (text/html)
76	35.854311799	192.168.1.8	192.168.1.7	TCP	66	59822 → 80 [ACK] Seq=641 Ack=836 Win=31360 Len=0 TSval=26071077…
77	36.695457285	192.168.1.8	52.85.156.44	TCP	66	[TCP Dup ACK 8#3] 35398 → 443 [ACK] Seq=1 Ack=1 Win=4429 Len=0 …
78	36.695491651	192.168.1.8	52.85.156.101	TCP	66	[TCP Dup ACK 9#3] 58846 → 443 [ACK] Seq=1 Ack=1 Win=956 Len=0 T…
79	36.715525840	52.85.156.44	192.168.1.8	TCP	66	[TCP Dup ACK 10#3] [TCP ACKed unseen segment] 443 → 35398 [ACK]…
80	36.715538218	52.85.156.101	192.168.1.8	TCP	66	[TCP Dup ACK 11#3] [TCP ACKed unseen segment] 443 → 58846 [ACK]…

Εικόνα 51. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης slowloris

Αξίζει να σημειωθεί ότι η παραπάνω εικόνα δεν παρουσιάζει όλα τα πακέτα της καταγραφής του Wireshark για την συγκεκριμένη κίνηση καθώς μία εικόνα δεν είναι δυνατό να περιέχει τον μεγάλο αριθμό των πακέτων που στάλθηκαν.

Στην παραπάνω καταγραφή παρατηρείται ότι έχει χρησιμοποιηθεί φίλτρο προκειμένου να εμφανίζονται μόνο τα πακέτα που έχουν ως αποστολέα τον εικονικό υπολογιστή του επιτιθέμενου. Το φίλτρο αυτό είναι το “ip.src == 192.168.1.8”.

Παρακάτω αποτυπώνεται η ίδια καταγραφή με επιλεγμένα όμως τέσσερα πακέτα:



Εικόνα 52. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης slowloris με σημειωμένα HTTP GET πακέτα

Τα επιλεγμένα πακέτα είναι τα πακέτα που στάλθηκαν για να ζητήσουν την σελίδα login.php, δηλαδή την σελίδα στην οποία βρίσκεται η φόρμα. Όπως γίνεται αντιληπτό τα πακέτα αυτά είναι τέσσερα όσες και οι προσπάθειες του χρήστη να συνδεθεί με την σελίδα login.php. Μία ακόμα σημαντική παρατήρηση είναι ότι πριν τα τέσσερα αυτά πακέτα υπάρχει ένα TCP πακέτο. Στα τρία πρώτα επιλεγμένα πακέτα υπάρχουν τρία TCP πακέτα, δύο TCP SYN και ένα TCP ACK. Τα τρία αυτά πακέτα δείχνουν ότι κάθε φορά δημιουργείται μία καινούρια σύνδεση μεταξύ του χρήστη και του εικονικού υπολογιστή του θύματος στο οποίο λειτουργεί η σελίδα login.php. Την διακοπή της κάθε σύνδεσης αναπαριστά το TCP πακέτο FIN, ACK. Ωστόσο παρατηρείται ότι πριν το τελευταίο επιλεγμένο πακέτο υπάρχει μόνο ένα TCP ACK πακέτο αντί για δύο TCP SYN πακέτα και ένα TCP ACK πακέτο. Ο λόγος που συμβαίνει αυτό είναι ότι δεν έχει περάσει αρκετός χρόνος μεταξύ της δημιουργίας σύνδεσης για την αποστολή του τρίτου επιλεγμένου πακέτου και της αποστολής του τέταρτου

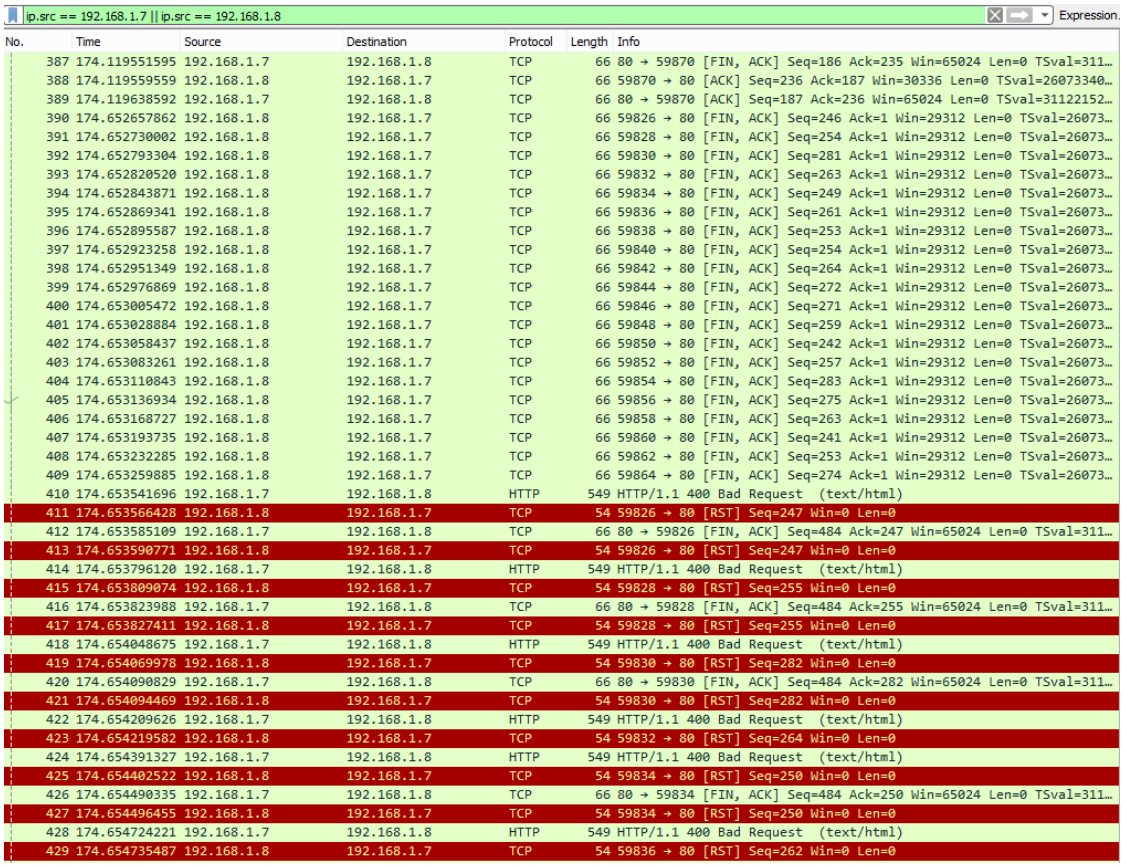
επιλεγμένου πακέτου με αποτέλεσμα η σύνδεση να έχει παραμείνει ανοιχτή κατά την αποστολή του τέταρτου επιλεγμένου πακέτου.

Παρακάτω παρουσιάζεται η καταγραφή της επίθεσης slowloris:

ip.src == 192.168.1.8 Expression

No.	Time	Source	Destination	Protocol	Length	Info
199	162.118353606	192.168.1.8	192.168.1.7	TCP	74	59824 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
200	162.118436665	192.168.1.8	192.168.1.7	TCP	74	59826 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
202	162.118674734	192.168.1.8	192.168.1.7	TCP	66	59824 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322096 T…
204	162.118702429	192.168.1.8	192.168.1.7	TCP	66	59826 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322097 T…
205	162.120165726	192.168.1.8	192.168.1.7	HTTP	300	GET / HTTP/1.1
206	162.120222248	192.168.1.8	192.168.1.7	TCP	298	59826 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
209	162.127209425	192.168.1.8	192.168.1.7	TCP	74	59828 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
211	162.127497841	192.168.1.8	192.168.1.7	TCP	66	59828 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322105 T…
212	162.132929530	192.168.1.8	192.168.1.7	TCP	74	59830 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
213	162.133057165	192.168.1.8	192.168.1.7	TCP	298	59828 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
215	162.133869933	192.168.1.8	192.168.1.7	TCP	66	59830 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322112 T…
217	162.138740130	192.168.1.8	192.168.1.7	TCP	74	59832 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
218	162.138825198	192.168.1.8	192.168.1.7	TCP	298	59830 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
220	162.139502248	192.168.1.8	192.168.1.7	TCP	66	59832 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322117 T…
222	162.144720891	192.168.1.8	192.168.1.7	TCP	74	59834 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
223	162.144844915	192.168.1.8	192.168.1.7	TCP	298	59832 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
225	162.145587753	192.168.1.8	192.168.1.7	TCP	66	59834 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322123 T…
227	162.150512699	192.168.1.8	192.168.1.7	TCP	74	59836 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
228	162.150641844	192.168.1.8	192.168.1.7	TCP	298	59834 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
230	162.151748845	192.168.1.8	192.168.1.7	TCP	66	59836 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322130 T…
233	162.155691639	192.168.1.8	192.168.1.7	TCP	66	59824 → 80 [ACK] Seq=235 Ack=186 Win=30336 Len=0 TSval=26073221…
235	162.155802007	192.168.1.8	192.168.1.7	TCP	74	59838 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
236	162.155889513	192.168.1.8	192.168.1.7	TCP	66	59824 → 80 [FIN, ACK] Seq=235 Ack=187 Win=30336 Len=0 TSval=260…
238	162.155986531	192.168.1.8	192.168.1.7	TCP	66	59838 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322134 T…
240	162.156040306	192.168.1.8	192.168.1.7	TCP	298	59836 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
242	162.161212919	192.168.1.8	192.168.1.7	TCP	74	59840 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
243	162.161289180	192.168.1.8	192.168.1.7	TCP	298	59838 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
245	162.161389237	192.168.1.8	192.168.1.7	TCP	66	59840 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322139 T…
247	162.170591891	192.168.1.8	192.168.1.7	TCP	74	59842 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
248	162.170744511	192.168.1.8	192.168.1.7	TCP	298	59840 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
250	162.170919890	192.168.1.8	192.168.1.7	TCP	66	59842 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322149 T…
252	162.175905675	192.168.1.8	192.168.1.7	TCP	74	59844 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
253	162.175998836	192.168.1.8	192.168.1.7	TCP	298	59842 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
255	162.176296424	192.168.1.8	192.168.1.7	TCP	66	59844 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322154 T…
257	162.181220436	192.168.1.8	192.168.1.7	TCP	74	59846 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
258	162.181331735	192.168.1.8	192.168.1.7	TCP	298	59844 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
260	162.181611586	192.168.1.8	192.168.1.7	TCP	66	59846 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322159 T…
262	162.186586123	192.168.1.8	192.168.1.7	TCP	74	59848 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
263	162.186742306	192.168.1.8	192.168.1.7	TCP	298	59846 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
265	162.187113121	192.168.1.8	192.168.1.7	TCP	66	59848 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322165 T…
267	162.192569385	192.168.1.8	192.168.1.7	TCP	74	59850 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
268	162.192659143	192.168.1.8	192.168.1.7	TCP	298	59848 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=232 TSval=26073…
270	162.192934233	192.168.1.8	192.168.1.7	TCP	66	59850 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2607322171 T…

Εικόνα 53. Πρώτο μέρος της καταγραφής Wireshark για την επίθεση slowloris



ip.src == 192.168.1.7 || ip.src == 192.168.1.8

No.	Time	Source	Destination	Protocol	Length	Info
387	174.119551595	192.168.1.7	192.168.1.8	TCP	66	80 → 59870 [FIN, ACK] Seq=186 Ack=235 Win=65024 Len=0 TSval=311…
388	174.119559559	192.168.1.8	192.168.1.7	TCP	66	59870 → 80 [ACK] Seq=236 Ack=187 Win=30336 Len=0 TSval=26073340…
389	174.119638592	192.168.1.7	192.168.1.8	TCP	66	80 → 59870 [ACK] Seq=187 Ack=236 Win=65024 Len=0 TSval=31122152…
390	174.652657862	192.168.1.8	192.168.1.7	TCP	66	59826 → 80 [FIN, ACK] Seq=246 Ack=1 Win=29312 Len=0 TSval=26073…
391	174.652730002	192.168.1.8	192.168.1.7	TCP	66	59828 → 80 [FIN, ACK] Seq=254 Ack=1 Win=29312 Len=0 TSval=26073…
392	174.652793304	192.168.1.8	192.168.1.7	TCP	66	59830 → 80 [FIN, ACK] Seq=281 Ack=1 Win=29312 Len=0 TSval=26073…
393	174.652820520	192.168.1.8	192.168.1.7	TCP	66	59832 → 80 [FIN, ACK] Seq=263 Ack=1 Win=29312 Len=0 TSval=26073…
394	174.652843871	192.168.1.8	192.168.1.7	TCP	66	59834 → 80 [FIN, ACK] Seq=249 Ack=1 Win=29312 Len=0 TSval=26073…
395	174.652869341	192.168.1.8	192.168.1.7	TCP	66	59836 → 80 [FIN, ACK] Seq=261 Ack=1 Win=29312 Len=0 TSval=26073…
396	174.652895587	192.168.1.8	192.168.1.7	TCP	66	59838 → 80 [FIN, ACK] Seq=253 Ack=1 Win=29312 Len=0 TSval=26073…
397	174.652923258	192.168.1.8	192.168.1.7	TCP	66	59840 → 80 [FIN, ACK] Seq=254 Ack=1 Win=29312 Len=0 TSval=26073…
398	174.652951349	192.168.1.8	192.168.1.7	TCP	66	59842 → 80 [FIN, ACK] Seq=264 Ack=1 Win=29312 Len=0 TSval=26073…
399	174.652976869	192.168.1.8	192.168.1.7	TCP	66	59844 → 80 [FIN, ACK] Seq=272 Ack=1 Win=29312 Len=0 TSval=26073…
400	174.653005472	192.168.1.8	192.168.1.7	TCP	66	59846 → 80 [FIN, ACK] Seq=271 Ack=1 Win=29312 Len=0 TSval=26073…
401	174.653028884	192.168.1.8	192.168.1.7	TCP	66	59848 → 80 [FIN, ACK] Seq=259 Ack=1 Win=29312 Len=0 TSval=26073…
402	174.653058437	192.168.1.8	192.168.1.7	TCP	66	59850 → 80 [FIN, ACK] Seq=242 Ack=1 Win=29312 Len=0 TSval=26073…
403	174.653083261	192.168.1.8	192.168.1.7	TCP	66	59852 → 80 [FIN, ACK] Seq=257 Ack=1 Win=29312 Len=0 TSval=26073…
404	174.653110843	192.168.1.8	192.168.1.7	TCP	66	59854 → 80 [FIN, ACK] Seq=283 Ack=1 Win=29312 Len=0 TSval=26073…
405	174.653136934	192.168.1.8	192.168.1.7	TCP	66	59856 → 80 [FIN, ACK] Seq=275 Ack=1 Win=29312 Len=0 TSval=26073…
406	174.653168727	192.168.1.8	192.168.1.7	TCP	66	59858 → 80 [FIN, ACK] Seq=263 Ack=1 Win=29312 Len=0 TSval=26073…
407	174.653193735	192.168.1.8	192.168.1.7	TCP	66	59860 → 80 [FIN, ACK] Seq=241 Ack=1 Win=29312 Len=0 TSval=26073…
408	174.653232285	192.168.1.8	192.168.1.7	TCP	66	59862 → 80 [FIN, ACK] Seq=253 Ack=1 Win=29312 Len=0 TSval=26073…
409	174.653259885	192.168.1.8	192.168.1.7	TCP	66	59864 → 80 [FIN, ACK] Seq=274 Ack=1 Win=29312 Len=0 TSval=26073…
410	174.653541696	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
411	174.653566428	192.168.1.8	192.168.1.7	TCP	54	59826 → 80 [RST] Seq=247 Win=0 Len=0
412	174.653585109	192.168.1.7	192.168.1.8	TCP	66	80 → 59826 [FIN, ACK] Seq=484 Ack=247 Win=65024 Len=0 TSval=311…
413	174.653590771	192.168.1.8	192.168.1.7	TCP	54	59826 → 80 [RST] Seq=247 Win=0 Len=0
414	174.653796120	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
415	174.653809074	192.168.1.8	192.168.1.7	TCP	54	59828 → 80 [RST] Seq=255 Win=0 Len=0
416	174.653823988	192.168.1.7	192.168.1.8	TCP	66	80 → 59828 [FIN, ACK] Seq=484 Ack=255 Win=65024 Len=0 TSval=311…
417	174.653827411	192.168.1.8	192.168.1.7	TCP	54	59828 → 80 [RST] Seq=255 Win=0 Len=0
418	174.654048675	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
419	174.654069978	192.168.1.8	192.168.1.7	TCP	54	59830 → 80 [RST] Seq=282 Win=0 Len=0
420	174.654090829	192.168.1.7	192.168.1.8	TCP	66	80 → 59830 [FIN, ACK] Seq=484 Ack=282 Win=65024 Len=0 TSval=311…
421	174.654094469	192.168.1.8	192.168.1.7	TCP	54	59830 → 80 [RST] Seq=282 Win=0 Len=0
422	174.654209626	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
423	174.654219582	192.168.1.8	192.168.1.7	TCP	54	59832 → 80 [RST] Seq=264 Win=0 Len=0
424	174.654391327	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
425	174.654402522	192.168.1.8	192.168.1.7	TCP	54	59834 → 80 [RST] Seq=250 Win=0 Len=0
426	174.654490335	192.168.1.7	192.168.1.8	TCP	66	80 → 59834 [FIN, ACK] Seq=484 Ack=250 Win=65024 Len=0 TSval=311…
427	174.654496455	192.168.1.8	192.168.1.7	TCP	54	59834 → 80 [RST] Seq=250 Win=0 Len=0
428	174.654724221	192.168.1.7	192.168.1.8	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
429	174.654735487	192.168.1.8	192.168.1.7	TCP	54	59836 → 80 [RST] Seq=262 Win=0 Len=0

Εικόνα 54. Δεύτερο μέρος της καταγραφής Wireshark για την επίθεση slowloris

Η καταγραφή της επίθεσης slowloris περιέχει αρκετά πακέτα τα οποία δεν είναι δυνατόν να παρουσιαστούν όλα σε μία εικόνα. Για τον λόγο αυτό εμφανίζονται σε δύο εικόνες δύο σημαντικά κομμάτια της καταγραφής, στην πρώτη εικόνα τα πακέτα TCP SYN, TCP ACK και στην δεύτερη εικόνα τα πακέτα TCP FIN-ACK.

Στην πρώτη εικόνα λοιπόν της καταγραφής έχει χρησιμοποιηθεί φίλτρο προκειμένου να εμφανίζονται μόνο τα πακέτα που έχουν σταλθεί από το εικονικό μηχάνημα του επιτιθέμενου (“ip.src==192.168.1.8”) , ενώ στην δεύτερη εικόνα έχει χρησιμοποιηθεί φίλτρο προκειμένου να εμφανίζονται τα πακέτα που έχουν ως αποστολέα είτε τον επιτιθέμενο είτε το θύμα (“ip.src==192.168.1.8 || ip.src==192.168.1.7”).

Στην πρώτη εικόνα της καταγραφής τα περισσότερα πακέτα που εμφανίζονται είναι πακέτα TCP SYN ή TCP ACK χωρίς να ακολουθεί αμέσως πακέτο TCP FIN,ACK. Επιπλέον βλέπουμε ότι κάθε TCP SYN πακέτο στέλνεται σε διαφορετική θύρα (port) κάθε φορά με αποτέλεσμα για κάθε τέτοιο πακέτα να δημιουργείται μία νέα σύνδεση.

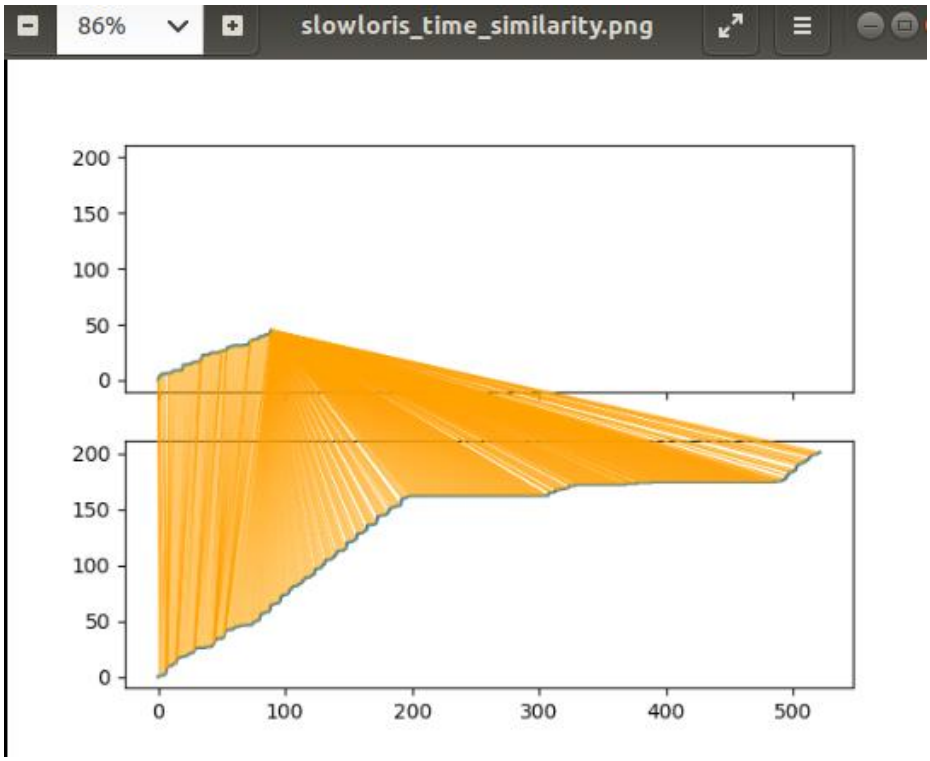
Στην δεύτερη εικόνα της καταγραφής φαίνονται τα πακέτα ανταλλάσσουν μεταξύ τους ο επιτιθέμενος και το θύμα όταν τελείωσε η επίθεση και ο επιτιθέμενος επιθυμεί να κλείσει τις συνδέσεις που έχει ανοίξει. Πέρα από τα πακέτα TCP FIN ,ACK, τα οποία δείχνουν την επιθυμία του αποστολέα να κλείσει την σύνδεση,

παρατηρούνται πακέτα TCP RST. Τα πακέτα αυτά στέλνονται ως απάντηση στα πακέτα TCP FIN, ACK που έστειλε το θύμα και δηλώνουν ότι η σύνδεση για την οποία στάλθηκαν FIN, ACK πακέτα από το θύμα δεν είναι πλέον έγκυρη (Stackoverflow - TCP RST packet details).

Επομένως η κύρια διαφορά που παρατηρείται μεταξύ των καταγραφών των δύο κινήσεων είναι ότι στην δεύτερη καταγραφή τα πακέτα προκειμένου να δημιουργηθεί μία σύνδεση στέλνονται με πολύ μικρή χρονική διαφορά μεταξύ τους σε αντίθεση με την πρώτη καταγραφή όπου τα TCP SYN πακέτα στέλνονται διάσπαρτα. Για να εντοπίσουμε όμως τις διαφορές μεταξύ των δύο καταγραφών είναι απαραίτητη η ανάλυση των καταγραφών με βάση τις παραμέτρους Time, Interval και Size.

4.4.3. Ανάλυση των καταγραφών με βάση την παράμετρο Time

Αρχικά παρουσιάζεται η ομοιότητα των καταγραφών των δύο κινήσεων με βάση την παράμετρο Time:

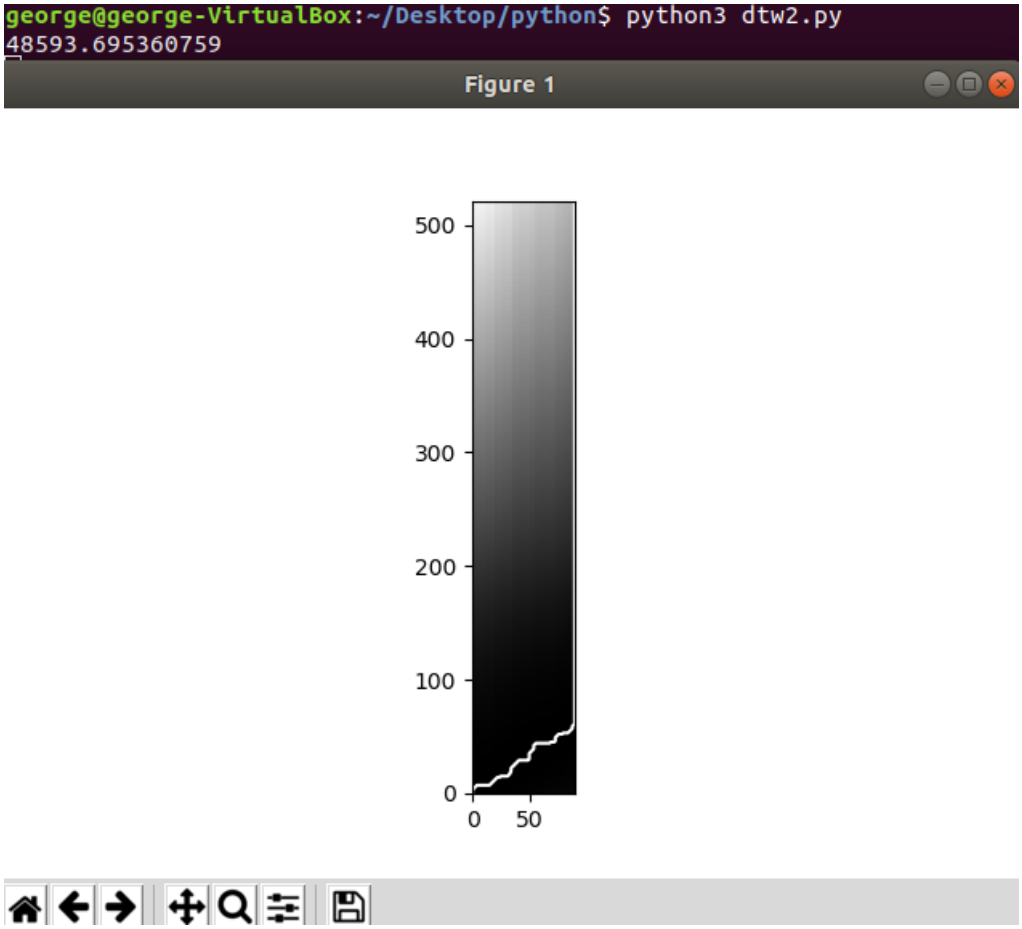


Εικόνα 55. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Time

Η πρώτη γραφική παράσταση αφορά την καταγραφή που απεικονίζει μία συνηθισμένη κίνηση πακέτων κατά την οποία στέλνονται τέσσερα GET requests. Συνεπώς η δεύτερη γραφική παράσταση αφορά την καταγραφή της επίθεσης slowloris.

Αρχικά η δεύτερη γραφική παράσταση είναι αρκετά μεγαλύτερη από την πρώτη το οποίο είναι αναμενόμενο καθώς κατά την προσομοίωση της επίθεσης στέλνονται

πολλά περισσότερα TCP πακέτα σε σχέση με την συνηθισμένη κίνηση πακέτων. Στην συνέχεια παρατηρούμε ότι το μεγαλύτερο κομμάτι της δεύτερης γραφικής παράστασης, δηλαδή το κομμάτι μεταξύ των αριθμών 100 και 500, περιέχει σημεία τα οποία αντιστοιχίζονται κυρίως στο τελευταίο σημείο της πρώτης γραφικής παράστασης. Επιπλέον βλέπουμε ότι στο κομμάτι της δεύτερης γραφικής παράστασης πριν από τον αριθμό 100 τα σημεία της αντιστοιχίζονται σε πολλά σημεία της γραφικής παράστασης. Ωστόσο η παρατήρηση αυτή δεν μπορεί να οδηγήσει στο συμπέρασμα ότι οι δύο γραφικές παραστάσεις έχουν μεγάλη ομοιότητα καθώς το μέγεθος των γραφικών παραστάσεων διαφέρει αρκετά , γεγονός που δυσκολεύει να καταλάβουμε εάν τα σημεία της πρώτης γραφικής παράστασης στα οποία αντιστοιχίζονται τα σημεία της δεύτερης γραφικής παράστασης , είναι διαφορετικά ή όχι. Για τον λόγο αυτό υπολογίζεται το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



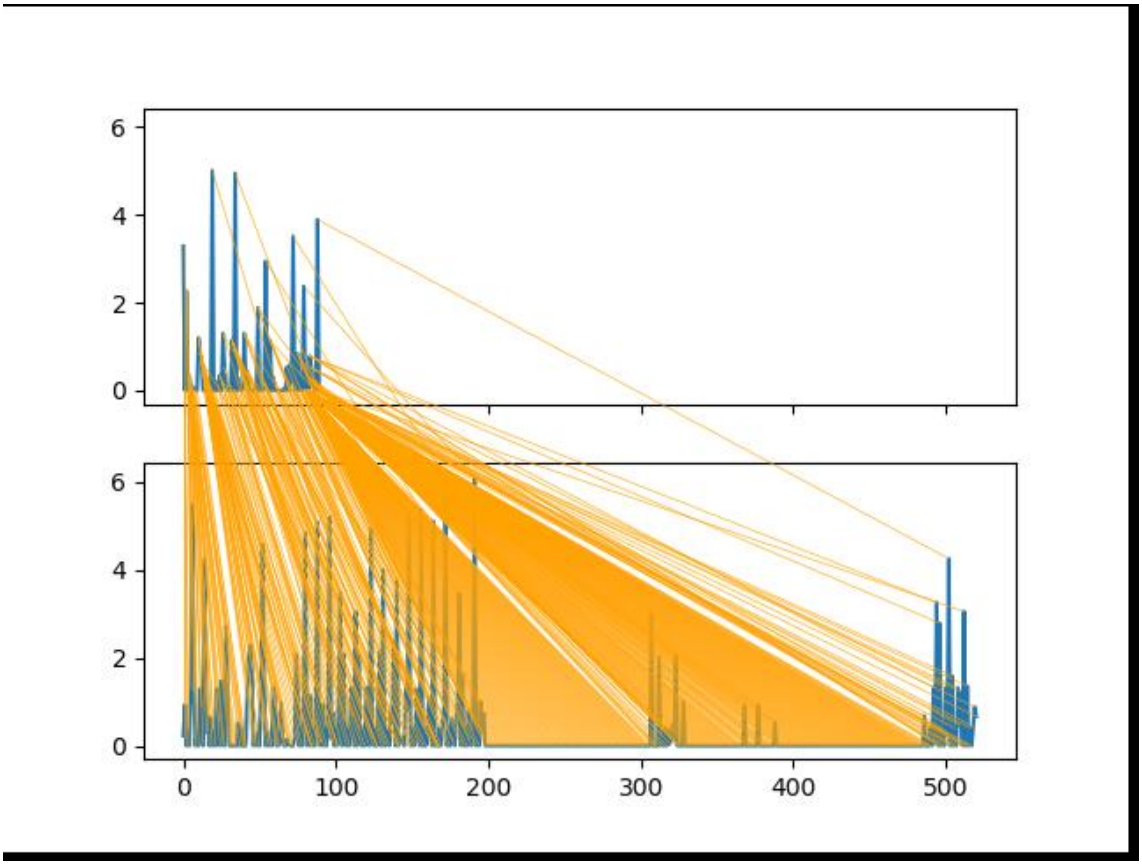
Εικόνα 56. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Time

Βλέπουμε ότι το βέλτιστο μονοπάτι αρχικά αποτελείται από διαγώνιες κινήσεις, που υποδεικνύουν ότι στο κομμάτι αυτό οι δύο γραφικές παραστάσεις παρουσιάζουν ομοιότητα. Στην συνέχεια όμως το βέλτιστο μονοπάτι αποτελείται από κάθετη κίνηση και άρα οι δύο συναρτήσεις δεν είναι όμοιες στο κομμάτι αυτό. Επιπλέον

παρατηρείται ότι το ελάχιστο κόστος είναι ίσο περίπου με 48593.7, αριθμός που απέχει αρκετά από το μηδέν. Επομένως γίνεται αντιληπτό ότι οι δύο κινήσεις δεν παρουσιάζουν μεγάλη ομοιότητα με βάση την παράμετρο Time. Όπως έχει αναφερθεί στις αναλύσεις των προηγούμενων επιθέσεων, η παράμετρος Time εξαρτάται από το πότε στάλθηκε το πρώτο frame στην καταγραφή του Wireshark , συνεπώς είναι σημαντικό να εξεταστούν και οι άλλες δύο παράμετροι.

4.4.4. Ανάλυση των καταγραφών με βάση την παράμετρο Interval

Η επόμενη ανάλυση των καταγραφών έγινε με βάση την παράμετρο Interval. Παρακάτω φαίνεται η ομοιότητα των καταγραφών:



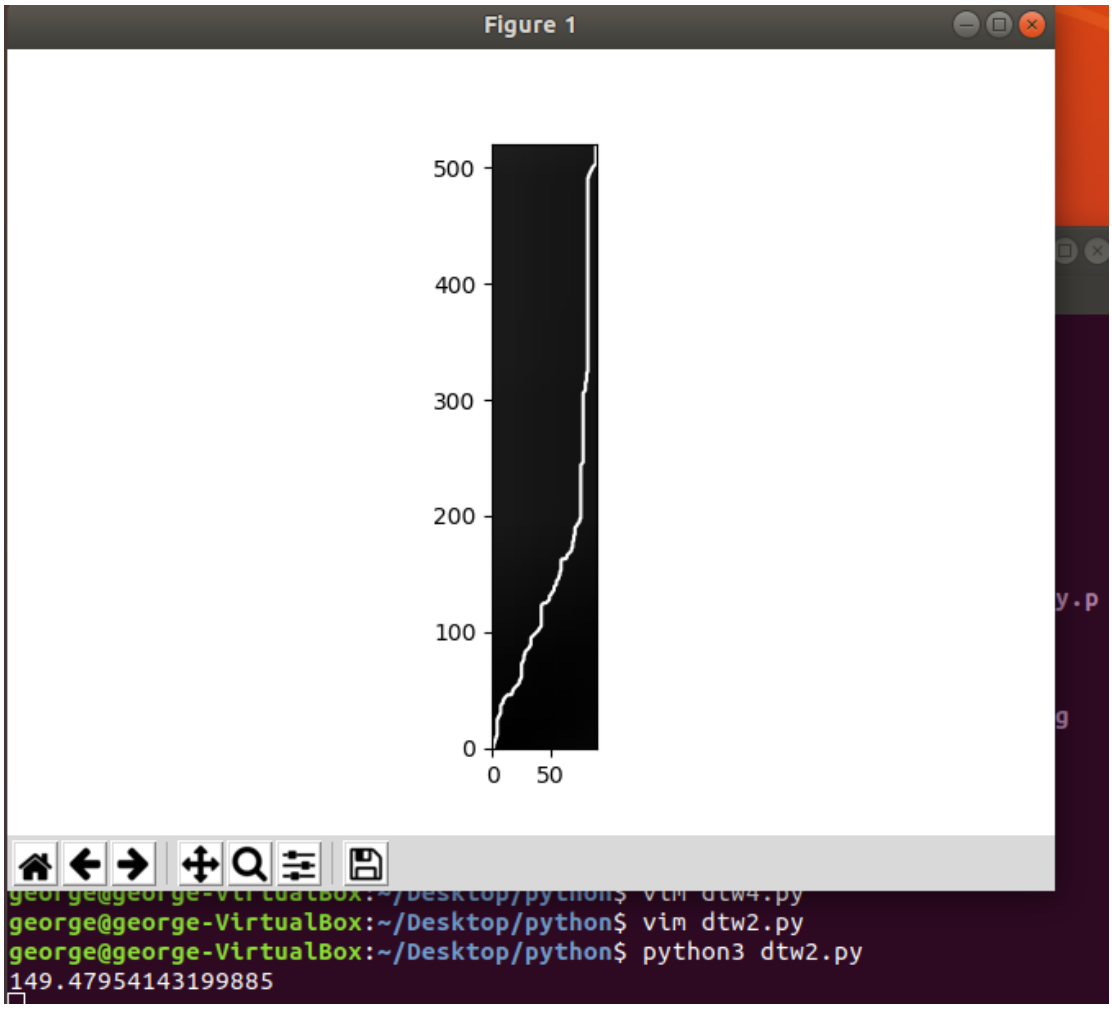
Εικόνα 57. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Interval

Η πρώτη γραφική παράσταση αφορά την συνηθισμένη κίνηση και η δεύτερη την επίθεση.

Παρατηρείται ότι η δεύτερη γραφική παράσταση αποτελείται από κομμάτια τα οποία είναι όμοια με αρκετά σημεία της πρώτης γραφικής παράστασης. Τα κομμάτια αυτά είναι αυτά πριν από την τιμή 200 και μετά την τιμή 500. Ωστόσο, μεταξύ των

κομματιών αυτών υπάρχουν ευθύγραμμα τμήματα που η τιμή τους είναι αρκετά κοντά στο μηδέν. Το γεγονός ότι η τιμή τους είναι αρκετά κοντά στο μηδέν σημαίνει ότι τα πακέτα που ανήκουν σε αυτά τα ευθύγραμμα τμήματα έχουν πολύ μικρή χρονική διαφορά μεταξύ τους. Οπότε είτε ο αποστολέας έστειλε πολλά πακέτα μαζί , το οποίο γίνεται μέσω κάποιας αυτοματοποιημένης διαδικασίας, είτε το θύμα έστειλε πολλά πακέτα μαζί. Και οι δύο αυτές περιπτώσεις συνέβησαν στο προσομοίωση της επίθεσης slowloris. Τα κομμάτια της δεύτερης γραφικής παράστασης που είναι όμοια με αυτά της πρώτης γραφικής παράστασης είναι πιθανό να αφορούν πακέτα τύπου broadcast προκειμένου να γίνει γνωστός ο κάτοχος της κάθε IP διεύθυνσης.

Στην συνέχεια βλέπουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



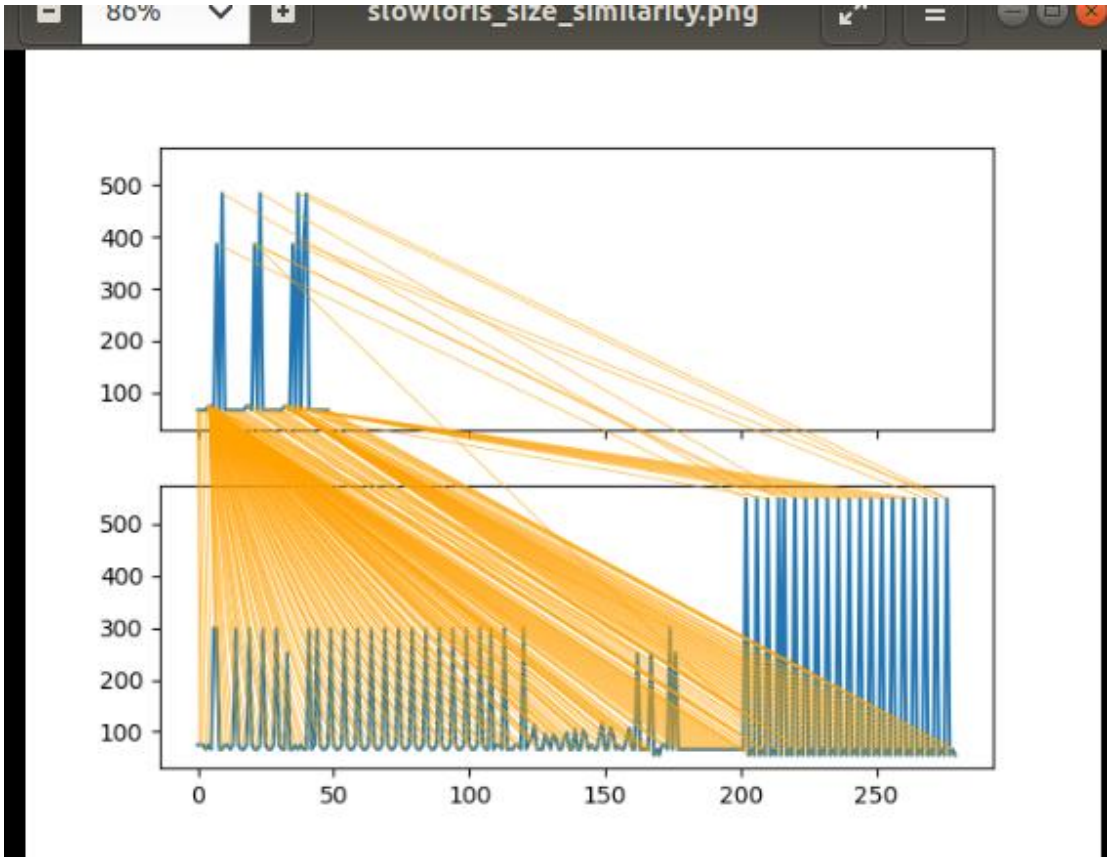
Εικόνα 58. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Interval

Παρατηρείται ότι το βέλτιστο μονοπάτι αποτελείται από μικρές διαγώνιες κινήσεις αλλά επίσης και από μεγάλες κάθετες κινήσεις κατά το τελευταίο κομμάτι του. Επομένως, στην αρχή οι δύο καταγραφές παρουσιάζουν αρκετές ομοιότητες, αλλά

στην συνέχεια παρουσιάζουν σημαντικές διαφορές όπως προκύπτει και από την εικόνα που συγκρίνει την ομοιότητα των δύο καταγραφών. Επιπλέον το ελάχιστο μονοπάτι είναι περίπου ίσο με 149.48, τιμή που απέχει αρκετά από το μηδέν έτσι ώστε να συμπεράνουμε ότι οι δύο καταγραφές παρουσιάζουν διαφορές.

4.4.5. Ανάλυση των καταγραφών με βάση την παράμετρο Size

Η τελευταία παράμετρος που εξετάζεται είναι η παράμετρος Size. Παρακάτω φαίνεται η ομοιότητα των δύο καταγραφών με βάση την παράμετρο Size:

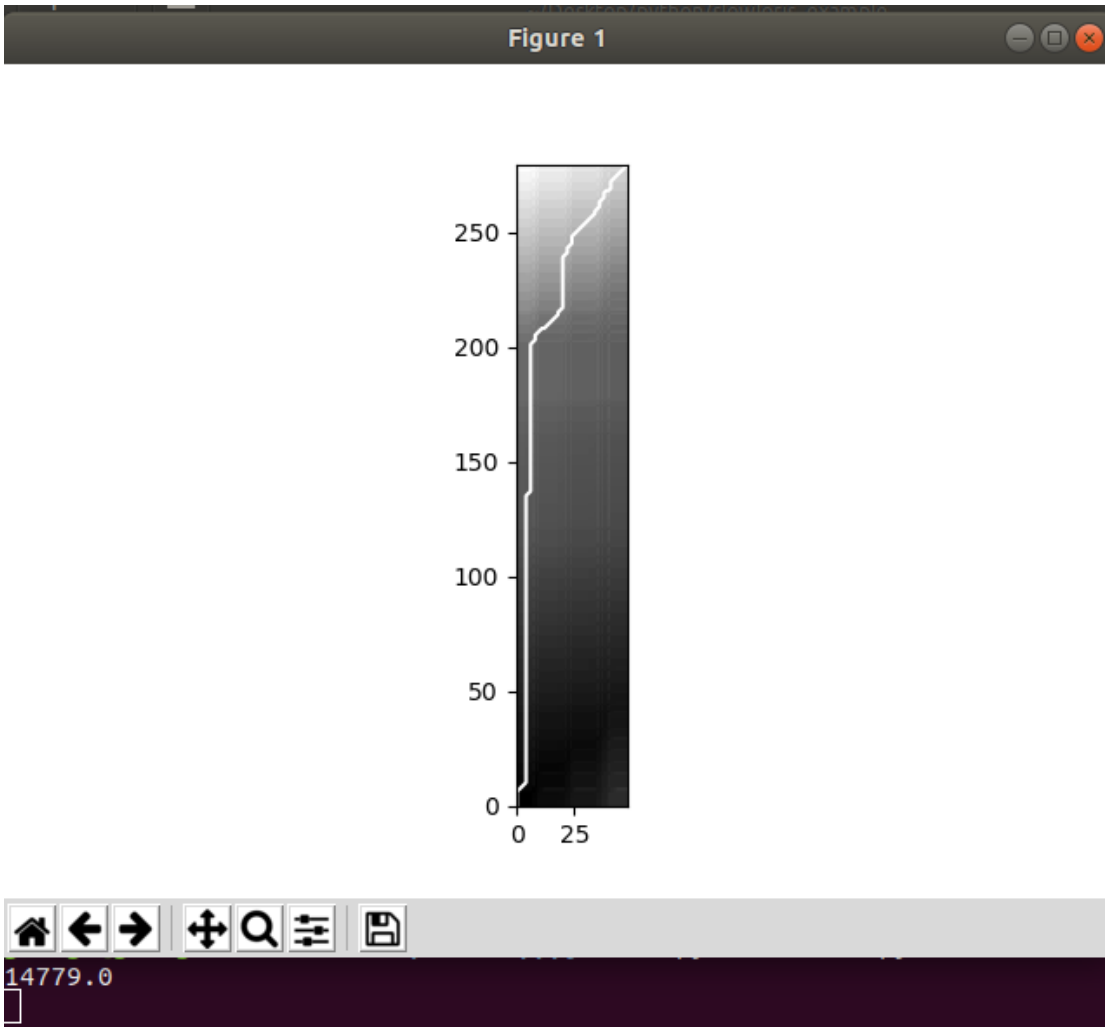


Εικόνα 59. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Size

Η πρώτη γραφική παράσταση αφορά την καταγραφή που αντιπροσωπεύει μία συνηθισμένη κίνηση στο δίκτυο, ενώ η δεύτερη αφορά την επίθεση slowloris.

Βλέπουμε ότι η δεύτερη γραφική παράσταση παρουσιάζει αρκετές ομοιότητες στα σημεία της γραφικής παράστασης που έχουν την μικρότερη τιμή. Ωστόσο αυτό δεν οδηγεί σε κάποιο συμπέρασμα. Επίσης παρατηρείται ότι στην δεύτερη γραφική παράσταση για x < 200 εμφανίζονται κορυφές με μικρή τιμή. Αυτό οφείλεται στο γεγονός ότι στην επίθεση slowloris τα πακέτα που στέλνονται στο θύμα δεν είναι ολοκληρωμένα και επομένως έχουν μικρότερο μέγεθος από ένα άλλο παρόμοιο πακέτο το οποίο στέλνεται ολοκληρωμένο.

Προκειμένου να καταλήξουμε σε κάποιο συμπέρασμα για την ανάλυση των δύο καταγραφών με βάση την παράμετρο Size είναι απαραίτητο να εξετάσουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



Εικόνα 60. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης slowloris με βάση την παράμετρο Size

Γίνεται αντιληπτό ότι το βέλτιστο μονοπάτι αποτελείται κυρίως από κάθετες κινήσεις, γεγονός που σημαίνει ότι οι δύο καταγραφές παρουσιάζουν διαφορές στο μεγαλύτερο κομμάτι τους. Επίσης, το ελάχιστο κόστος είναι ίσο με 14779, αριθμός που απέχει αρκετά από το μηδέν. Συνεπώς οι δύο καταγραφές παρουσιάζουν αρκετές διαφορές με βάση την παράμετρο Size.

4.4.6. Συμπέρασμα των αναλύσεων

Με βάση λοιπόν τις παραπάνω αναλύσεις η παράμετρος που οδηγεί στις μεγαλύτερες διαφορές μεταξύ των δύο καταγραφών είναι η παράμετρος Size. Το παραπάνω συμπέρασμα θα πρέπει να θεωρείται αναμενόμενο, καθώς στην επίθεση slowloris ο επιτιθέμενος στέλνει μη ολοκληρωμένα πακέτα προκειμένου να κρατήσει την σύνδεση μεταξύ του ίδιου και του θύματος ανοιχτή. Όλες όμως οι παράμετροι

οδηγούν σε διαφορές μεταξύ των καταγραφών. Επομένως το σύστημα θα θεωρήσει μία επίθεση slowloris ως πιθανή απειλή.

4.5. TCP SYN flood attack

4.5.1. Προσομοίωση της επίθεσης

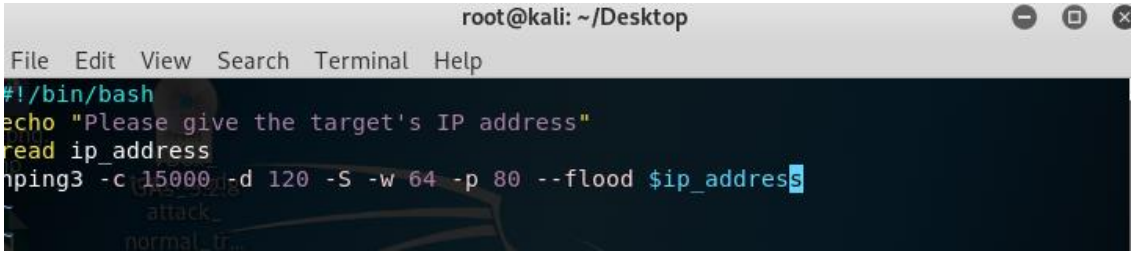
Η επίθεση TCP SYN flood αποτελεί επίθεση τύπου denial-of-service (DDOS). Προκειμένου να γίνει κατανοητή η παραπάνω επίθεση είναι αναγκαίο να δούμε πώς ξεκινάει μία σύνδεση μεταξύ client και server.

Προκειμένου λοιπόν να δημιουργηθεί μία σύνδεση μεταξύ client και server και στην συνέχεια να στείλει δεδομένα, για παράδειγμα μία ιστοσελίδας, ο server στον client πρέπει να πραγματοποιηθούν τα παρακάτω βήματα:

- Ο client στέλνει στον server TCP SYN (synchronize) πακέτο.
- Ο server αποδέχεται το πακέτο στέλνοντας TCP SYN-ACK (synchronize-acknowledge) πακέτο στον client.
- Ο client απαντά με ένα TCP ACK (acknowledge) πακέτο και δημιουργείται η σύνδεση.

Κατά την επίθεση TCP SYN flood ο επιτιθέμενος στέλνει TCP SYN πακέτα στο θύμα με διαφορετική θύρα κάθε φορά. Το θύμα στέλνει TCP SYN-ACK πακέτα στον επιτιθέμενο περιμένοντας ως απάντηση TCP ACK πακέτα. Ο επιτιθέμενος όμως δεν στέλνει TCP ACK πακέτα με αποτέλεσμα το θύμα να κρατάει ανοιχτή κάθε υποψήφια σύνδεση η οποία έχει προκύψει από τα πακέτα που στάλθηκαν. Αυτό οδηγεί σε πολλές ανοιχτές συνδέσεις χωρίς σκοπό και κατά συνέπεια το θύμα δεν μπορεί να εξυπηρετήσει άλλα αιτήματα για συνδέσεις από μη κακόβουλους clients (imperva – TCP SYN Flood, n.d.).

Για το πείραμα στην γραμμή εντολών του εικονικού μηχανήματος του θύματος εκτελέστηκε η εντολή «python -m SimpleHTTPServer» προκειμένου να δημιουργηθεί ένας τοπικός test server. Στην συνέχεια εκτελέστηκε πρόγραμμα το οποίο πραγματοποιεί την επίθεση tcp syn flood. Παρακάτω φαίνεται το πρόγραμμα:



Εικόνα 61. Script για την εκτέλεση της επίθεσης TCP SYN flood

Το παραπάνω πρόγραμμα εμφανίζει μήνυμα στον χρήστη να δώσει την IP διεύθυνση του θύματος. Στην συνέχεια δέχεται ως είσοδο την IP διεύθυνση που πληκτρολόγησε

ο χρήστης και μέσω του εργαλείου hping3 εκτελείται η επίθεση. Προκειμένου όμως να χρησιμοποιηθεί το εργαλείο hping3 πρέπει να ρυθμιστούν ορισμένες επιλογές πρώτα:

- -c. Η επιλογή αυτή ορίζει τον αριθμό των πακέτων που θα σταλθούν.
- -d. Η επιλογή αυτή ορίζει το μέγεθος των δεδομένων του κάθε πακέτου.
- -S. Η επιλογή αυτή θέτει την σημαία SYN προκειμένου να σταλθούν TCP SYN πακέτα.
- -w. Καθορίζει πόσα δεδομένα σε bytes μπορεί να δεχθεί η συσκευή προορισμού.
- -p. Η επιλογή αυτή προσδιορίζει την θύρα προορισμού στην οποία θα σταλθούν τα πακέτα.
- --flood. Μέσω της επιλογής αυτής το εργαλείο στέλνει πακέτα όσο πιο γρήγορα γίνεται χωρίς να περιμένει απάντηση για κάθε πακέτο.

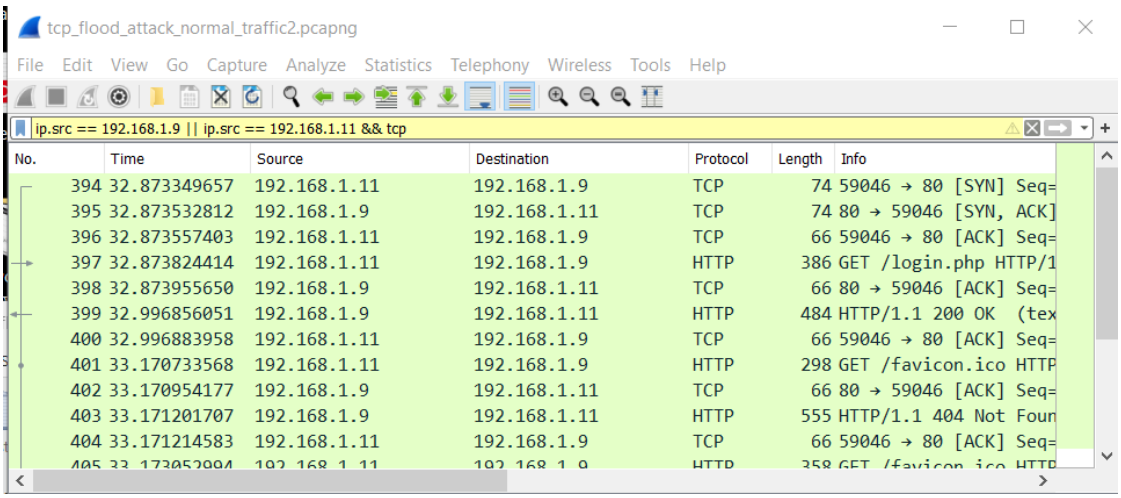
Στο πείραμα στάλθηκαν 15000 πακέτα με 120 bytes ως δεδομένα και θύρα προορισμού 80. Είναι σημαντικό να σημειωθεί ότι λόγω του μεγάλου αριθμού των πακέτων που στάλθηκαν δεν ήταν δυνατό να καταγραφούν όλα τα πακέτα που στάλθηκαν κατά την διάρκεια της επίθεσης, καθώς το εικονικό μηχάνημα του επιτιθέμενου στο οποίο γινόταν η καταγραφή δεν έχει αρκετή μνήμη με αποτέλεσμα να σταματάει να λειτουργεί.

Πέρα από την επίθεση στάλθηκε request για την login φόρμα προκειμένου να προσομοιωθεί μία συνηθισμένη κίνηση καταγραφών.

4.5.2. Καταγραφή και ανάλυση της επίθεσης

Στην συνέχεια παρουσιάζονται οι καταγραφές των παραπάνω κινήσεων μέσω του εργαλείου Wireshark.

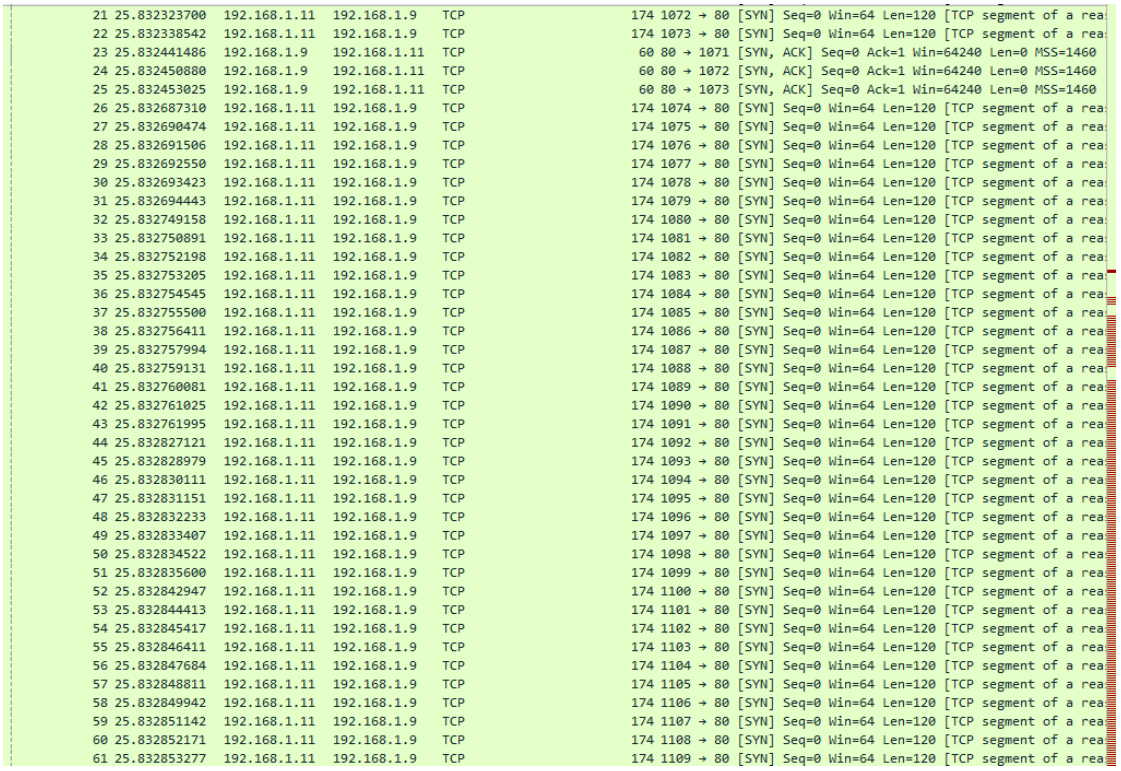
Παρακάτω φαίνεται η καταγραφή της κίνησης που προσομοιάζει μία συνηθισμένη κίνηση πακέτων:



Εικόνα 62. Καταγραφή Wireshark για συνηθισμένη κίνηση κατά την προσομοίωση της επίθεσης TCP SYN flood

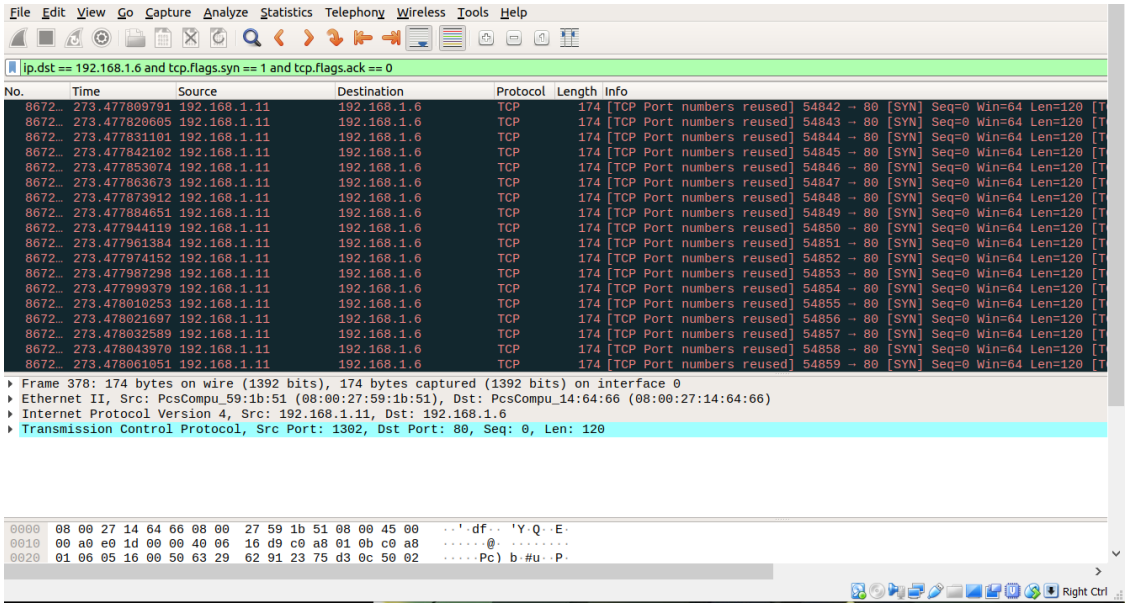
Στην παραπάνω εικόνα φαίνεται ένα GET request για την login φόρμα. Πριν το πακέτο που αφορά το GET request φαίνεται μία χειραψία TCP μεταξύ client και server, δηλαδή ένα πακέτο SYN από τον επιτιθέμενο (client), ένα πακέτο SYN-ACK από το θύμα (server) και τέλος ένα πακέτο ACK από τον επιτιθέμενο (client). Είναι σημαντικό να σημειωθεί ότι στην παραπάνω καταγραφή έχει χρησιμοποιηθεί φίλτρο για να φαίνονται TCP/HTTP πακέτα τα οποία στάλθηκαν είτε από τον επιτιθέμενο είτε από το θύμα.

Στην συνέχεια παρουσιάζεται η καταγραφή των πακέτων της επίθεσης TCP SYN flood:



21	25.832323700	192.168.1.11	192.168.1.9	TCP	174	1072 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
22	25.832338542	192.168.1.11	192.168.1.9	TCP	174	1073 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
23	25.832441486	192.168.1.9	192.168.1.11	TCP	60	80 → 1071 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
24	25.832450880	192.168.1.9	192.168.1.11	TCP	60	80 → 1072 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
25	25.832453025	192.168.1.9	192.168.1.11	TCP	60	80 → 1073 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
26	25.832687310	192.168.1.11	192.168.1.9	TCP	174	1074 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
27	25.832690474	192.168.1.11	192.168.1.9	TCP	174	1075 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
28	25.832691506	192.168.1.11	192.168.1.9	TCP	174	1076 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
29	25.832692550	192.168.1.11	192.168.1.9	TCP	174	1077 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
30	25.832693423	192.168.1.11	192.168.1.9	TCP	174	1078 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
31	25.832694443	192.168.1.11	192.168.1.9	TCP	174	1079 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
32	25.832749158	192.168.1.11	192.168.1.9	TCP	174	1080 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
33	25.832750891	192.168.1.11	192.168.1.9	TCP	174	1081 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
34	25.832752198	192.168.1.11	192.168.1.9	TCP	174	1082 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
35	25.832753205	192.168.1.11	192.168.1.9	TCP	174	1083 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
36	25.832754545	192.168.1.11	192.168.1.9	TCP	174	1084 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
37	25.832755500	192.168.1.11	192.168.1.9	TCP	174	1085 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
38	25.832756411	192.168.1.11	192.168.1.9	TCP	174	1086 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
39	25.832757994	192.168.1.11	192.168.1.9	TCP	174	1087 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
40	25.832759131	192.168.1.11	192.168.1.9	TCP	174	1088 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
41	25.832760081	192.168.1.11	192.168.1.9	TCP	174	1089 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
42	25.832761025	192.168.1.11	192.168.1.9	TCP	174	1090 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
43	25.832761995	192.168.1.11	192.168.1.9	TCP	174	1091 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
44	25.832827121	192.168.1.11	192.168.1.9	TCP	174	1092 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
45	25.832828979	192.168.1.11	192.168.1.9	TCP	174	1093 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
46	25.832830111	192.168.1.11	192.168.1.9	TCP	174	1094 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
47	25.832831151	192.168.1.11	192.168.1.9	TCP	174	1095 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
48	25.832832233	192.168.1.11	192.168.1.9	TCP	174	1096 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
49	25.832833407	192.168.1.11	192.168.1.9	TCP	174	1097 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
50	25.832834522	192.168.1.11	192.168.1.9	TCP	174	1098 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
51	25.832835600	192.168.1.11	192.168.1.9	TCP	174	1099 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
52	25.832842947	192.168.1.11	192.168.1.9	TCP	174	1100 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
53	25.832844413	192.168.1.11	192.168.1.9	TCP	174	1101 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
54	25.832845417	192.168.1.11	192.168.1.9	TCP	174	1102 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
55	25.832846411	192.168.1.11	192.168.1.9	TCP	174	1103 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
56	25.832847684	192.168.1.11	192.168.1.9	TCP	174	1104 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
57	25.832848811	192.168.1.11	192.168.1.9	TCP	174	1105 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
58	25.832849942	192.168.1.11	192.168.1.9	TCP	174	1106 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
59	25.832851142	192.168.1.11	192.168.1.9	TCP	174	1107 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
60	25.832852171	192.168.1.11	192.168.1.9	TCP	174	1108 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea
61	25.832853277	192.168.1.11	192.168.1.9	TCP	174	1109 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a rea

Εικόνα 63. Πρώτο μέρος της καταγραφής Wireshark για την επίθεση TCP SYN flood



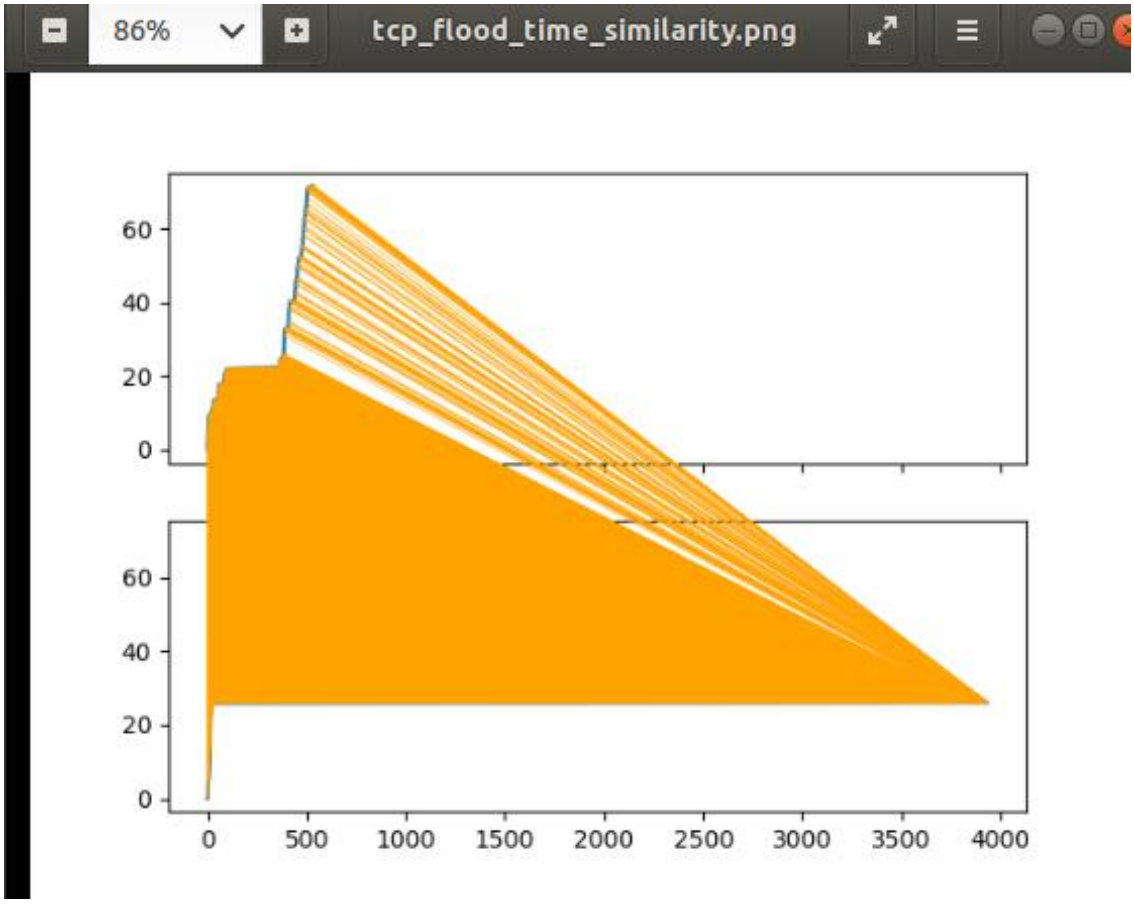
Εικόνα 64. Δεύτερο μέρος της καταγραφής Wireshark για την επίθεση TCP SYN flood

Στην δεύτερη εικόνα της παραπάνω καταγραφής έχει χρησιμοποιηθεί φίλτρο προκειμένου να εμφανίζονται τα TCP SYN πακέτα τα οποία έχουν ως προορισμό την IP διεύθυνση του εικονικού μηχανήματος του θύματος (“ip.dst == 192.168.1.6 and tcp.flags.syn == 1 and tcp.flags.ack == 0”). Επίσης λόγω του μεγάλου αριθμού των πακέτων δεν ήταν δυνατό να παρουσιαστεί ολόκληρη η καταγραφή της επίθεσης σε μία εικόνα οπότε παρουσιάζεται ένα κομμάτι της καταγραφής.

Η κύρια διαφορά μεταξύ των δύο καταγραφών είναι ότι στην πρώτη καταγραφή μετά από το πακέτο TCP SYN ακολουθεί πακέτο TCP SYN-ACK, ενώ όπως φαίνεται στην πρώτη εικόνα της δεύτερης καταγραφής μετά τα πακέτα TCP SYN δεν ακολουθεί κάποιο πακέτο TCP SYN-ACK. Προκειμένου να εντοπιστούν οι διαφορές μεταξύ των δύο καταγραφών είναι απαραίτητο να γίνει ανάλυση των καταγραφών αυτών με βάση τις παραμέτρους: Time, Interval και Size.

4.5.3. Ανάλυση των καταγραφών με βάση την παράμετρο Time

Αρχικά εξετάζεται η ομοιότητα των καταγραφών των δύο κινήσεων με βάση την παράμετρο Time:



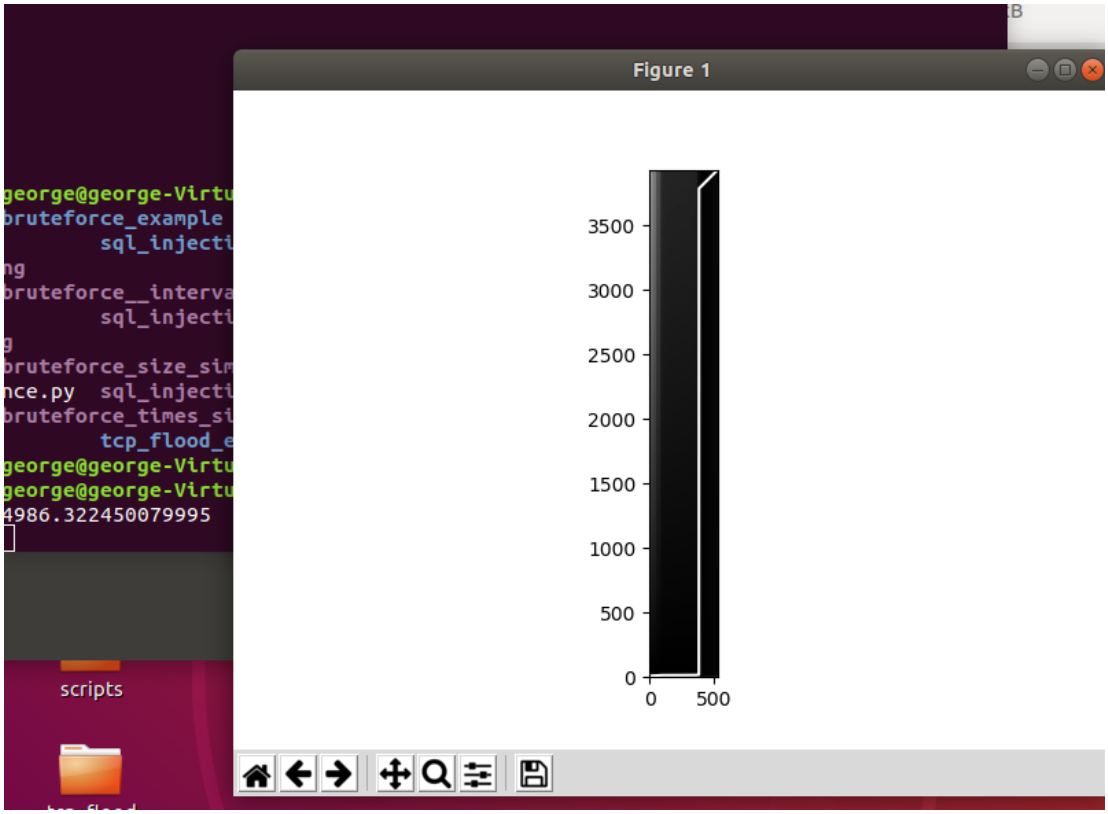
Εικόνα 65. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Time

Η πρώτη γραφική παράσταση αφορά την καταγραφή της συνηθισμένης κίνησης, ενώ η δεύτερη γραφική παράσταση αφορά την επίθεση.

Είναι σημαντικό να σημειωθεί ότι λόγω του μεγάλου αριθμού των πακέτων της επίθεσης δεν ήταν δυνατόν να χρησιμοποιηθούν όλα τα πακέτα της δεύτερης καταγραφής στην ανάλυση της ομοιότητας των δύο καταγραφών αλλά και στο βέλτιστο μονοπάτι και στο ελάχιστο κόστος.

Η δεύτερη γραφική παράσταση στο μεγαλύτερο κομμάτι της αποτελείται από ένα ευθύγραμμο τμήμα το οποίο είναι σχεδόν παράλληλο με τον άξονα x. Το γεγονός αυτό σημαίνει ότι κατά την επίθεση τα πακέτα στέλνονται με πολύ μικρή χρονική διαφορά μεταξύ τους.

Επιπλέον βλέπουμε ότι η δεύτερη γραφική παράσταση είναι αρκετά όμοια με το πρώτο κομμάτι της πρώτης γραφικής παράστασης, δηλαδή με το κομμάτι όπου $x < 500$. Αυτό όμως δεν μπορεί να οδηγήσει σε κάποιο συμπέρασμα. Για τον λόγο αυτό είναι σημαντικό να εξεταστούν το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



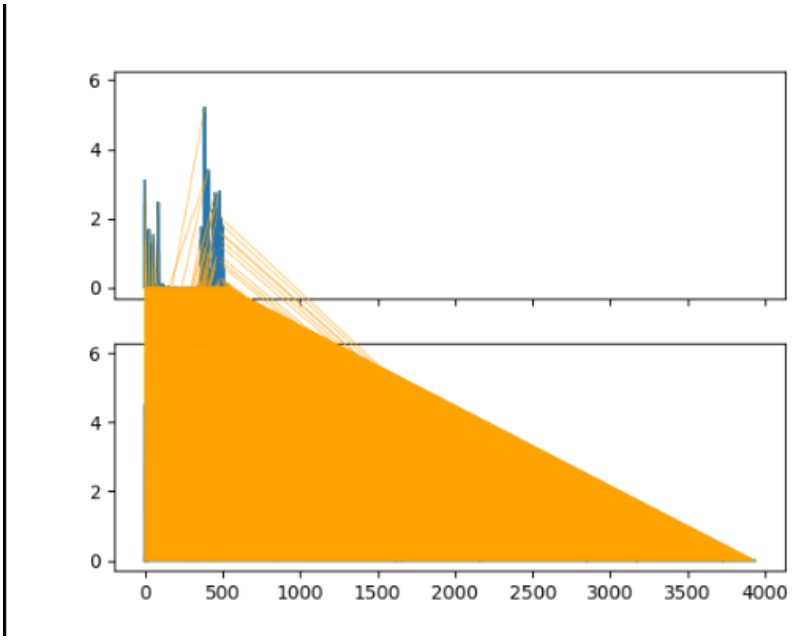
Εικόνα 66. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Time

Στην παραπάνω εικόνα φαίνεται ότι το βέλτιστο μονοπάτι αποτελείται κυρίως από οριζόντιες και κάθετες κινήσεις. Αυτό σημαίνει ότι στο μεγαλύτερο κομμάτι του βέλτιστου μονοπατιού οι δύο καταγραφές δεν παρουσιάζουν ομοιότητα. Επίσης το ελάχιστο κόστος είναι ίσο περίπου με 4986.3 , τιμή που απέχει αρκετά από το μηδέν.

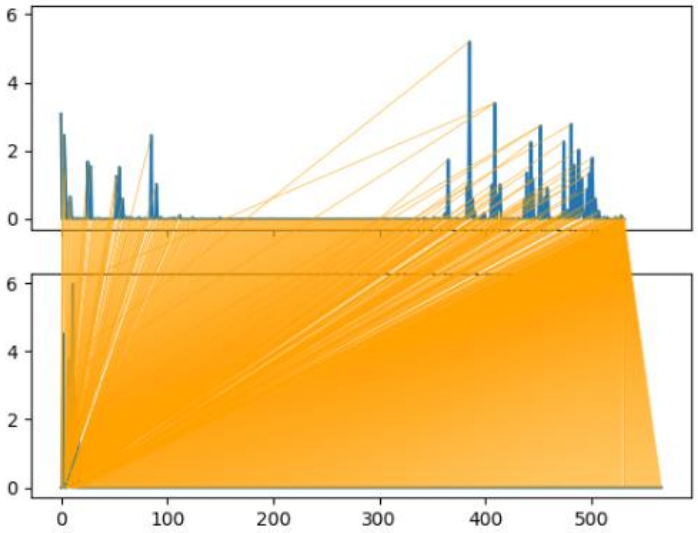
Επομένως οι δύο καταγραφές παρουσιάζουν μικρή ομοιότητα με βάση την παράμετρο Time. Ωστόσο η ανάλυση με βάση την παράμετρο Time δεν είναι πάντα έγκυρη και άρα είναι αναγκαίο να γίνει ανάλυση των καταγραφών με βάση τις παραμέτρους Interval και Size.

4.5.4. Ανάλυση των καταγραφών με βάση την παράμετρο Interval

Στην συνέχεια έγινε ανάλυση των καταγραφών με βάση την παράμετρο Interval. Παρακάτω φαίνεται η ομοιότητα των καταγραφών με βάση την παράμετρο αυτή:



Εικόνα 67. Πρώτη ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Interval



Εικόνα 68. Δεύτερη ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Interval

Παραπάνω έχουν δύο εικόνες για την ομοιότητα των καταγραφών. Και οι δύο εικόνες αφορούν τις ίδιες κινήσεις. Ωστόσο στην πρώτη εικόνα η δεύτερη γραφική παράσταση , η οποία αντιπροσωπεύει την επίθεση tcp syn flood , αποτελείται από όλα τα πακέτα που καταγράφηκαν για την επίθεση. Λόγω όμως της μεγάλης διαφοράς του αριθμού των πακέτων μεταξύ των δύο κινήσεων η πρώτη εικόνα δεν είναι αρκετά ξεκάθαρη. Για τον λόγο αυτό στην δεύτερη εικόνα μειώθηκε ο αριθμός των πακέτων της δεύτερης γραφικής παράστασης οι γραμμές που αντιστοιχίζουν τις δύο γραφικές παραστάσεις να είναι πιο κατανοητές.

Συνεπώς από την δεύτερη εικόνα φαίνεται ότι το μεγαλύτερο κομμάτι της δεύτερης γραφικής παράστασης (x > 50) αντιστοιχίζεται με το τελευταίο κομμάτι της πρώτης γραφικής παράστασης (x > 500). Το κομμάτι της δεύτερης γραφικής παράστασης για x > 50 αποτελεί το κομμάτι στο οποίο στάλθηκαν τα TCP SYN πακέτα κατά την διάρκεια της επίθεσης καθώς είναι σχεδόν παράλληλο με τον άξονα x και άρα η χρονική διαφορά μεταξύ των πακέτων πλησιάζει το μηδέν.

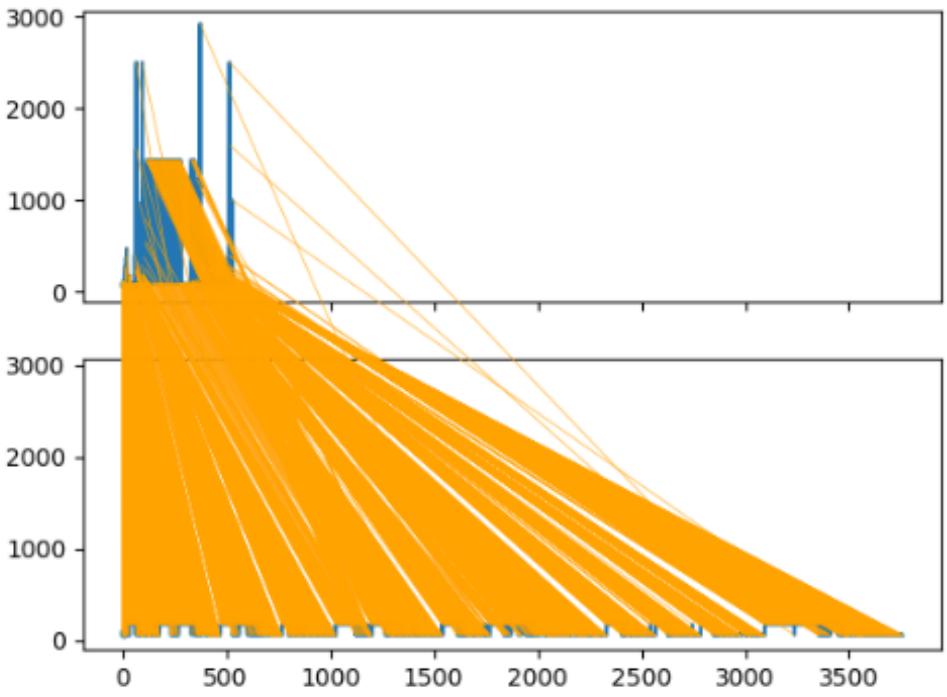
Στην συνέχεια εξετάζεται το βέλτιστο μονοπάτι και το ελάχιστο κόστος:

Εικόνα 69. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Interval

Βλέπουμε ότι το βέλτιστο μονοπάτι δεν αποτελείται καθόλου από διαγώνιες κινήσεις και άρα σε κανένα μέρος του βέλτιστου μονοπατιού οι δύο καταγραφές δεν είναι όμοιες. Το ελάχιστο κόστος όμως είναι ίσο περίπου με 60.65 το οποίο δεν απέχει αρκετά από το μηδέν. Ωστόσο σε συνδυασμό με το βέλτιστο μονοπάτι και την ομοιότητα μπορούμε να καταλήξουμε στο συμπέρασμα ότι οι δύο καταγραφές διαφέρουν.

4.5.5. Ανάλυση των καταγραφών με βάση την παράμετρο Size

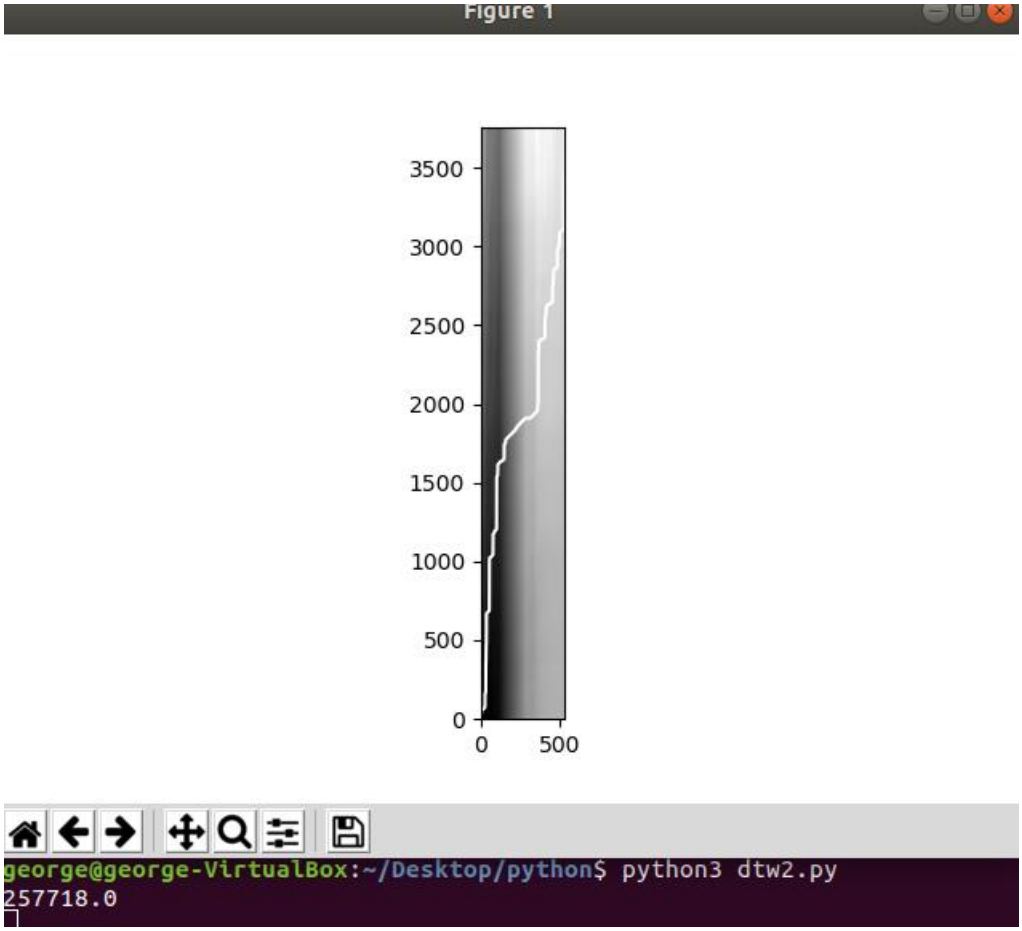
Τέλος εξετάζεται οι ομοιότητα των καταγραφών με βάση την παράμετρο Size:



Εικόνα 70. Ομοιότητα των καταγραφών κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Size

Βλέπουμε ότι η δεύτερη γραφική παράσταση είναι όμοια με σημεία της πρώτης γραφικής παράστασης τα οποία δεν αποτελούν peaks. Επίσης φαίνεται ότι οι κορυφές (peaks) της δεύτερης γραφικής παράστασης έχουν πολύ μικρότερη τιμή από αυτή των κορυφών (peaks) της πρώτης γραφικής παράστασης. Τέλος στην δεύτερη γραφική παράσταση οι κορυφές έχουν όλες το ίδιο ύψος και εμφανίζονται ανά τακτά χρονικά διαστήματα (500 , 1000, 1500 κ.ο.κ.), σε αντίθεση με την πρώτη γραφική παράσταση όπου οι κορυφές δεν έχουν το ίδιο ύψος και εμφανίζονται τυχαία στον χρόνο.

Οι παραπάνω διαφορές δεν είναι αρκετές για να μας οδηγήσουν σε κάποιο συμπέρασμα. Επομένως πρέπει να εξετάσουμε το βέλτιστο μονοπάτι και το ελάχιστο κόστος:



Εικόνα 71. Βέλτιστο μονοπάτι και ελάχιστο κόστος κατά την ανάλυση της επίθεσης TCP SYN flood με βάση την παράμετρο Size

Γίνεται αντιληπτό ότι το βέλτιστο μονοπάτι αποτελείται από ορισμένες μικρές διαγώνιες κινήσεις. Ωστόσο στο μεγαλύτερο μέρος του αποτελείται από κάθετες κινήσεις που σημαίνει ότι στο μεγαλύτερο μέρος του οι δύο καταγραφές δεν είναι όμοιες. Στην συνέχεια βλέπουμε ότι το ελάχιστο κόστος ισούται με 257718, αριθμός που απέχει πολύ από το μηδέν. Επομένως οι δύο καταγραφές δεν παρουσιάζουν αρκετές ομοιότητες με βάση την παράμετρο Size.

4.5.6. Συμπέρασμα των αναλύσεων

Με βάση τις προηγούμενες αναλύσεις μπορούμε να συμπεράνουμε ότι οι δύο καταγραφές δεν μπορούν να θεωρηθούν όμοιες για καμία από τις τρεις παραμέτρους Time, Interval ή Size. Επομένως ένα σύστημα θα μπορούσε να θεωρήσει μία κίνηση σαν αυτή που προσομοίωσε στη επίθεση TCP SYN flood ως πιθανή απειλή.

Κεφάλαιο 5. Επίλογος

5.1. Σύνοψη

Η παρούσα διπλωματική εργασία επιχείρησε να προσομοιώσει ορισμένες από τις πιο γνωστές κυβερνο-επιθέσεις και να τις συγκρίνει με καταστάσεις κατά τις οποίες δεν πραγματοποιούνται επιθέσεις προκειμένου να εξετάσει την δυνατότητα ενός συστήματος να διακρίνει μία επίθεση.

5.2. Μελλοντικοί Στόχοι

Η ανάλυση της παρούσας διπλωματικής εργασίας κρίνεται σκόπιμο να χρησιμοποιηθεί σε πραγματικά συστήματα προκειμένου να ελεγχθεί η ικανότητα εντοπισμού των επιθέσεων από τα υφιστάμενα συστήματα ασφαλείας. Βασικό στόχο αποτελεί η ανάδειξη του βαθμού ετοιμότητας των συστημάτων ασφάλειας σε επιθέσεις παρόμοιου τύπου.

Βιβλιογραφία

Cloudflare(n.d.). What is the Internet Protocol. Ανάκτηση από: https://www.cloudflare.com/learning/network-layer/internet-protocol/

Guru 99(n.d.). TCP/IP Model: What is TCP IP Stack? Protocol Layers, Advantages. Ανάκτηση από: https://www.guru99.com/tcp-ip-model.html

Steven Li (01/08/2017). How Does The Internet Work? Ανάκτηση από την ιστοσελίδα Medium: https://medium.com/@User3141592/how-does-the-internet-work-edc2e22e7eb8

Michael Mullins CCNA (02/07/2001). Exploring the anatomy of a data packet. Ανάκτηση από TechRepublic: https://www.techrepublic.com/article/exploring-the-anatomy-of-a-data-packet/

Sean Wilkins (29/02/2012). Anatomy of an IPv4 Packet. Ανάκτηση από PEARSON IT CERTIFICATION: https://www.pearsonitcertification.com/articles/article.aspx?p=1843887

LearnTomato (09/052014). What is a Client? What is a Server? And What is a Host?. Ανάκτηση από: https://learntomato.flashrouters.com/what-is-a-client-what-is-a-server-what-is-a-host/

Ian Muscat (15/04/2019). What is Code Injection? Ανάκτηση από την ιστοσελίδα acunetix: https://www.acunetix.com/blog/articles/code-injection/

Veracode (n.d.). CRLF INJECTION TUTORIAL: LEARN ABOUT CRLF INJECTION VULNERABILITIES AND PREVENTION. Ανάκτηση από: https://www.veracode.com/security/crlf-injection

Netsparker Security Team (23/05/2019). CRLF Injection and HTTP Response Splitting Vulnerability. Ανάκτηση από την ιστοσελίδα netsparker: https://www.netsparker.com/blog/web-security/crlf-http-header/

Acunetix (n.d.). Cross-site Scripting(XSS). Ανάκτηση από: https://www.acunetix.com/websitesecurity/cross-site-scripting/

Ian Muscat (27/06/2019). What Are Email Injection Attacks. Ανάκτηση από την ιστοσελίδα acunetix: https://www.acunetix.com/blog/articles/email-header-injection/

Ian Muscat (25/04/2017). What is a Host Header Attack? Ανάκτηση από την ιστοσελίδα acunetix: https://www.acunetix.com/blog/articles/automated-detection-of-host-header-attacks/

Ram Kumar (12/04/2018). HOST HEADER INJECTION ATTACK. Ανάκτηση από την ιστοσελίδα Medium: https://medium.com/@rockerramg94/host-header-injection-attack-6cf4ffeb5a03

Akash Sharan (Last Updated: 12-07-2019). LDAP and LDAP Injection/Prevention. Ανάκτηση από GeeksforGeeks: https://www.geeksforgeeks.org/ldap-ldap-injectionprevention/

PortSwigger (n.d.). OS command injection. Ανάκτηση από: https://portswigger.net/web-security/os-command-injection

Tomasz Andrzej Nidecki (01/07/2019). What Is OS Command Injection. Ανάκτηση από την ιστοσελίδα acunetix: https://www.acunetix.com/blog/web-security-zone/os-command-injection/

Acunetix (n.d.) . What is SQL Injection (SQLi) and How to Prevent It. Ανάκτηση από: https://www.acunetix.com/websitesecurity/sql-injection/

W3schools.com (n.d.). XML and XPath. Ανάκτηση από: https://www.w3schools.com/xml/xml_xpath.asp

Ian Muscat (18/04/2019). What Are Injection Attacks. Ανάκτηση από την ιστοσελίδα acunetix: https://www.acunetix.com/blog/articles/injection-attacks/

Michael Bose (16/07/2019). VirtualBox Network Settings: Complete Guide. Ανάκτηση από την ιστοσελίδα Nakivo: https://www.nakivo.com/blog/virtualbox-network-setting-guide/

Jeremy Zhang (n.d.). Dynamic Time Warping. Ανάκτηση από την ιστοσελίδα towards data science: https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd

Shachia Kyaagba (07/09/2018). Dynamic Time Warping with Time Series. Ανάκτηση από την ιστοσελίδα Medium: https://medium.com/@shachiakyaagba_41915/dynamic-time-warping-with-time-series-1f5c05fb8950

RIP Tutorial (n.d.). Introduction To Dynamic Time Warping. Ανάκτηση από: https://riptutorial.com/algorithm/example/24981/introduction-to-dynamic-time-warping

Cloudflare(n.d.). Slowloris DDos Attack. Ανάκτηση από: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/

KALI TOOLS(n.d.). SlowHTTPTest Package Description. Ανάκτηση από: https://tools.kali.org/stress-testing/slowhttptest

Stackoverflow. TCP RST packet details. Ανάκτηση από: https://stackoverflow.com/questions/7735618/tcp-rst-packet-details

Imperva(n.d.). TCP SYN Flood. Ανάκτηση από: https://www.imperva.com/learn/ddos/syn-flood/

Bojana Dobran (21/02/2019). 17 Types of Cyber Attacks To Secure Your Company From in 2020. Ανάκτηση από την ιστοσελίδα phoenixNAP: https://phoenixnap.com/blog/cyber-security-attack-types

Bojana Dobran (11/01/2019). Preventing a Phising Attack: How to Identify Types of Phising. Ανάκτηση από την ιστοσελίδα phoenixNAP*:* https://phoenixnap.com/blog/what-phishing-attack-how-to-identify-protect

Nena Giandomenico (06/10/2020). What is Spear-phising? Defining and Differentiating Spear-phising from Phising. Ανάκτηση από την ιστοσελίδα Digital Guardian: https://digitalguardian.com/blog/what-is-spear-phishing-defining-and-differentiating-spear-phishing-and-phishing

Nena Giandomenico (27/07/2017). WHAT IS A WHALING ATTACK? DEFINING AND IDENTIFYING WHALING ATTACKS. Ανάκτηση από την ιστοσελίδα Digital Guardian: https://digitalguardian.com/blog/what-whaling-attack-defining-and-identifying-whaling-attacks

Comodo Antivirus(n.d.). What is a Malware Attack? | Different Types of Malware Attacks. Ανάκτηση από: https://antivirus.comodo.com/security/malware-attack.php

Imperva(n.d.). DDos Attack Types & Mitigation Methods. Ανάκτηση από: https://www.imperva.com/learn/ddos/ddos-attacks/