T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN PAKET ANALİZİ İLE DDoS ATAKLARININ TESPİTİ VE DLP UYGULAMASI

YÜKSEK LİSANS TEZİ

Erman ÖZER

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Doç. Dr. Celal ÇEKEN**

Temmuz 2015

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN PAKET ANALİZİ İLE DDoS ATAKLARININ TESPİTİ VE DLP UYGULAMASI

YÜKSEK LİSANS TEZİ

Erman ÖZER

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 06.07.2015 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Doç.Dr. Celal ÇEKEN	Doç.Dr. İBRAHİM ÖZÇELİK	Doç.Dr. Mehmet YILDIRIM
Jüri Başkanı	Üye	Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Erman ÖZER
06.07.2015

TEŞEKKÜR

Projenin hazırlanması aşaması boyunca ilminden faydalandığım, ayrıca tecrübelerinden faydalanırken göstermiş olduğu ilgiden dolayı değerli hocam Doç. Dr. Celal ÇEKEN'e, bugünlere gelmemde büyük pay sahibi olan canım aileme ve yardımlarını esirgemeyen tüm mesai arkadaşlarıma teşekkürü bir borç bilirim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ACK	: ACKnowledgement
DHCP	: Dynamic Host Configuration Protocol
DoS	: Denial-of-service
DDoS	: Distributed Denial of Service
FIN	: Finish
FTP	: File Transfer Protocol
HIPS	: Host-based intrusion prevention system
HTTP	: Hyper-Text Transfer Protocol
HTTPS	: Secure Hypertext Transfer Protocol
IDS	: Intrusion Detection System
IP	: Internet Protocol
IPS	: Intrusion Prevention System
ISO	: International Organization for Standardization
NBA	: Network Behavior Analysis
NIPS	: Network-based Intrusion Prevention
OSI	: Open Systems Interconnection
POP	: Post Office Protokol
PSH	: Push
RST	: Reset
SMTP	: Simple Mail Transfer Protocol
SNMP	: Simple Network Management Protocol
SYN	: Synchronize
TCP	: Transmission Control Protocol

TFTP	: Trivial File Transfer Protocol
UDP	: User Datagram Protocol
URG	: Urgent
WIPS	: Wireless Intrusion Prevention System

ŞEKİLLER LİSTESİ

ÖZET

Anahtar kelimeler: Siber güvenlik, Derin paket analizi, DDoS saldırısı, Veri ihlali

Günümüz bilişim teknolojilerinin en büyük sorunlarından biri siber saldırılardır. Bu tezde derin paket analizi kullanılarak DDoS saldırısı olup olmadığı ve veri mahremiyeti ihlali tespit edilmiştir. Ayrıca yakalanan paketlerin grafiği çizilmiştir.

Öncelikle ağ trafiği dinlenerek gelen paketler yakalanmıştır. Paketlerin türüne ve sayısına göre filtreleme işlemi yapılmıştır. Bu paketler veritabanına kaydedilerek analiz edildi, anlık değerler ve ortalama değerler bilinen saldırı desenleriyle karşılaştırıldı ve bir DDoS saldırı girişimi olup olmadığı tespit edilmiştir.

Kaydedilen paketlerin toplam değeri, anlık değerleri ve ortalama değeri güncel olarak elde edilmiştir. Paketler saniye baz alınarak grafiğe dökülmüştür.

TCP türünde olan paketlerin payload kısmı incelenmiştir. Veri ihlali olup olmadığı tespit edilmiştir.

TOWARDS DDoS ATTACK DETECTION USING WITH PACKET ANALYSIS AND DLP APPLICATION

SUMMARY

Keywords: Cyber security, Deep packet analysis, DDoS attack, Data violation

One of the biggest problems of today's informatics technology is cyber attacks. In this thesis whether DDoS attacks and data privacy violation were determined by deep packet inspection. In addition to graph of packets which are captured was drawn.

Initially packets are captured by listening of network traffic. Filtering is performed in accordance with the type and the number of packets. These packets are recorded to database to be analyzed, instant values and average values are compared by known attack patterns and will be determined whether a DDoS attack attempts.

The total value, instant values and average values of the saved packet is obtained to date. Packets were plotted on the basis of seconds.

Payload of packets which are TCP type were examined. Data violation were identified.

BÖLÜM 1. GİRİŞ

Bilişim ve iletişim teknolojisinin oluşturduğu, adına siber dünya dediğimiz bu dünyada mesafeler kalkmış, ülkeler arasında var olan sınırlar yok olmuş, bunun neticesi olarak da dünkü dünyadan farklı bir dünya meydana gelmiştir. Bu dünya çok yenidir ve milyonlarca yıla ihtiyaç duyulmadan, son derece kısa bir zamanda oluşmuştur. Bu dünyanın bir üyesi olmak ve burada yaşamak için nüfus kağıdı gibi bir belgeye ülkeler arasında dolaşmak için de pasaporta gerek yoktur. Bir bilgisayar ve modeme sahip olmak yetmektedir. Bunlara sahip olduktan ve internete bağlandıktan sonra yeni dünyanın yeni bir ferdi olarak bu dünyaya adım atabilmekte internetin yarattığı geniş olanaklardan faydalanabilmekte sınırları aşarak hayal dahi edilemeyecek alanlarda dolaşma, görme ve bunlardan bilgi edinebilme şansına kavuşmaktadır. Her geçen gün bu dünyaya katılanların sayısı artmakta ve bu artış da doğal olarak da bazı sorunları beraberinde getirmektedir. Bu dünyada iyiler kadar kötüler de yer almaktadır, bu değişmez bir olgudur, her bulunan şey başlangıçta insanın yararı için düşünülmekte ve tasarlanmakta sonra da insanın kullanımına sunulmaktadır. Ama kötüler de iyilerin sahip olacağı imkânlardan faydalanarak gecikmeksizin kendi kötü amaçları için kullanma yoluna gitmektedirler. Siber dünyada da bunun gerçek dünyadan bir farkı yoktur. Kötüler bunu kötü emelleri iyiler ise insanlık için kullanma uğraşısı içerisindedirler. Bunun neticesi olarak bu siber dünyada iyiler ve kötüler arasında bitmek ve tükenmek bilmeyen bir çatışma olması doğaldır ve bu, günümüzde de dolu dolu yaşanmaktadır. Yani dünya gerçek ya da siber olsa da değişen bir şey olmamakta, değişen yalnızca ortam olmaktadır. Kötüler, mevcut imkânları kullanarak, bilgisayarlara ve bilgi sistemlerine saldırılar yapan saldırganlar, organize suç örgütü ve teröristler olarak tanımlanabilir [1].

İnternet kullanımı ve teknolojinin hızla geliştiği günümüzde daha etkin ve güçlü savunma sistemlerinin inşa edilmesi, acil durum hazırlığı ve bunların kullanım

süreçlerinin oluşturulması çok önemlidir. Saldırıların, gerçekleştiği ilk anda tespit edilmesi, sanal ya da fiziksel bariyer inşa edilmesi, ulusal ve bölgesel siber güvenlik politikalarının geliştirilmesi siber güvenliğin sağlanması için zorunlu hale gelmiştir.

Siber güvenlik göreceli olarak yeni bir alan olmasına rağmen konu üzerinde çok sayıda çalışma yapılmaktadır. Yapılan çalışmaların sadece başlıklarına bile bakıldığında bunların büyük kısmının teknoloji temelli olduğu ve sistem ve ağ yöneticilerine hitap ettiği görülmektedir [2].

Dünyadaki gelişime ayak uydurma insanlığa daha iyi hizmet verebilme ve refahı artırma çabaları içerisinde; elektrik üretim santralleri; iletişim sistemleri, seyri sefer sistemleri, ulaştırma, barajlar gibi alt yapı sistemleri ile fabrikalar, bankacılık ile finans kuruluşları gibi önemli ekonomik sektörler kuruldu. Kamu ve özel sektör dahi tüm bu tesis ve işletmeler modernizasyon çalışmaları içerisinde bilgisayar sistemleri ile donatılarak bilgisayara bağımlı hale getirdiler. Daha sonra da bunların çoğu gerekli şekilde birbirleri ile bağlantılı çalışmaya başladılar. Bu işin başlangıcında bunun ne kadar önemli bir hassasiyet yaratabileceği anlaşılmadı. Bilgisayarın sağladığı geniş imkan ve kabiliyetler herkesin başını döndürdü, verim ve kalite arttı, bunun sonunda maliyet azaldı, büyük çaptaki bilgi işlemleri depolama ve gönderme olanakları çoğaldı, buna internetin sağladığı geniş olanaklar ve küreselleşme ile gelen kolaylıklar da eklenince kamu ve özel sektör alanında baş döndürücü bir gelişme kendini gösterdi. Her şey insan için düşünüldü ve tesis edildi. Bu güzel tablo ilk defa bilgisayar ve sistemlere yapılan saldırılar ile karşı karşıya kalıncaya kadar devam etti. Bu saldırıları kimlerin yapabileceği ve saldırıların ne kadar büyük zarar verebileceklerin anlaşılması da uzun zaman almadı. Siber iyiler kadar siber kötüler siber dünyada yerlerini aldılar ve önemli bir rol oynamaya başladılar. Kamu ve özel sektörde gerekli ve yeterli güvenlik önlemleri alınmadan tesis edilen sistemlerin ciddi problemlere neden olabileceği kısa zamanda görülmeye ve sıkıntıları hissedilmeye başlandı. Güvenlik tesis için gerekli mali kaynak ayırmaktan kaçınan kuruluşlar bunun bedelini çok pahalı olma durumu ile karşı karşıya kaldılar. Kötülerin iyi organize olmaları; kendilerini güçlendirmenin verdiği avantajların yanında teknolojinin

verdiği olanakların ve ayrıca kuruluşların güvenirliği ikinci plana atmasından faydalanarak yaptıkları saldırılar ile kamu ve özel sektöre verdiği zararların boyutları her geçen gün arttırdılar [1].

Siber ortamda karşılaşılabilecek ve teknik detayları başka çalışmaların konusu olabilecek başlıca saldırı araç ve yöntemleri arasında virüsler, Truva atları, kurtçuklar (worms), zombie ve botnetler, istem dışı elektronik postalar (spam), klavye işlemlerini kaydeden programlar (key loggers), casus yazılımlar (spyware), servis dışı bırakma (DoS), aldatma (IP spoofing), şebeke trafiğinin dinlenmesi (sniffers), yemlemeler (phishing) ve propaganda sayılabilir [2].

Saldırılarla, akıl almaz neticeleri başarabilmek az bir yatırım maliyetini gerektirir. Gelişen teknolojiyi çok iyi kullanan ve çok iyi teşkilatlanan bu saldırganların en büyük avantajlarından birisi de kullandıkları teçhizatın her yerde bulunması, fiyatının ucuz olması ve kolayca satın alınabilmesidir. Gerektiğinde, bir bilgisayar ekibinin kiralanması bir savaş açmaktan daha ucuza gelebilmektedir. Elde edebileceği bilgiler de açık olarak internette mevcuttur. Geriye kalan niyet, beceri ve biraz da bilgidir. Bu saldırganların çoğunun genç yaşta ve bu işi zevk için yapan kişiler oldukları görülmektedir. Diğer bir avantajları ise, yakalanmalarının çok zor olmasıdır. Hiçbir kayıp vermeden çok az bir riskle karşı tarafa çok büyük hasarlar verebilirler. Bu gibi kişilerin yaptıkları hasarlar göz önüne alınarak, bunları yapanların çok daha yetenekli, devlet destekli, organize suç örgütleri ya da teröristler oldukları düşünüldüğünde verebilecekleri zararların neler olabileceği hiç de küçümsenmemelidir [1].

Saldırı tespit sistemleri, internet üzerindeki veri trafiğinin artması ve internet sayfalarının popüler hale gelmesiyle birlikte kişisel ya da kurum sayfalarına yapılan saldırılardan dolayı ihtiyaç duyulan en önemli servislerden biri haline gelmiştir. Bununla birlikte kurumların tüm dünyaya açık olan mail, dns, ftp ve veritabanı gibi sunucularının saldırılara maruz kalması, saldırı tespit sistemlerini, internet güvenliğinin vazgeçilmez bir parçası haline getirmiştir. Saldırı tespit sistemleri; ağ tabanlı saldırı tespit sistemleri ve sunucu tabanlı saldırı tespit sistemleri olarak karşımıza çıkmaktadır.

Saldırı önleme sistemlerinin temel görevi kötü niyetli aktiviteleri belirlemek, kötü niyetli saldırıları durdurmak ve saldırıların türünü rapor etmektir [3]. Saldırı önleme sistemleri saldırı tespit sistemlerinin uzantısı olarak değerlendirilirler. Aralarındaki temel fark aktif olarak önleme yeteneğine sahip olmasıdır [4][5]. Saldırı önleme sistemleri alarm gönderme, istenmeyen paketleri atma ve ağı sıfırlama gibi işlemleri gerçekleştirir [6].

Saldırı Önleme Sistemleri kendi arasında 4'e ayrılır [3][6]:

1. Ağ Tabanlı Saldırı Önleme (Network-based Intrusion Prevention) (NIPS): Tüm ağı şüpheli durumları tespit etmek için protokol analizi yöntemini kullanarak izler.
2. Kablosuz Saldırı Önleme Sistemleri (Wireless Intrusion Prevention Systems) (WIPS): Kablosuz ağ üzerindeki şüpheli durumları tespit etmek için kablosuz ağ protokolü ile izler.
3. Ağ Davranış Analizi (Network Behavior Analysis) (NBA): Ağ trafiğindeki davranışların analizinde kullanılır.
4. Host tabanlı Saldırı Önleme (Host-based Intrusion Prevention) (HIPS): Ana bilgisayara yönelik saldırıları önlemek için kullanılır.

Derin paket analizi ağ üzerinden geçen paketlerinin veri bölümlerine erişerek analiz yapmayı amaçlayan bir tekniktir. Bu çalışma kapsamında derin paket analizi kullanarak DDoS saldırılarının tespiti ve DLP uygulaması gerçekleştirilmiştir. Bu tez çalışması 6 ayrı bölümden oluşmaktadır. 2. Bölümde OSI modeli ve OSI katmanları ayrıntılı şekilde anlatılmıştır. Taşıma katmanı ve taşıma katmanı protokollerinden detaylı bilgiler verilmiştir.

Tezin 3. bölümünde hizmet engelleme saldırılarından bahsedilmiştir. DoS saldırıları karşı sistemde çalışan servisin durdurulmasını amaç edinir. Web sunucuları artık http servisi veremez, e-posta sunucuları artık posta gönderip alamaz hale gelirler. DDoS yani Distributed Denial of Service (Dağıtık Hizmet Engelleme) saldırıları öncesinde sadece DoS (Denial of Service) olarak ortaya çıkmış, yani tek bir kaynaktan hedefe doğru saldırı yapılması şeklinde ortaya

çıkan bu saldırı türü, zamanla şiddetinin arttırılması için çok sayıda kaynaktan tek hedefe yapılan saldırı şekline dönüşmüştür.

Tezin 4.bölümünde DLP (Data Loss Prevention – Veri Sızıntısı Önleme) ' nin ne olduğu, ne yaptığı ve piyasada kullanılan ürünler hakkında detaylı bilgiler verilmiştir.

Tezin 5.bölümünde uygulama detaylı olarak anlatılmıştır. Uygulamanın adımları tek tek anlatılmış, kütüphanelerin nasıl kullanılacağı ve uygulamanın kodları sunulmuştur.

Tezin 6. bölümünde uygulamanın sonucundan bahsedilmiştir.

BÖLÜM 2. OSI KATMANLARI

OSI referans modeli ISO(International Standards Office) tarafından tanımlanmış ve ağ uygulamasında kullanılan örnek bir modeldir; diğer tüm modeller OSI modeli baz alınarak açıklanır. Ayrıca anahtar, HUB, yönlendirici, geçit yolu gibi ağ cihazlarının fonksiyonu OSI başvuru modeline dayanılarak açıklanır [7,8]. ISO dünya standartlarını belirleyen kurumdur. 1970'lerde ISO, veri iletişim standartlarını belirleyen bir OSI modelini belirledi [9].

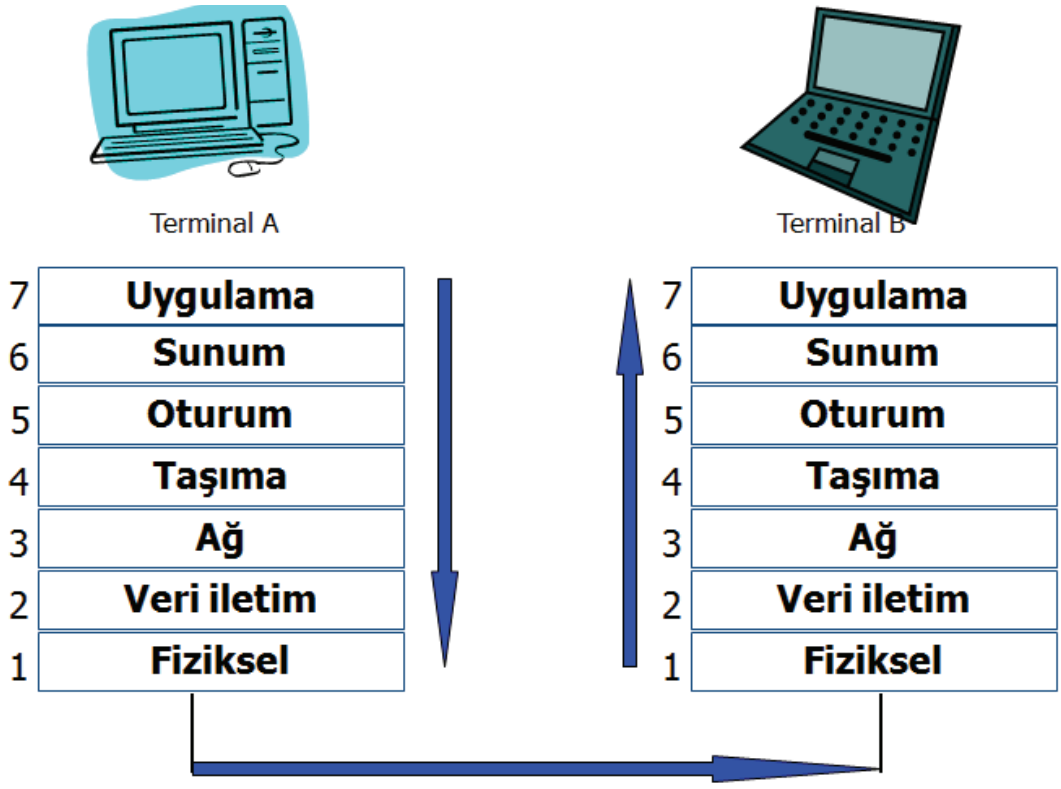
OSI modeli, bilgisayar ağlarında ortak bir dil konuşulmasını sağlamak amacıyla geliştirilmiş katmanlı bir modeldir. OSI modeli katmanlardan oluşur ve bir katman görevini yerine getirmeden diğer katmana geçiş gerçekleşmez. OSI modelinde katmanlar, işlemleri bir sıra dahilinde gerçekleştirir.

Her katman bir üst katmana hizmet sunar ve bir alt katmanın sunduğu hizmetten faydalanır. Aynı şekilde her katman bir alt katmana hizmet sunar ve bir üst katmanın sunduğu hizmetten faydalanır. Yani OSI modelinde işlemler iki taraflı gerçekleşir:

1. Bilgisayar ağı cihazı kullanıcı verisini bilgisayar ağına gönderirken.
2. Bilgisayar ağı cihazı kullanıcı verisini bilgisayar ağından alırken.

OSI sayesinde bilgisayar ağı cihazları arasında ortak bir dil konuşulması sağlanmış olur [7].

OSI modeli 7 katmanlı bir yapıdan meydana gelir:



Şekil 2.1. OSI katmanları

2.1. Fiziksel Katman

Bu katman ağın elektriksel ve mekanik karakteristiklerini belirler. Fiziksel katman standartlarından birisi olan RS-232C aynı zamanda bir kablolama ve sinyalleme standardıdır [8,9].

Fiziksel ortamı oluşturan kablolu ve kablosuz bağlantılar üzerinden aktarılan sinyallerin yapısını anlamak için:

1. Veri aktarımında kullanılan bakır ve fiber optik kablo yapılarını
2. Verilerin iletimini sağlayan sinyal yapılarını
3. Bilgisayar ağı alt yapısını oluşturan mimariyi
4. Veri aktarım ortamında kullanılan bağlantı şekillerini iyi bilmek gerekir [7].

2.2. Veri İletim Katmanı

Fiziksel katmanda kullanıcı verileri elektriksel sinyallere dönüştürülür. Kullanıcının üretmiş olduğu bu verilerin herhangi bir değişime uğramadan fiziksel katmana iletilmesi gerekir. Bu iletim sırasındaki basamaklardan birisi OSI 2. Katman olan Veri İletim Katmanıdır. Fiziksel adresleme, ağ topolojisi vb. bu katmanın görevlerindendir [7,9].

2.3. Ağ Katmanı

Veri paketlerinin bir uçtan bir uca ağdaki çeşitli düğümler üzerinden geçirilip ulaşmasını sağlar. Buradaki bilgi bloklarına paket adı verilir. TCP/IP'de IP protokolü bu katmana ait protokoldür [8]. Ağ katmanı üst katmandan gelen bilgilerde herhangi bir değişiklik yapmadan, kendi kontrol bilgisini paketin başına ekler. Bu bilgi kaynak ve hedef tarafından kullanılan IP adresi bilgisini içerir. IP adresi veri paketlerinin bir uçtan bir uca alıcısına ulaşmasını sağlar. Hangi yolun uygun olacağını ve ağ koşulları bu katmanda değerlendirilir [7].

2.4. Taşıma Katmanı

Farklı ağlardaki uygulamaların birbirleriyle haberleşmesi için kullanılır. Uçtan uca akış kontrolü yapılır.

Veri hedef birim tarafından düzgün biçimde alınmamışsa ya verinin yeniden gönderilmesi istenir ya da üst katmanlara bilgi verilir. Üst katmanlar bozulma oranına göre veriyi tekrar düzenleyebilir [7].

Bu katmanın asli görevi bir cihaz üzerinde aynı anda birden fazla uygulamanın ağ üzerinden haberleşmesini sağlamaktır. Adresleme yapısı olarak her uygulama için kaynak ve hedefte taşıma katmanında port numaraları kullanılır. Aynı zamanda kullanılan taşıma katmanı protokolüne bağlı olarak protokole oturum katmanından alınan verinin küçük parçalara ayrılması ve hedefte doğru bir şekilde birleştirilmek üzere bu kesimlere sıra numaralarının verilmesi işlemleri yapılabilir.

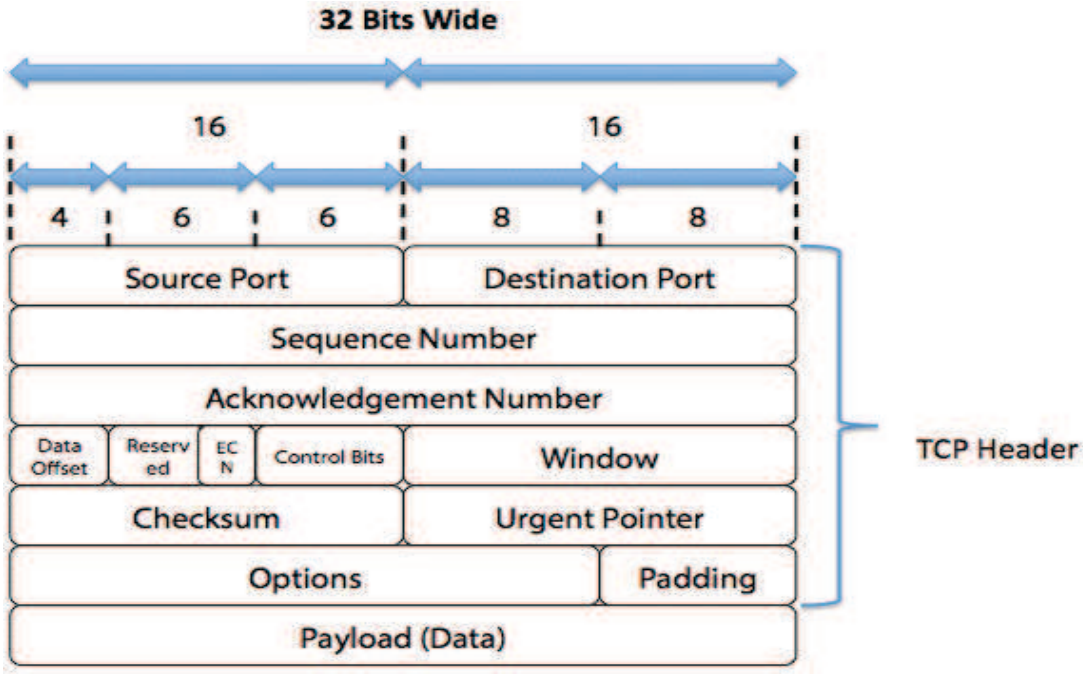
Taşıma katmanı protokolleri bağlantı odaklı ve bağlantısız olmak üzere iki ana grupta incelenebilir. Bağlantı odaklı protokoller verilerin güvenli bir şekilde ve doğru sıra ile hedefe ulaştırılıp ulaştırılmasını sağlayabilir. Herhangi bir sorun çıkması durumunda bu hataların tespit edilip düzeltilmesi de bağlantı odaklı protokollerin sorumluluğundadır. En yaygın kullanılan bağlantı odaklı taşıma katmanı protokolü TCP'dir. Bağlantısız taşıma katmanı protokollerinde ise taşınacak verilere en az ek yük getirecek biçimde ve verilerin hızlıca hedefe iletilmesi esas alınır. UDP en yaygın kullanılan bağlantısız taşıma katmanı protokollerindendir [11].

2.4.1. TCP

TCP'nin temel görevleri aşağıdaki gibi sıralanmıştır:

1. Bir üst katmandan gelen verinin uygun uzunlukta parçalara bölünmesi
2. Herbir parçaya sıraya koyulabilmesi adına sıra numarası verilmesi
3. Kaybolan ya da bozuk gelen parçaların tekrar olarak verilmesi [8]
4. Veriyi tamponlayabilir.
5. Veriyi doğrudan gönderebilir.

Tcp başlığının genel yapısı aşağıdaki şekilde gösterilmiştir:



Şekil 2.2. Tcp başlığının genel yapısı

Kaynak Port(Source Port):Veriyi gönderen bilgisayarın kullandığı TCP portudur.

Hedef Port(Destination Port):Hedef bilgisayarın TCP portudur.

Sıra Numarası(Sequence Number):TCP'nin verinin böldüğü her bir segmentine verdiği numaradır.

Onay Numarası(ACK Number):Alınan bir SYN paketine karşılık olan onay mesajı ACK biti ile gönderilir.

Başlık Uzunluğu(Header Length):TCP başlığının uzunluğunu gösterir.

Rezerve Edilmiş(Reserved):İlerde kullanılmak üzere saklı tutulur.

Kod Bitleri ya da Bayraklar(Code Bits or Flags):Segment ile ilgili kontrol bilgilerini taşır.

Pencere(Window):Akış denetimi için kullanılır.

Hata Kontrol Bitleri(Checksum):Segmentin hatalı ulaşıp ulaşmadığını kontrol etmek için kullanılır.

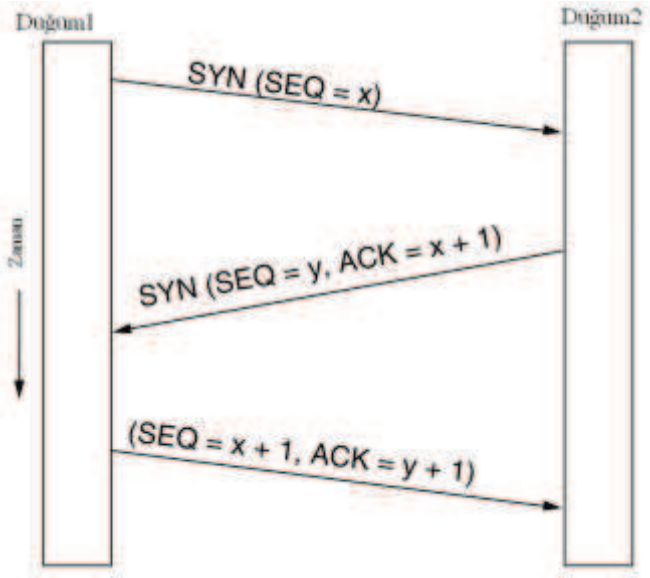
Acil İşaretçisi(Urgent Pointer):Bir verinin acil olarak iletilmek istendiği durumlarda kullanılır.

Seçenek(Options):TCP segmentinin maksimum boyutunun bilgisini taşır.

Veri(Data):Verinin bulunduğu kısım [12].

TCP bayrakları 6 bitten oluşurlar:

1. URG: "1" olması Acil Göstergesi bölgesinin kullanıldığını belirtir.
2. ACK: "1" olması onay alanının geçerli olduğunu gösterir.
3. PSH: Gönderen TCP'nin veriyi hemen göndermesini bildirir. TCP'nin kendi öncelik sırası vardır. PSH bayrağı bu önceliği değiştirir.
4. RST: Sorunlu veya kopmak üzere olan bağlantıları başlangıç durumuna getirmekte kullanılır.
5. SYN: Gönderilen ilk paket ise gönderici ve alıcı tarafından kurulur. Gönderici ve alıcının sanal bağlantı isteğinde bulundukları anlamına gelir.
6. FIN: Son segmentin gönderildiğini bildirir ve bağlantı koparılabilir [10].



Şekil 2.3. TCP bağlantı

TCP'de bağlantılar üç yollu el sıkışma denen şekilde kurulur. Şekil 2.3'te bu yöntemin işleyişi gösterilmiştir. Bu durumda kaynak düğümü SYN bitini 1 yaparak bağlantı kurma isteğini iletir. Bu segmentin alındısını ve SYN değerini 1 olarak göndererek varış düğümü isteği onayladığını bildirir ve kaynaktan veri iletimi başlar. SYN segmenti sıra numaralarından birini kullanır.

2.4.2. UDP

Güvenli bir aktarım yapısına gerek duymayan iletim için kullanılır. İletim için karşı tarafla anlaşma gerekmez. Giden mesajın kontrolü için ACK yapılmaz, hedefe ulaşan paketler üzerinde sıralama yapmaz. UDP minumum protokol kullanımı yaptığı için veri iletiminde başarıyı sağlayacak yapıya sahip olmadığı için veri iletimini garantilemez. Bu sebeplerden ötürü veri iletimi TCP'ye göre oldukça hızlıdır [7].

UDP (User Datagram Protocol – Kullanıcı Datagram Protokolü) TCP ile birlikte aynı taşıma katmanında bulunan, kararlı bir iletim gerektirmeyen uygulamalar için geliştirilmiş basit bir iletim protokolüdür.

UDP, TCP'den daha hızlı çalışmaktadır ve daha hızlı iletim yapmaktadır. TCP her veri paketini gönderdikten sonra alıcıdan onay bekler, paket hatalı iletilmiş mi, bağlantı zayıflamış mı, alıcı doğru paketi mi almış, araya başka bağlantılar mı girmiş, karşı tarafın alabileceğinden daha fazla mı veri gönderilmiş, onay gelmemiş paketleri belirlendi mi, onlar tekrar gönderildi mi gibi birçok olasılığı sürekli kontrol eder. Bu olasılıkların her biri gerçekleştiği anda ayrı ayrı çözümleniyor ve veri iletimi kesinlikle tam olarak gerçekleştirilmiş oluyor [10].

Gelişmiş bilgisayar ağlarında paket anahtarlı bilgisayar iletişiminde bir datagram modu oluşturabilmek için UDP protokolü yazılmıştır. Bu protokol minimum protokol mekanizmasıyla bir uygulama programından diğerine mesaj göndermek için bir prosedür içerir. Bu protokol 'transaction' yönlendirmelidir. Paketin teslim garantisini isteyen uygulamalar TCP protokolünü kullanır.

1. Geniş alan ağlarında (WAN) ses ve görüntü aktarımı gibi gerçek zamanlı veri aktarımlarında UDP kullanılır.
2. UDP bağlantı kurulum işlemlerini,akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini en aza indirir.
3. UDP ve TCP aynı iletişim yolunu kullandıklarında UDP ile yapılan gerçek zamanlı veri transferinin servis kalitesi TCP'nin oluşturduğu yüksek veri trafiği nedeniyle azalır.

UDP'yi kullanan protokollerden bazıları DNS, TFTP, ve SNMP protokolleridir. Uygulama programcıları birçok zaman UDP'yi TCP'ye tercih eder, zira UDP ağ üzerinde fazla bant genişliği kaplamaz.

UDP güvenilir olmayan bir aktarım protokolüdür. Ağ üzerinden paketi gönderir ama gidip gitmediğini takip etmez ve paketin yerine ulaşıp ulaşmayacağına onay verme yetkisi yoktur. UDP üzerinden güvenilir şekilde veri göndermek isteyen bir uygulama bunu kendi yöntemleriyle yapmak zorundadır [13].

Udp başlık yapısı şekil 2.4'te gösterilmiştir.

Şekil 2.4. UDP başlık yapısı

Kaynak Port: Opsiyonel bir alandır, bir bilgi kaybı durumunda ya da başka bir bilgi iletileceğinde mesaj gönderilecek adresin neresi olduğunu, yani göndericinin port numarasını belirtir. Eğer gönderici portu belirtilmez ise bu alan "0" bilgisiyle doldurulur.

Hedef Port: Bir hedef İnternet adresiyle birlikte anlam kazanan port numarasıdır. Bilgi paketlerinin iletileceği hedef adresi belirtir.

Uzunluk: Başlık ve veri alanları dahil UDP paketinin tamamının uzunluğunu belirtir. Bu durumda veri olmasa bile başlık alanları nedeniyle paketin minimum uzunluğu 8 bayt olacaktır.

Checksum: IP başlığı bilgileri ile mantıksal başlığın toplamının 16 bitlik birlere göre tümleyenidir. Bu alanda, UDP başlığı ve veri sonu, (gerekli ise) iki bayta tamamlamak için "0" ile doldurularak karşı tarafa gönderilir. Alıcı kendisine gelen bu 16 bitlik alanı alarak tümleyen aritmetiğinde çözümler. Sonuçta bütün bitler "1" ise hatasız iletilmiştir. Eğer herhangi bir "0" görülürse, hata olduğu anlaşılır. Bu durumda alınan datagram atılır [10].

2.5. Oturum Katmanı

Uç düğümler arasında aşağıdaki işlemleri kapsar:

1. Oturumun kurulması
2. Oturumun yönetilmesi
3. Oturumun sonlandırılması

İletişimin mantıksal sürekliliğini sağlanması için, iletişimin kopması durumunda bir senkronizasyon noktasından başlayarak iletişim kaldığı yerden devam etmesini sağlar [8]. Bunun için bağlantının kesildiği anda gönderilemeyen verilerin belirlenmesi ve bağlantı sağlandığında gönderilemeyen verilerden başlayıp veri iletiminin kaldığı yerden devam etmesi için bir eşleşme noktası belirlenir. Bu tür eşleşme noktaları OSI alt katmanları ile ilişkili çalışarak veri iletiminin devamını sağlar.

Oturum katmanı kullanıcı verisini oluşturan bloğun tamamını doğru olarak elde edip üst katmana teslim eder. Blokla ilgili işin bittiğini gönderdiği birime bildirebilir. Bu işlemlerin sonucunda veriyi gönderen birim kontrol için belleğinde tuttuğu veriyi siler. Fakat birimler arasında iletişim koparsa, iki taraf arasında yeniden bağlantı kurulana kadar veriyi gönderen birim hafızasında bu bilgileri tutmaya devam eder. Bağlantı sağlandığında veriyi gönderen birim belleğinde tuttuğu bilgiyi tekrar gönderir [7].

2.6. Sunum Katmanı

Bu katmanda gelen paketler bilgi haline dönüştürülür. Bilginin karakter set çevirimi veya değiştirilmesi, şifreleme vb. görevleri bu katman üstlenir [9].

Sunum katmanının iki temel görevi vardır:

1. Farklı işletim sistemleri arasında ortak dil kullanımını sağlar: Bunlardan birkaçı aşağıdaki gibidir:
   a. Windows
   b. Unix
   c. Linux
   d. Mac OS
   e. Android
   f. iOS
   g. Pardus

2. Bilgilerin bilgisayar ağı üzerinde çalışan uygulamalar ve ağ biriminin kullandığı formata göre düzenlenmesini sağlar. Veriyi gönderen birim ve alan birim arasında aşağıdaki işlemleri yerine getirir:
   a. Sıkıştırma/Açma
   b. Şifreleme/Şifre çözme
   c. EBCDIC-ASCII dönüşümü ve ters dönüşümü
   d. PICT-.JPG dönüşümü ve ters dönüşümü
   e. WAV-MP3 dönüşümü ve ters dönüşümü [7]

2.7. Uygulama Katmanı

Kullanıcıya en yakın olan katmandır. Kelime işlemci, haber grubu yazılımı, banka terminali programları vs. bu katmanın parçalarıdır [9].

Uygulama katmanında kullanılan protokoller şunlardır:

1. DHCP
2. HTTP
3. HTTPS
4. SMTP
5. TELNET
6. TFTP
7. SNMP
8. FTP
9. POP3

Bilgisayar ağı yazılımcıları uygulama programı arabirimleri ile uygulama katmanı üzerinden işlemleri başlatır, devam ettirir ve sonlandırır [7].

BÖLÜM 3. HİZMET ENGELLEME SALDIRISI(DDoS)

DoS saldırıları karşı sistemde çalışan servisin durdurulmasını amaç edinir. Web sunucuları artık http servisi veremez, e-posta sunucuları artık posta gönderip alamaz hale gelirler.

Bu saldırılar iki şekilde meydana gelir:

1. Aşağıda belirtilen kaynakları tüketerek:
   a) Hafıza
   b) İşlemci
   c) Bant genişliği
2. Serviste bulunan bir zayıflığı kullanarak [14]

DoS saldırılarının belirtileri:

1. Beklenmedik biçimde düşük ağ performansı
2. Web sitelerine erişimde yavaşlık
3. Ağ bağlantılarında kesilmeler
4. Spam e-postaların sayısında artış
5. Bir web sitesinin belli bölümlerine erişimin imkânsız hale gelmesi

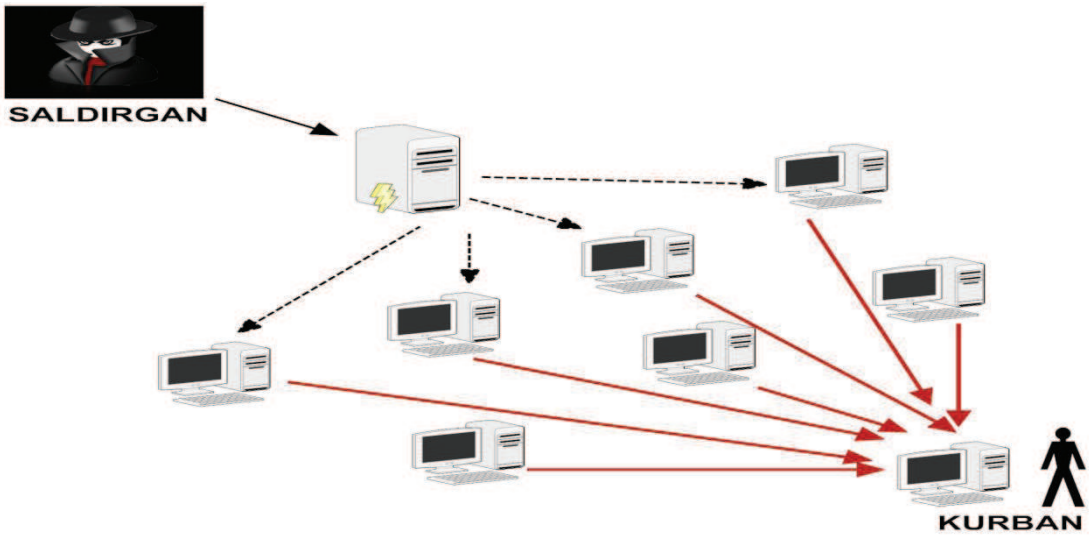
DoS saldırı çeşitleri:

1. TCP/SYN taşma saldırısı
2. TCP tekrar taşma saldırısı
3. Land saldırısı
4. Kaba kuvvet saldırısı
5. Ping taşma saldırısı
6. Gözyaşı saldırısı
7. UDP taşma saldırısı [15]

DDoS yani Distributed Denial of Service (Dağıtık Hizmet Engelleme) saldırıları, tamamen Bilgi güvenliği unsurlarından erişilebilirliği hedef almaktadır. Öncesinde sadece DoS (Denial of Service), yani tek bir kaynaktan hedefe doğru saldırı yapılması şeklinde ortaya çıkan bu saldırı türü, zamanla şiddetinin arttırılması için çok sayıda kaynaktan tek hedefe yapılan saldırı şekline dönüşmüştür [16].

DDoS hakkında bilinen temel özellikler şunlardır:

1. Binlerce, yüzbinlerce sistem kullanılarak gerçekleştirilir.
2. Genellikle sahte IP adresleri kullanılır.
3. BotNet'ler kullanılır.
4. Saldırgan kendini gizler.
5. Engellemesi zordur! [17]



Şekil 3.1. Örnek bir saldırı çeşidi

DDoS saldırı örneklerinden biri Mydoom virüsüdür. Mydoom virüsü hakkında bilinen bazı özellikler şunlardır:

1. En çok bilinen saldırı türü Mydoom virüsüdür [16].
2. En hızlı yayılan virüs türüdür.

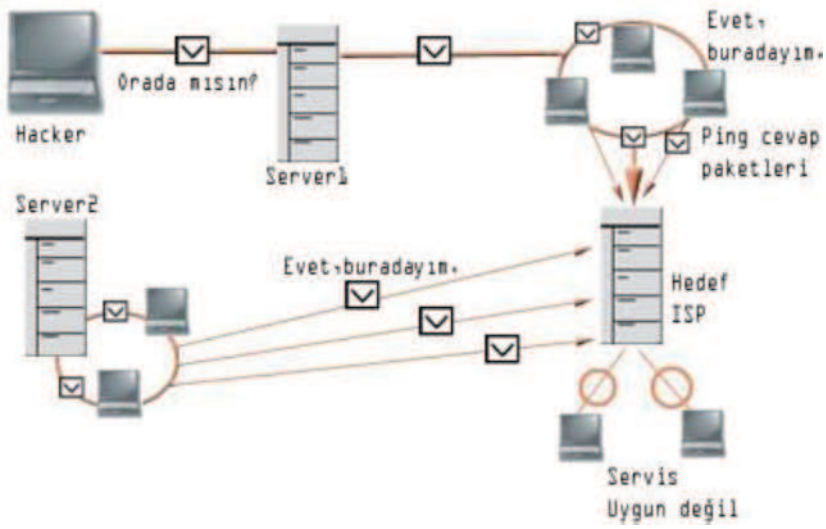
3. Bize gelen her 12 mailden biri bu virüsten etkilendiği tahmin edilmektedir [18].
4. Windows tabanlı sistemlerde etkili olduğu söyleniyor. Virüs outlook'taki adres defterini tarayarak kendi göndereceği yeni adresler buluyor.
5. Kendi e-posta motorunu kullanır.
6. Açığa çıkmamak için kendini antivirüs firmalarına, devlet kuruluşlarına ve askeri kurumlara göndermiyor [19].

Bilinen diğer saldırı örneği Stacheldraht virüsüdür. Stacheldraht virüsü hakkında bilinen bazı özellikler şunlardır:

1. Stacheldraht, DDoS aracının klasik bir örneğidir. İstemci programın kulanıldığı DDoS saldırısını kolaylaştıran katmanlı bir yapıdan faydalanır.
2. Hedeflenen uzak bilgisayarda uzaktan bağlantıların çalıştırılmasını kabul eden programlar, saldırganlar tarafından işleyiciler üzerinden tehlikeye sokulabilir. Her işleyici bin ajanı kontrol edebilir.
3. Stacheldraht'ın yeni sürümü olarak 'Stacheldraht 1.666+antigl+yps' ve 'Stacheldraht 1.666+smurf+yps' olduğu biliniyor [20].
4. Stacheldraht smurf(smurf attack) ve fraggle saldırıları gibi amplifikasyon ve ip spoofing(sahtekarlık) merkezli klasik Dos saldırı metodlarını kullanır [16].

Bir diğer örnek ise Smurf saldırısıdır. Smurf saldırısı ile internet servis sağlayıcılara aşağıdaki adımları izleyerek saldırılır:

1. Öncelikle saldırgan internete bağlı networke ping yollar.
2. Hacker ping paketleri üzerindeki dönüş adresiyle oynar. Kendi adresi yerine, saldıracağı ISP(Internet Servis Sağlayıcılar)'nin adresini kullanır. Bunun iki nedeni vardır:
    a) Hem ISP'ye saldırır.
    b) Hem de ping request üzerinde kendi adresi bulunmadığı için yakalanmaktan kendini korur.
3. Ping istekleri aralıksız bir şekilde devam ettiği için sistem kilitlenir [21].



Şekil 3.2. Smurf atak gösterimi

Saldırı örneklerinden biri de Fraggle saldırılarıdır. Fraggle saldırıları da smurf saldırıları gibi nerdeyse aynıdır. Tek farklılığı kullanıcı protokolünü veya UDP'yi kullanır. Smurf ataklar gibi router'ları durdurmaya çalışır [22]. Smurf ataklara göre daha fazla zarar verir ve sistemdeki trafik yoğunluğunun artmasına neden olur [23].

3.1. DDoS Saldırı Nedenleri

DDoS saldırganları genellikle çeşitli teşviklerle motive edilir. DDoS saldırıları saldırganların motivasyonlarına göre beş ana kategoride sınıflandırılabilir [24]:

1. Finansal /ekonomik kazanç: Bu saldırılarda şirketlerin büyük endişeleri vardır. Bu kategoride en teknik ve en deneyimli saldırganlar vardır. Finansal kazanç sağlamak için başlatılan bu saldırılar genellikle en tehlikeli ve durdurulması zor olan saldırılardır.
2. İntikam : Bu kategoride saldırganlar hüsrana uğramış ve düşük tekniğe sahip kişilerdir. Saldırganlar saldırıları adaletsiz bir biçimde gerçekleştirirler.

3. İdeolojik inanç: Bu kategoride saldırganlar ideolojik inançlar doğrultusunda saldırırlar. Bu kategori günümüzde DDoS saldırılarını başlatmak için en önemli teşviklerden biridir. Örnek olarak 2007 yılında Estonya'da, 2009 yılında İran'da, 2010 yılında Wikileaks'ta siyasi teşvikler sabotaja yol açmıştır.
4. Entelektüel Mücadele: Bu kategoride saldırganlar hedeflenen sistemlere tecrübe kazanmak ve saldırıları nasıl başlatacaklarını öğrenmek için saldırırlar. Genellikle genç saldırganlar yeteneklerini göstermek için saldırırlar.
5. Siber savaş: Bu kategoride saldırganlar o ülkedeki askeri veya terör örgütü mensuplarıdır. Politik nedenlerden dolayı diğer ülkenin kritik bölgelerine saldırırlar. Potansiyel hedefler arasında sivil bölümler, özel veya genel finansal kuruluşlar, ajanslar, telekomünikasyon ve mobil servis sağlayıcıları bulunmaktadır. Siber savaş saldırganları bol kaynaklara sahip ve eğitimli bireylerdir.

3.2. Saldırı Sınıfları

DDoS saldırıları iki ana kategoride sınıflandırılmaktadır [24]:

1. Ağ/Taşıma katmanındaki DDoS flooding atakları: Bu ataklar genellikle TCP, UDP, ICMP ve DNS protokol paketlerini kullanarak başlatılır. Bu kategoride dört tür saldırı türü vardır.
    a) Flooding saldırılar: Saldırganlar kurbanın bant genişliğine zarar vererek saldırırlar.
    b) Protocol istismar flooding saldırılar: Saldırganlar kurbanın kaynaklarını aşırı miktarda tüketerek saldırıda bulunurlar.
    c) Yansıma-bazlı flooding saldırılar: Saldırganlar genellikle doğrudan istekler yerine sahte istekler gönderirler, böylece reflektörler kurbana cevapları yollar ve kurbanın kaynaklarını tüketirler.
    d) Amplifikasyon-bazlı flooding saldırılar: Saldırganlar uzun mesajlar ve birden fazla mesajlar üretir, kurban bu mesajlara cevap verdiği için trafiği artırır ve servisi sömürürler.

2. Uygulama düzeyinde DDoS flooding saldırılar: Bu saldırılar sunucu kaynaklarını tüketerek kullanıcının servisini bozmaya odaklanmıştır.
    a) Yansıma / amplifikasyon bazlı flooding saldırılar: Bu saldırılar ağ / taşıma katmanında kullanılan tekniklerin aynısını kullanırlar. Örnek olarak DNS amplifikasyon saldırıları hem yansıma hem de yükseltme tekniklerini kullanır.
    b) HTTP flooding atakları: Bu kategoride dört tür atak vardır:
        1. Oturum flooding atakları: Bu atak türünde kullanılacak yasal istek sınırından çok daha fazla istek yollanır, böylece sunucu kaynakları tükenir ve DDoS flooding saldırısına yol açar.
        2. İstek flooding atakları: Bu atak türünde saldırganlar normal istek sayısından çok daha fazla istek yollarlar ve sunucuda DDoS flooding saldırısına yol açar. Bu kategoride bilinen en iyi atak tek oturum HTTP get/post flooding ataklarıdır.
        3. Asimetrik ataklar: Bu saldırı türünde saldırganlar yüksek iş yükünde istek yollarlar. Birden fazla HTTP get/post flood ve bozuk uygulama olmak üzere iki türde sınıflandırılır.
        4. Yavaş istek / yanıt ataklar: Bu atak türü saldırı başlık atağı, HTTP parçalanma atağı, yavaş istek atağı ve yavaş okuma atağı olmak üzere dört kategoride sınıflandırılır.

3.3. Botnet Tabanlı DDoS Saldırılar

Botnetler bilgisayar ağları ve uygulamaları üzerinde DDoS saldırıları kolaylaştıran mekanizmalardır. Son zamanlarda DDoS saldırıları botnetler kullanılarak yapılmıştır. Botnetler, istenmeyen e-posta mesajları göndermek, virüsleri yaymak, bilgisayar ve sunuculara saldırmak amacıyla kullanılabilir. Bu işlemler uzaktan yönetilerek de yapılabilir. Botnetler ustalar, işleyiciler ve robotlar olmak üzere üç kısım içerir. Botnetler IRC tabanlı ,web tabanlı ve P2P tabanlı olarak sınıflandırılır. IRC tabanlı ve web tabanlı botnetler en çok kullanılan saldırı sistemleridir. IRC metin tabanlı protokoldür. Saldırganlar IRC sunucusuna bağlanırlar ve dosyalarını kolayca paylaştırır. Web tabanlı botnetler

kendilerini HTTP trafiğinde gizleyerek saldırırlar. Karmaşık PHP komutları robotlar tarafından kontrol edilir ve yapılandırılır [24].

3.4. DDoS Önleme Yöntemleri

DDoS saldırıları çok fazla sayıda kaynakları tüketmeye sebep olur. DDoS defans mekanizması onları en kısa sürede tespit eder ve durdurarak kaynak tüketimini durdurmaya çalışır. [24]

1. Ağ / taşıma seviyesindeki DDoS flooding saldırılarına karşı savunma mekanizması:
    a. Kaynak tabanlı mekanizmalar: Kaynak tabanlı mekanizmalar DDoS saldırılarını önlemek için kaynaklarının yanına yerleştirilir. Kaynak tabanlı mekanizmalar şunlardır: Yönlendiriciye olan giriş çıkışları filtreleme mekanizması, D-WARD, Multops, Mananet'in arka güvenlik duvarı.
    b. Hedef tabanlı mekanizmalar: Hedef tabanlı savunma mekanizmalarda bulma ve yanıt saldırının hedefinde yapılır. En önemli hedef tabanlı mekanizmalardan bazıları şunlardır: IP geri izleme mekanizması, yönetim bilgi tabanı, paket işaretleme ve filtreleme mekanizması, sıkışıklık seviyesine göre paket düşürülmesi.
    c. Ağ tabanlı mekanizmalar: Bu mekanizmalar ağ içerisinde kullanılır. Saldırı trafiğini tespit ederler ve durdurmaya çalışırlar. İki önemli mekanizması bulunmaktadır: Route tabanlı paket filtreleme, kötü niyetli yönlendiricileri belirleme ve filtreleme.
    d. Dağıtık mekanizmalar: Bu mekanizmalar kaynak ve hedef gibi birden fazla yerde konuşlandırılmıştır. Kullanılan bazı mekanizmalar şunlardır: Dağıtık paket işaretleme ve azaltma / filtreleme mekanizması, DEFCOM, COSSACK, yetenek tabanlı mekanizmalar, aktif internet trafiğini filtreleme, StopIt.

DDoS saldırıları ile ilgili incelenen makalelerde yapılan çalışmalar şunlardır:

1. Guang Jin ve arkadaşları Packet Asymmetry Path Marking(PAMP) sistemini kullanarak kötü niyetli flooding trafiğini belirlemişlerdir. Kullandıkları sistem paketlerin asimetrik olarak sergilenmesidir [25].
2. Yen-Hun-Hu ve arkadaşları Window-Based Packet Filtering(WBPF) sistemini kullanmışlardır. Bu sistemde akış oranı normal bant genişliğinden fazla ise sistemde saldırı olduğunu tespit eder. Kuyrukta çok fazla paket olup olmadığına bakarlar [26].
3. Theerasak Thapngam ve arkadaşları paket varışlarını gözlemleyerek, ağ trafiğindeki davranışlarıyla çözüm bulmuşlardır. Bu teknik ile DDoS atak kaynakları ve gerçek kullanıcılar arasında ayırt etme sağlanır. Paket varış oranını kullanarak atak olup olmadığı anlaşılır [27].
4. You-ye Sun ve arkadaşları Deterministic Packet Marking(DPM) sistemini kullanarak DDoS atak olup olmadığını tespit etmişlerdir. Bu metotla çok sayıda eşzamanlı DDoS saldırganlarının izini surer [28].

BÖLÜM 4. DLP SİSTEMLERİ

Data Loss Prevention, bilgi güvenliğinde değerli verilerin başkalarının eline geçmesini engellemek üzere tasarlanmış teknolojiye verilen bir isimdir [29].

Veri sızıntısı önleme çözümleri, kurum için gizli olan verinin, kurum dışına sızmasını önlemek amacıyla geliştirilmiştir. Verinin saklandığı ortamlarda (dosya sunucusu gibi), iletildiği ortamlarda (e-mail, instant messenger, ftp gibi) ve kullanıldığı ortamlarda (kullanıcı bilgisayarları, USB Flash bellekler gibi) saptanması, izlenmesi ve verinin korunması sağlanır. Veriyi tespit etmek ve korumak için çeşitli "deep content analysis" teknikleri kullanılır. DLP çözümleri, ağ üzerinde iletilen gizli bilginin korunması amacıyla geliştirilmiştir [30].

DLP, birbirlerinin yerine de kullanılabilen data loss prevention ya da data leak prevention sözcüklerinin kısaltması olup veri sızıntısı önleme, veri kaybı önleme anlamlarına gelmektedir. Standart DLP firewall, IDS (intrusion detection systems) ve anti-virüs yazılımlarını içerir. Sadece DLP konusuna odaklanmış yazılımlar ise hassas verilerin yetkisiz kullanımı, iletiminin izlenmesini ve korunmasını amaçlar [31].

Bu noktada DLP odaklı yazılımlar, bir takım veri sızıntısını engelleyen tanımlanmış kurallar eşliğinde, ağ trafiğini gözetler, eposta sunucusuyla entegre çalışır ve kullanıcı bilgisayarında kurallar çerçevesinin dışına çıkılmamasını denetler. Kurallar aşağıdaki şekilde sıralanmıştır:

1. İçinde örneğin kredi kartı numarası bulunan epostaların kaydını almak ya da engellemek
2. Dosyaların USB harici disklere ya da CD/DVD ROM'lara taşınmasını engellemek

3. Print screen ya da kopyala/yapıştır tuşlarıyla uygulama içinden veri almanın engellenmesi
4. Belli dosyalara erişim kısıtlanmaları
5. Webmail sitelerine giriş
6. FTP ile dosya yükleme
7. Web tarayıcı ile dosya yükleme (80 ve 443 portu)

Buna rağmen DLP, henüz bilgi güvenliği sektöründe tam bir standarda kavuşmuş bir terim değildir [31].

DLP teknolojisi verinin kaynağı ile ilgilenir. Özel kullanıcılar için ekstra yetkiler tanımlanabilmesine karşın genel olarak bilgi koruma mantığı aynıdır. Hatta bazı ürünlerde güvenilir kullanıcı tanımı bulunmaz.

Merkezi olarak yönetilen DLP sistemi, önceden tanımlanmış kaynakları koruma altına alır. Korunan kaynaklar üzerinde yapılan tüm işlemler raporlanır ve kaynağın güvenlik gözü ile izlenmesine imkân sağlanır.

DLP ürünlerinin tamamı giriş/çıkış aygıtlarının denetimini yapabilir. Değerli verinin uygulamalar üzerinden (email, webmail, IM, P2P, Skype), ağ sistemleri üzerinden (wifi, bluetooth, http(s), ftp), fiziksel aygıtlar üzerinden (yazıcılar, usb hafızalar, faks, cd/dvd) başka sistemlere gönderilmesini engeller.

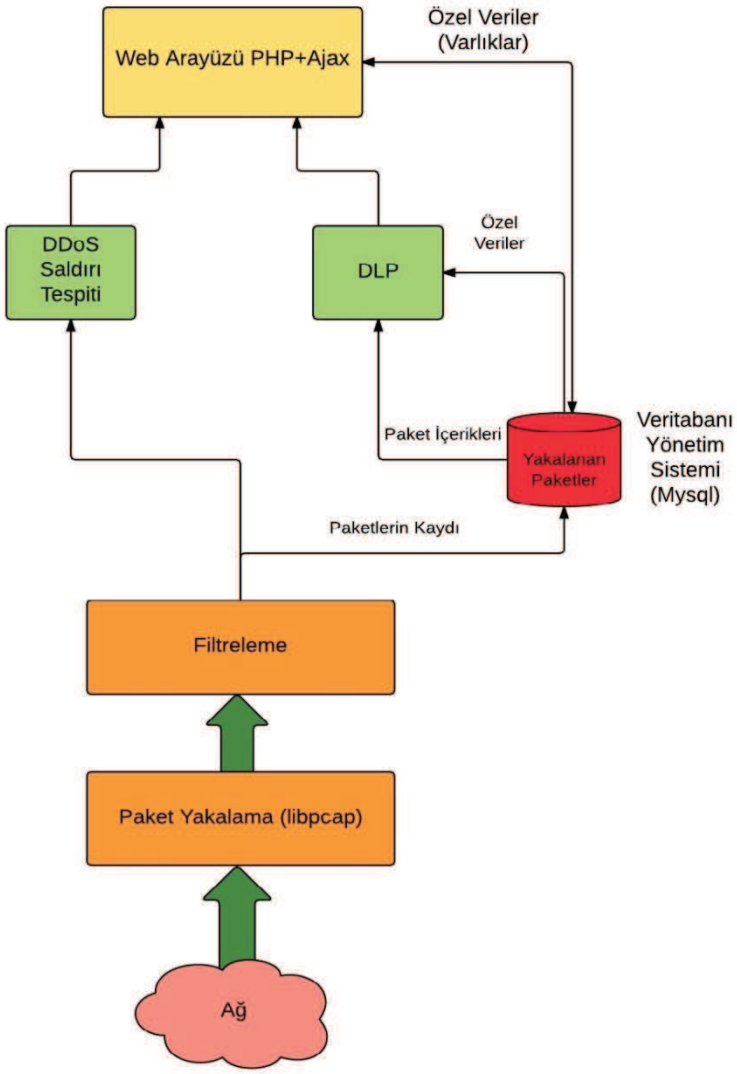
DLP veriyi korumaya yönelik geliştirildiği için hack edilmiş, Truva atı bulaşmış, güvenlik zaafı bulunan veya hatalı konfigüre edilmiş bilgisayarlardan veri çalınmasını engeller. Bazı DLP ürünleri için kullanıcının nerede olduğu önemli değildir. Taşınabilir bilgisayarlar şirket ağında değilken bile agent mimarisi sayesinde koruma kalkanını her zaman çalıştırabilir. DLP ürünleri sunucu sistemlerinde, istemcilerde, taşınabilir bilgisayarlarda, Blackberry gibi avuçiçi bilgisayarlarda çalıştırılabilir. Tüm DLP sistemi dağıtık ortamlarda olmasına rağmen merkezi olarak yönetilebilir ve raporlanabilir. DLP ürünleri değişik biçimlerde bulunan kaynakları koruyabilir. Filtrelenmiş dosyalar, klasörler, bilgi bankaları, sıkıştırılmış dosyalar koruma altına alınabilir [29].

Şu anda piyasada olan DLP ürünlerinden bazıları şunlardır:

1. Cisco Iron Port
2. End Point Protector
3. Websense DLP
4. Trend Micro DLP
5. Check Point DLP Software Blade
6. Digital Guardian
7. McAfee DLP Endpoint
8. Trustware [31]

BÖLÜM 5. SİBER SALDIRI TESPİTİ VE DLP UYGULAMASI

Çalışma kapsamında geliştirilen uygulamanın ana hatları şekil 5.1'de gösterilmiştir:



Şekil 5.1. Uygulama topolojisi

Uygulama 5 ana kısımdan oluşmaktadır:

1. Paketlerin yakalanması
2. Paketlerin filtrelenmesi
3. Yakalanan ve filtrelenen paketlerin veritabanına kaydedilmesi
4. DDoS Uygulaması
5. DLP Uygulaması

5.1. Paketlerin Yakalanması

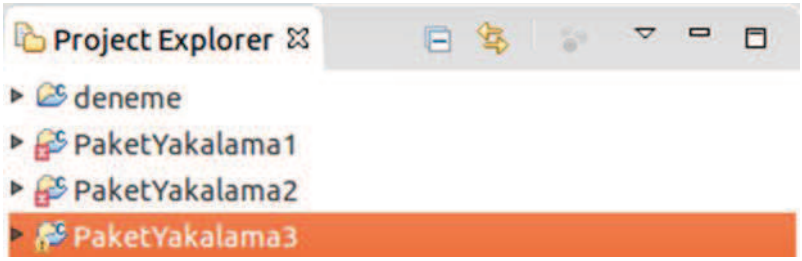
Uygulama ağ trafiği dinlenerek paketlerin yakalanması işlemiyle başlamaktadır. Paketleri dinlemek ve yakalamak için pcap kütüphanesi kullanılmıştır. Uygulama Ubuntu işletim sistemi üzerinde olduğu için, pcap kütüphanesinin Linux versiyonu olan kütüphane (libpcap) yüklenmiştir.

Libpcap kütüphanesini yüklemek için terminalde şu komut yazılması yeterlidir:

sudo apt-get install libpcap

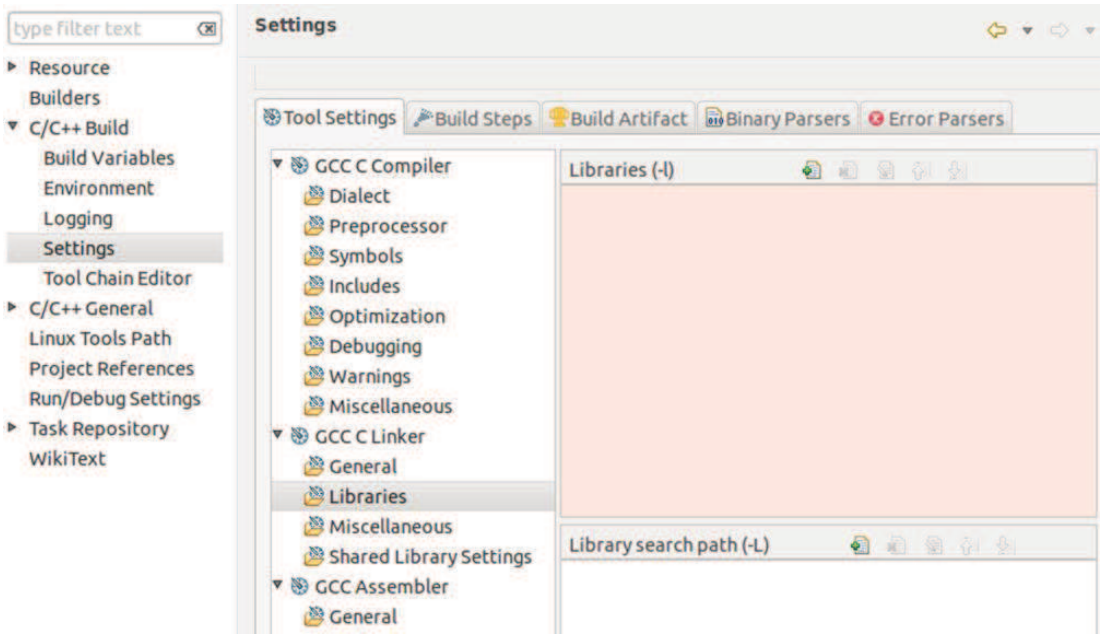
Sudo komutu normal sistem kullanıcıların, çeşitli komutlar üzerinde geçici root kullanıcı haklarına sahip olmalarını sağlar. Böylelikle, hazırlanan veya kullanılan bir program, düzgün çalışabilmesi için kök kullanıcı haklarına ihtiyaç duyuyorsa, ve bu program root kullanıcısı dışındaki kullanıcılar tarafından da kullanılacaksa, söz konusu programın düzgün çalışabilmesi sağlanmış olur.

Pcap kütüphanesini kurduktan sonra sistemimize yüklemek için aşağıdaki işlemler sırasıyla yapılmaktadır. Örnek olarak PaketYakalama3 projemiz elimizde olsun.



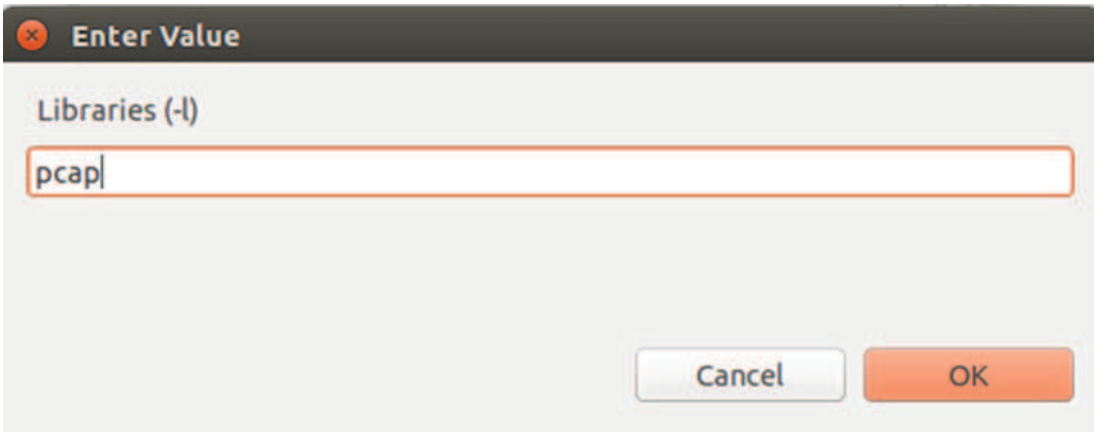
Şekil 5.2. Proje ekranı

1. Sağ tıklanır ve properties içine girilir.
2. C/C++ Build -> Settings -> Tool Settings – Libraries içine girilir.



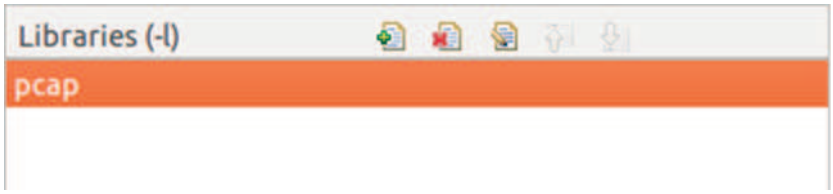
Şekil 5.3. Uygulama ayarları

3. Libraries içinde add komutuna tıklanır ve “pcap” yazılır ve ok tıklanır.



Şekil 5.4. Kütüphane ekleme

4. Böylece pcap kütüphanesi çalıştıracağımız projemiz içine eklenmiş olur.



Şekil 5.5. Pcap kütüphanesi

Paket yakalamak için kullanılan 'C kodu' ve açıklaması aşağıdaki gibidir:

```
#include <stdio.h>
#include <stdlib.h>
#include <pcap.h>
#include <string.h>
#include <unistd.h>
#define MAXBYTES2CAPTURE 2048

/* processPacket():pcap_loop() ile yapılan geri çağırma işleminde paket her zaman ağ kartına ulaşır. */
/* bu fonksiyon yakalanan işlenmemiş veriyi onaltılık olarak yazdırır. */

void processPacket(u_char *arg, const struct pcap_pkthdr* pkthdr, const u_char * packet){
 int i=0, *counter = (int *)arg;
 printf("Packet Count: %d\n", ++(*counter));
 printf("Received Packet Size: %d\n", pkthdr->len);
 printf("Payload:\n");
 for (i=0; i<pkthdr->len; i++){
   if ( isprint(packet[i]) ) /* Eğer yazdırılabilir karakter ise yazdır.*/
      printf("%c ", packet[i]);
   else
```

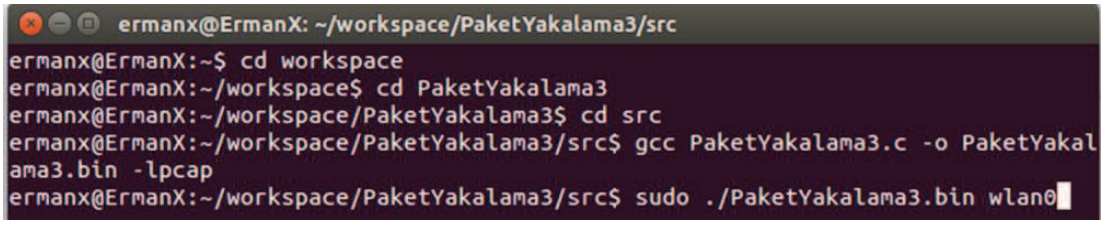
```
      printf(". ");
    if( (i%16 == 0 && i!=0) || i==pkthdr->len-1 )
      printf("\n");
  }
 return;
}

/* main(): Ana görev. Ağ arayüzünü açar ve pcap_loop()'u çağırır. */
int main(int argc, char *argv[] ){
 int i=0, count=0;
 pcap_t *descr = NULL;
 char errbuf[PCAP_ERRBUF_SIZE], *device=NULL;
 memset(errbuf,0,PCAP_ERRBUF_SIZE);
 if( argc > 1){  /* Eğer kullanıcı arayüz adını temin ediyorsa bu adı kullan. */
   device = argv[1];
 }
 else{ /* Bu durumda paket yakalamak için uygun olan ilk cihazı kullan. */
   if ( (device = pcap_lookupdev(errbuf)) == NULL){
      fprintf(stderr, "ERROR: %s\n", errbuf);
      exit(1);
   }
 }
 printf("Opening device %s\n", device);
/* Cihazı karışık yani random modda aç*/
 if ( (descr = pcap_open_live(device, MAXBYTES2CAPTURE, 1,  512, errbuf))
== NULL){
   fprintf(stderr, "ERROR: %s\n", errbuf);
   exit(1);
 }
  /* Alınan her pakette döngüyü sonsuza kadar çağır ve processPacket()
fonksiyonunu çağır.*/
 if ( pcap_loop(descr, -1, processPacket, (u_char *)&count) == -1){
   fprintf(stderr, "ERROR: %s\n", pcap_geterr(descr) );
```

```
    exit(1);
  }
return 0;
}
```

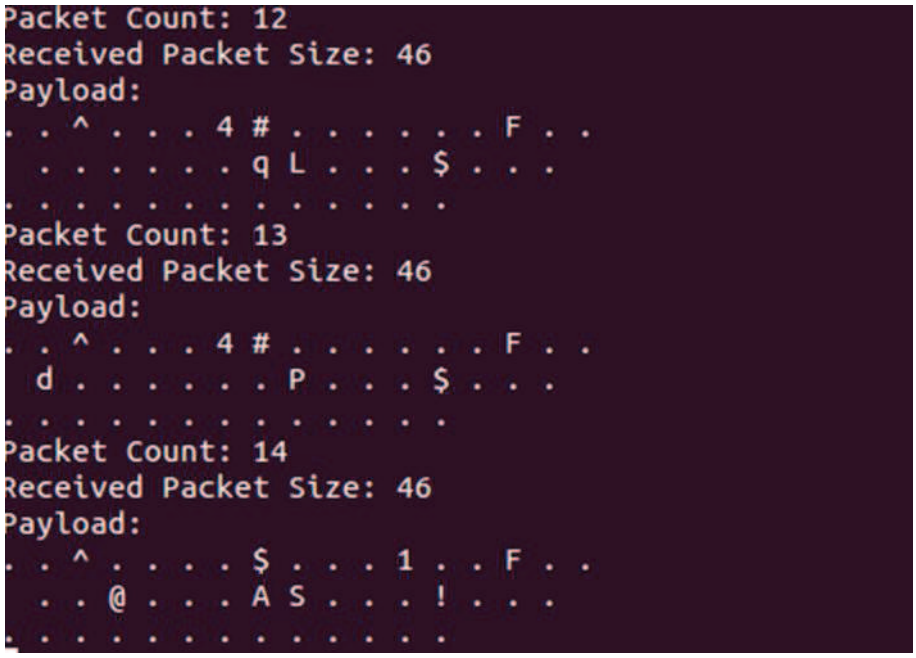
Uygulamanın çalışabilmesi terminalde aşağıdaki komutlar sırasıyla girilmelidir. Öncelikle uygulamaya erişmemiz gerekir.

1. ermanx@ErmanX:~$ cd workspace
2. ermanx@ErmanX:~/workspace$ cd PaketYakalama3
3. ermanx@ErmanX:~/workspace/PaketYakalama3$ cd src
4. ermanx@ErmanX:~/workspace/PaketYakalama3/src$ gcc PaketYakalama3.c -o PaketYakalama3.bin –lpcap
5. ermanx@ErmanX:~/workspace/PaketYakalama3/src$ sudo ./PaketYakalama3.bin wlan0 komutları sırasıyla yazılır ve entere basılır.



Şekil 5.6. Terminal işlemleri

Önce cihaz açılır ve paketler yakalanmaya başlar.



Şekil 5.7. Paket görüntüleri

5.2. Paketlerin Filtrelenmesi

Uygulama ile belirtilen ağ arayüzü üzerinden gelen paketlerle ilgili aşağıdaki veriler elde edilir:

1. Packet number
2. From
3. To
4. Protocol
5. Src port
6. Dst port
7. Payload (eğer TCP türünde paket yakalanıyorsa)

Yakalanan paketlerin sayısına ve protokol türüne göre filtreleme işlemi yapılmıştır. Uygulamada kullanılan protokol türleri aşağıdaki gibidir:

1. TCP
2. UDP
3. ICMP

Paket yakalama ve filtreleme işleminde kullanılan örnek ‘C kodu’:

```
#include <pcap.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define SNAP_LEN 1518
#define SIZE_ETHERNET 14
#define ETHER_ADDR_LEN 6
struct sniff_ethernet {
        u_char  ether_dhost[ETHER_ADDR_LEN];
        u_char  ether_shost[ETHER_ADDR_LEN];
        u_short ether_type;
};
struct sniff_ip {
        u_char  ip_vhl;
        u_char  ip_tos;
        u_short ip_len;
        u_short ip_id;
        u_short ip_off;
        #define IP_RF 0x8000
        #define IP_DF 0x4000
        #define IP_MF 0x2000
        #define IP_OFFMASK 0x1fff
        u_char  ip_ttl;
        u_char  ip_p;
        u_short ip_sum;
```

```
        struct  in_addr ip_src,ip_dst;
};
#define IP_HL(ip)               (((ip)->ip_vhl) & 0x0f)
#define IP_V(ip)                (((ip)->ip_vhl) >> 4)
typedef u_int tcp_seq;
struct sniff_tcp {
        u_short th_sport;
        u_short th_dport;
        tcp_seq th_seq;
        tcp_seq th_ack;
        u_char  th_offx2;
#define TH_OFF(th)      (((th)->th_offx2 & 0xf0) >> 4)
        u_char  th_flags;
        #define TH_FIN  0x01
        #define TH_SYN  0x02
        #define TH_RST  0x04
        #define TH_PUSH 0x08
        #define TH_ACK  0x10
        #define TH_URG  0x20
        #define TH_ECE  0x40
        #define TH_CWR  0x80
        #define TH_FLAGS
(TH_FIN|TH_SYN|TH_RST|TH_ACK|TH_URG|TH_ECE|TH_CWR)
        u_short th_win;
        u_short th_sum;
        u_short th_urp;
};
void
got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char
*packet);
void
print_payload(const u_char *payload, int len);
void
```

```
print_hex_ascii_line(const u_char *payload, int len, int offset);
void
print_hex_ascii_line(const u_char *payload, int len, int offset)
{
 int i;
 int gap;
 const u_char *ch;
 printf("%05d   ", offset);
 ch = payload;
 for(i = 0; i < len; i++) {
  printf("%02x ", *ch);
  ch++;
  if (i == 7)
   printf(" ");
 }
 if (len < 8)
  printf(" ");
 if (len < 16) {
  gap = 16 - len;
  for (i = 0; i < gap; i++) {
   printf("   ");
  }
 }
 printf("   ");
 ch = payload;
 for(i = 0; i < len; i++) {
  if (isprint(*ch))
   printf("%c", *ch);
  else
   printf(".");
  ch++;
 }
 printf("\n");
```

```
return;}
void
print_payload(const u_char *payload, int len)
{
 int len_rem = len;
 int line_width = 16;
 int line_len;
 int offset = 0;
 const u_char *ch = payload;
 if (len <= 0)
  return;
 if (len <= line_width) {
  print_hex_ascii_line(ch, len, offset);
  return;
 }
 for ( ;; ) {
  line_len = line_width % len_rem;
  print_hex_ascii_line(ch, line_len, offset);
  len_rem = len_rem - line_len;
  ch = ch + line_len;
  offset = offset + line_width;
  if (len_rem <= line_width) {
   print_hex_ascii_line(ch, len_rem, offset);
   break;
  }
 }
return;
}
void
got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char
*packet)
{
 static int count = 1;
```

```
const struct sniff_ethernet *ethernet;
const struct sniff_ip *ip;
const struct sniff_tcp *tcp;
const char *payload;
int size_ip;
int size_tcp;
int size_payload;
printf("\nPacket number %d:\n", count);
count++;
ethernet = (struct sniff_ethernet*)(packet);
ip = (struct sniff_ip*)(packet + SIZE_ETHERNET);
size_ip = IP_HL(ip)*4;
if (size_ip < 20) {
 printf("   * Invalid IP header length: %u bytes\n", size_ip);
 return;
}
printf("       From: %s\n", inet_ntoa(ip->ip_src));
printf("         To: %s\n", inet_ntoa(ip->ip_dst));
switch(ip->ip_p) {
 case IPPROTO_TCP:
  printf("   Protocol: TCP\n");
  break;
 case IPPROTO_UDP:
  printf("   Protocol: UDP\n");
  return;
 case IPPROTO_ICMP:
  printf("   Protocol: ICMP\n");
  return;
 case IPPROTO_IP:
  printf("   Protocol: IP\n");
  return;
 default:
  printf("   Protocol: unknown\n");
```

```
  return;
 }
 tcp = (struct sniff_tcp*)(packet + SIZE_ETHERNET + size_ip);
 size_tcp = TH_OFF(tcp)*4;
 if (size_tcp < 20) {
  printf("   * Invalid TCP header length: %u bytes\n", size_tcp);
  return;
 }
 printf("   Src port: %d\n", ntohs(tcp->th_sport));
 printf("   Dst port: %d\n", ntohs(tcp->th_dport));
 payload = (u_char *)(packet + SIZE_ETHERNET + size_ip + size_tcp);
 size_payload = ntohs(ip->ip_len) - (size_ip + size_tcp);
 if (size_payload > 0) {
  printf("   Payload (%d bytes):\n", size_payload);
  print_payload(payload, size_payload);
 }
return;
}
int main(int argc, char **argv)
{
 char *dev = NULL;
 char errbuf[PCAP_ERRBUF_SIZE];
 pcap_t *handle;
 char filter_exp[] = "ip";
 struct bpf_program fp;
 bpf_u_int32 mask;
 bpf_u_int32 net;
 int num_packets ;
 if (argc == 2) {
  dev = argv[1];
 }
 else if (argc > 3) {
  fprintf(stderr, "error: unrecognized command-line options\n\n");
```

```
printf("Usage: %s [interface]\n", argv[0]);
printf("\n");
printf("Options:\n");
printf("    interface    Listen on <interface> for packets.\n");
printf("\n");
 exit(EXIT_FAILURE);
}
else {
 dev = pcap_lookupdev(errbuf);
 if (dev == NULL) {
  fprintf(stderr, "Couldn't find default device: %s\n",
      errbuf);
  exit(EXIT_FAILURE);
 }
}
printf("\nEnter no. of packets you want to capture: ");
       scanf("%d",&num_packets);
       printf("\nWhich kind of packets you want to capture : ");
       scanf("%s",filter_exp);
if (pcap_lookupnet(dev, &net, &mask, errbuf) == -1) {
 fprintf(stderr, "Couldn't get netmask for device %s: %s\n",
     dev, errbuf);
 net = 0;
 mask = 0;
}
printf("Device: %s\n", dev);
printf("Number of packets: %d\n", num_packets);
printf("Filter expression: %s\n", filter_exp);
handle = pcap_open_live(dev, SNAP_LEN, 1, 1000, errbuf);
if (handle == NULL) {
 fprintf(stderr, "Couldn't open device %s: %s\n", dev, errbuf);
 exit(EXIT_FAILURE);
}
```

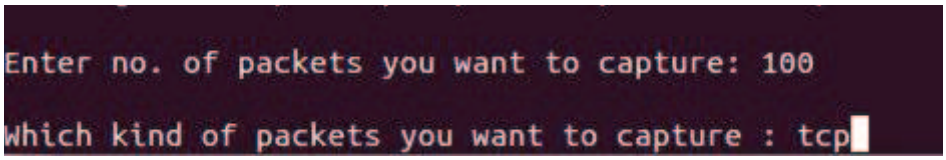
```
 if (pcap_datalink(handle) != DLT_EN10MB) {
  fprintf(stderr, "%s is not an Ethernet\n", dev);
  exit(EXIT_FAILURE);
 }
 if (pcap_compile(handle, &fp, filter_exp, 0, net) == -1) {
  fprintf(stderr, "Couldn't parse filter %s: %s\n",
     filter_exp, pcap_geterr(handle));
  exit(EXIT_FAILURE);
 }
 if (pcap_setfilter(handle, &fp) == -1) {
  fprintf(stderr, "Couldn't install filter %s: %s\n",
     filter_exp, pcap_geterr(handle));
  exit(EXIT_FAILURE);
 }
 pcap_loop(handle, num_packets, got_packet, NULL);
 pcap_freecode(&fp);
 pcap_close(handle);
 printf("\nCapture complete.\n");
return 0;
}
```

Oluşturduğumuz c dosyasının ismi PaketYakalama4.c olduğunu var sayalım. Terminalde sırasıyla şu işlemler yapılmalıdır:

1. ermanx@ErmanX:~$ cd workspace
2. ermanx@ErmanX:~/workspace$ cd PaketYakalama4
3. ermanx@ErmanX:~/workspace/PaketYakalama4$ cd src
4. ermanx@ErmanX:~/workspace/PaketYakalama4/src$ gcc PaketYakalama4.c -o PaketYakalama4.bin -lpcap
5. ermanx@ErmanX:~/workspace/PaketYakalama4/src$ sudo ./PaketYakalama4.bin wlan0 komutları sırasıyla yazılır ve entere basılır.

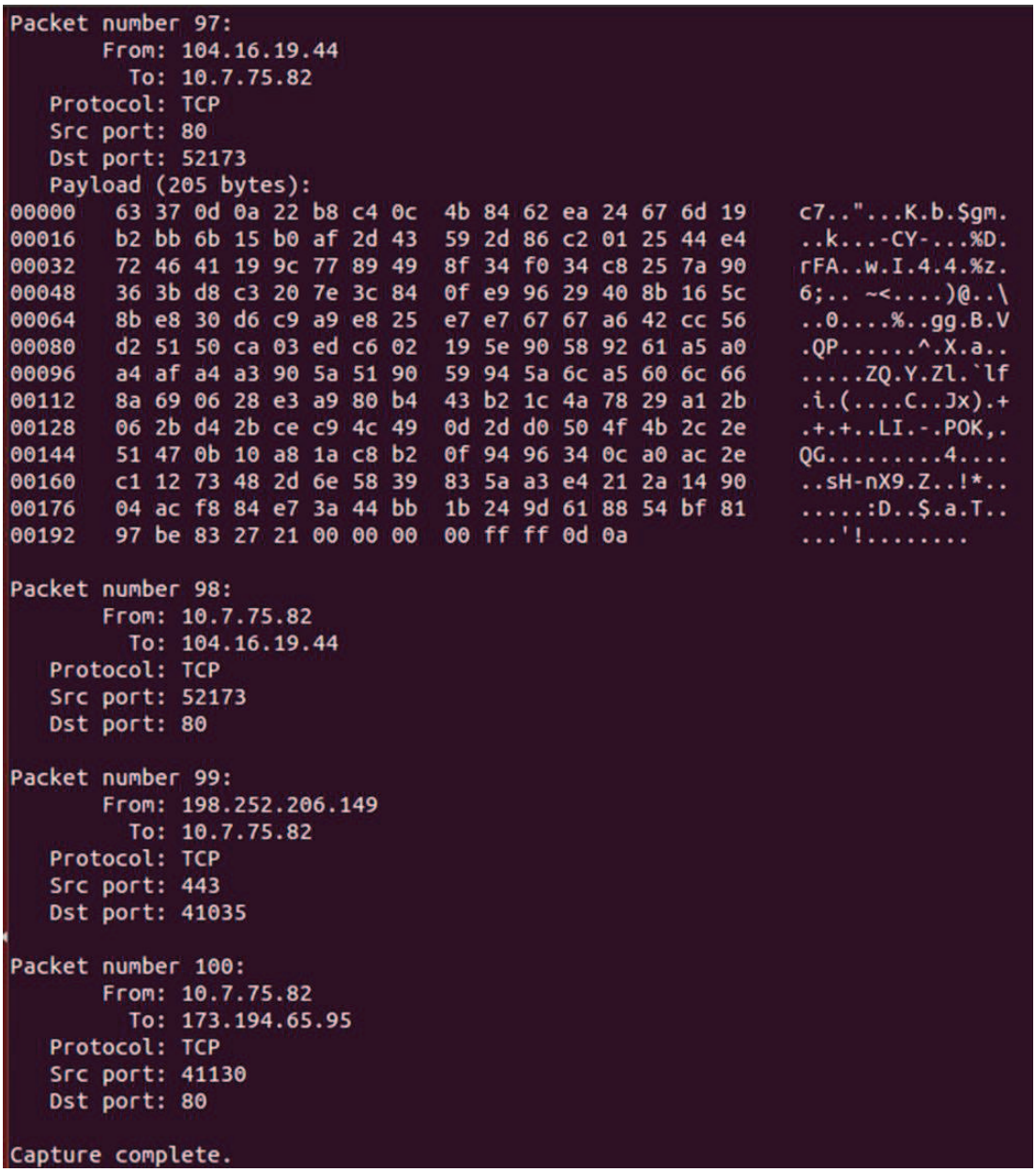
Uygulama kısmında C program dilini kullanarak geliştirilen program ile ağ trafiği

dinlenerek paketler yakalanmaktadır. Şekil 5.8'de görüldüğü gibi yakalanmak istenen paket sayısı ve paketlerin türüne göre filtrelenme işlemi yapılmıştır. Çalıştırılan uygulamada 100 adet paketin yakalanması ve tcp türünde filtrelenmesi istenmiştir.



Şekil 5.8. Paket filtreleme

Şekil 5.9'de yakalanan paketlerden herhangi birinin terminaldeki görüntüsü eklenmiştir. Şekilde de görüldüğü üzere paketler; paket numarası, geldiği adres, gönderildiği adres, protokol türü, kaynak portu, hedef portu ve payload(paket hakkındaki veri) gibi türlere göre yakalanmıştır.



Şekil 5.9. TCP türünde paket

5.3. Yakalanan Ve Filtrelenen Paketlerin Veritabanına Kaydedilmesi

Veri tabanı oluşturmadan önce sistemize aşağıdaki yüklemeleri yapmalıyız. Komut satırını açtıktan sonra aşağıdaki komutları sırasıyla yazmalıyız.

$ sudo apt-get update

$ sudo apt-get install mysql-server-5.5

$ sudo apt-get install mysql-client-5.5

$ sudo apt-get install mysql-common
$ sudo apt-get install glade
$ sudo apt-get install ntp

Bu komutları yazdıktan sonra genelde ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES) hatası alırız. Bu hatanın çözümü ve mysql root şifresinin çözümü aynıdır. Komut satırına aşağıdaki komutu yazıp mysql root şifresini çözmüş oluruz:

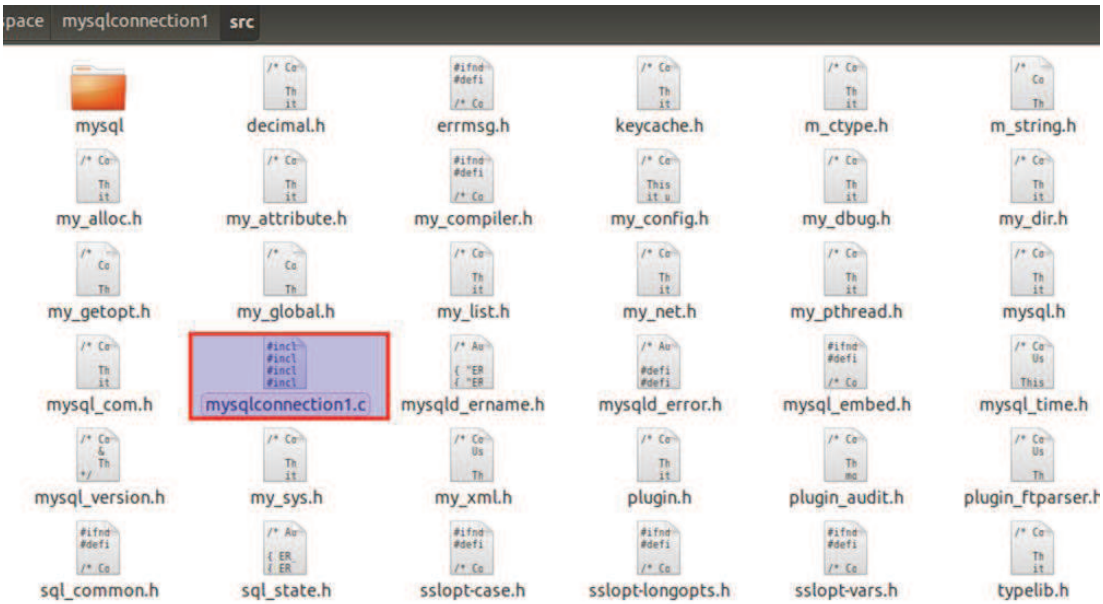
$ sudo dpkg-reconfigure mysql-server-5.5

Ubuntu üzerinde Mysql ve C arasında bağlantıyı yapmak ve çalıştırmak için sırasıyla;

1. Mysql ve c konnektörünün linux versiyounu indirmemiz gerek. Örnek olarak aşağıdaki bağlantıdan indirilebilinir:
   http://dev.mysql.com/downloads/connector/c/
2. Dosyamız sıkıştırılmış olarak indirilir ve bu dosyayı çıkartmalıyız.
3. Klasörün içindeki “include” klasörü içindeki dosyalar kendi oluşturduğumuz projenin içine aktarılmalıdır.

Şekil 5.10. Veritabanı kütüphaneleri

4. Kendi oluşturduğumuz projemizin içine aktarmalıyız.



Şekil 5.11. Proje ve veritabanı kütüphaneleri

Projemizin içindeki dosyalarımızda uzantısı .c olan dosyamız kendi src dosyamızdır. Veri tabanını oluştururken yazacağımız c kodu şu şekildedir:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "my_global.h"
#include "mysql.h"
int main(int argc, char **argv)
{
  MYSQL *con = mysql_init(NULL);
  if (con == NULL)
  {
      fprintf(stderr, "%s\n", mysql_error(con));
      exit(1);
  }
  if (mysql_real_connect(con, "localhost", "root", "password",
          NULL, 0, NULL, 0) == NULL)
  {
      fprintf(stderr, "%s\n", mysql_error(con));
      mysql_close(con);
      exit(1);
  }
  if (mysql_query(con, "CREATE DATABASE testdb"))
  {
      fprintf(stderr, "%s\n", mysql_error(con));
      mysql_close(con);
      exit(1);
  }
  mysql_close(con);
  exit(0);
}
```

Oluşturduğumuz c kodunu derlemek ve veritabanını oluşturmak için komut satırına aşağıdaki komutu girmemiz gereklidir:

```
$ gcc -I/usr/include/mysql XXX.c -lmysqlclient -o XXX –Wall
```

$ ./xxx.bin

komutuyla projemizi çalıştırırız.

XXX:oluşturduğumuz c dosyasının adı

Terminalde superuser konumundan mysql konumuna geçmek için aşağıdaki komutu yazmalıyız:

$ mysql -u root -p

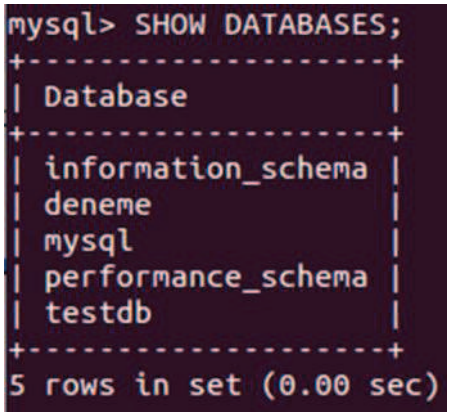
Ekran görüntümüz şu şekilde olacaktır:

mysql>

Veri tabanını görüntülemek için şu komutu yazmamız yeterlidir: Yeni veritabanımızın ismi c kodunda belirtilen testdb'dir.

mysql>SHOW DATABASES;

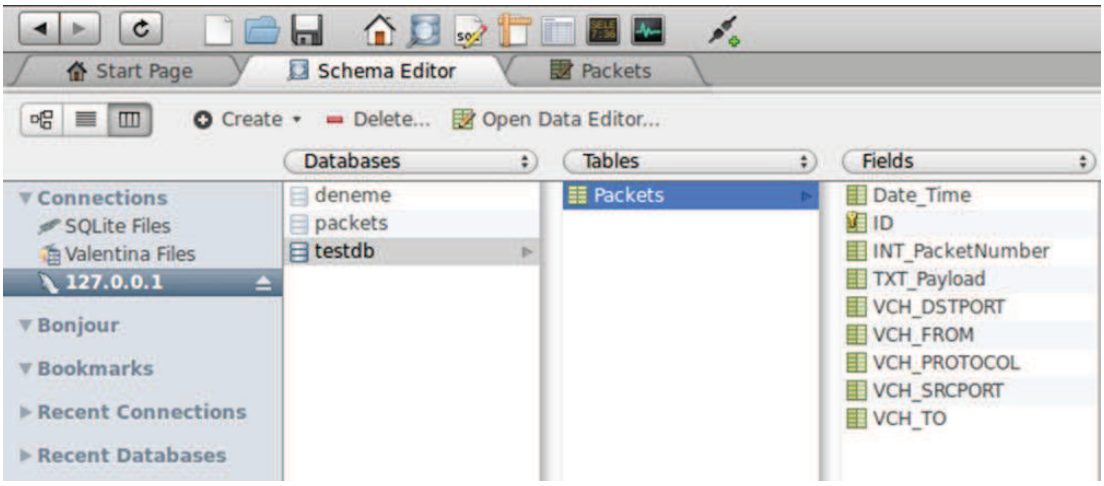
Ekran görüntümüz şekil 5.12'de gösterilmiştir.



Şekil 5.12. Veritabanları

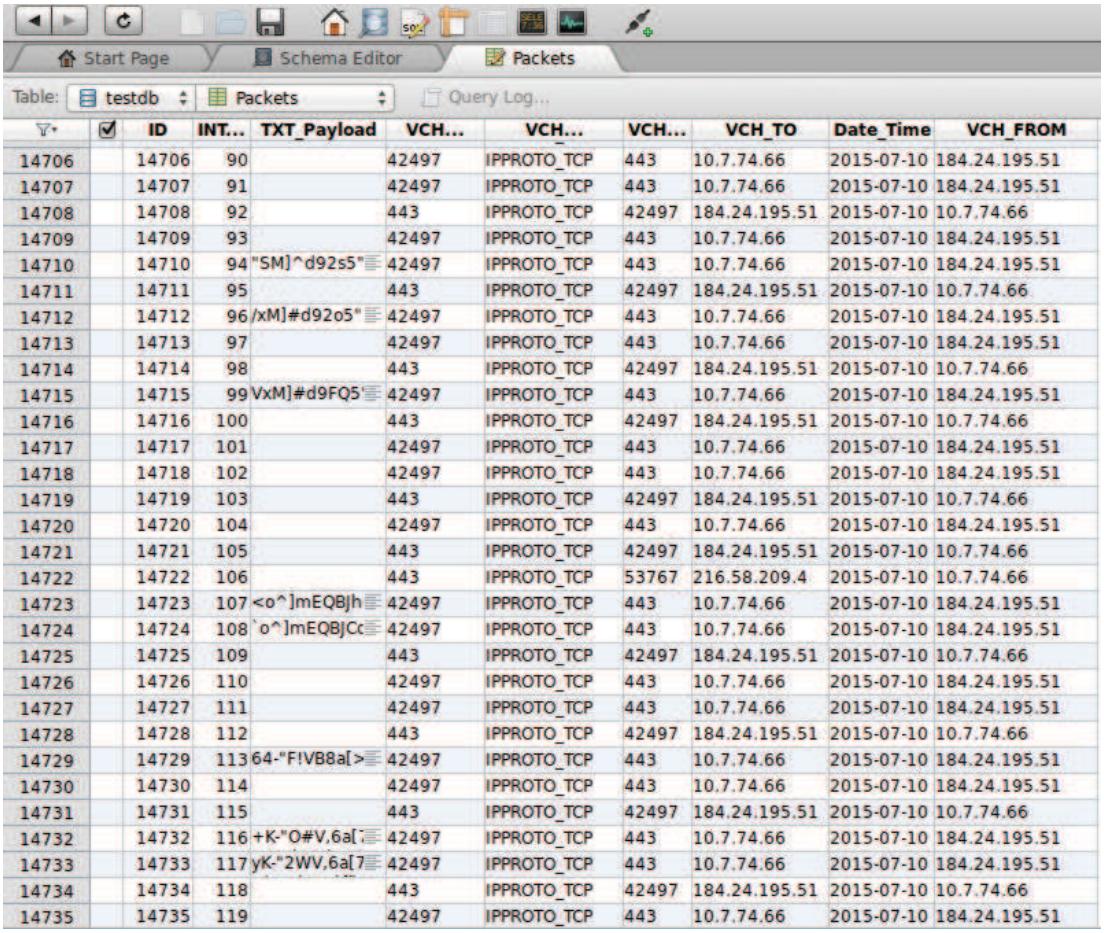
Derin paket analizi yapılarak yakalanan paketler Mysql veritabanı kullanılarak kaydedilmektedir. Veritabanı 9 kısımdan oluşmaktadır, terminaldeki 7 ekran

görüntüsü dışında zaman ve id alanı da bulunmaktadır.



Şekil 5.13. Veritabanı ana ekran

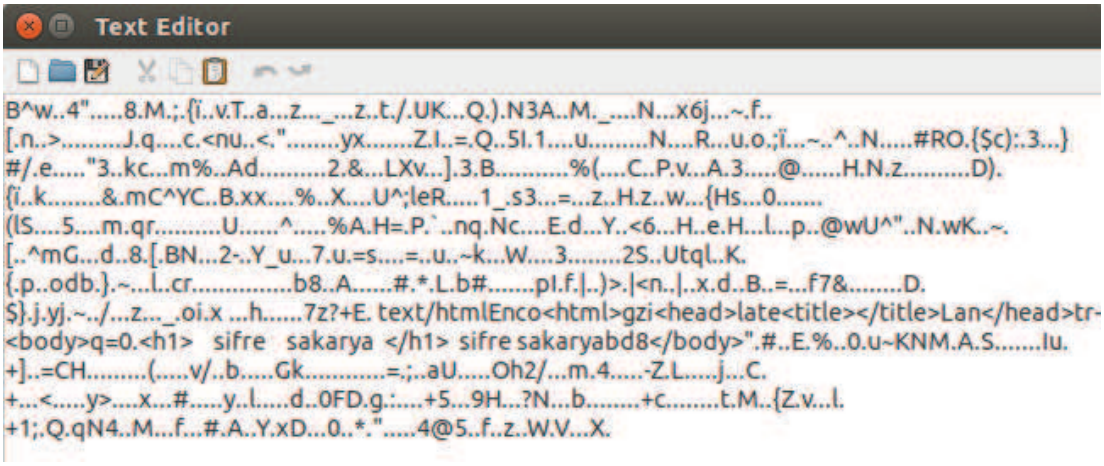
Örnek olarak yakalanan yüz adet Tcp paketi veritabanı görüntüsü şu şekilde gözükmektedir:



	ID	INT...	TXT_Payload	VCH...	VCH...	VCH...	VCH_TO	Date_Time	VCH_FROM
14706	14706	90		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14707	14707	91		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14708	14708	92		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14709	14709	93		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14710	14710	94	"SM]^d92s5"	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14711	14711	95		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14712	14712	96	/xM]#d92o5"	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14713	14713	97		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14714	14714	98		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14715	14715	99	VxM]#d9FQ5'	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14716	14716	100		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14717	14717	101		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14718	14718	102		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14719	14719	103		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14720	14720	104		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14721	14721	105		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14722	14722	106		443	IPPROTO_TCP	53767	216.58.209.4	2015-07-10	10.7.74.66
14723	14723	107	<o^]mEQBJh	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14724	14724	108	`o^]mEQBJCc	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14725	14725	109		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14726	14726	110		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14727	14727	111		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14728	14728	112		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14729	14729	113	64-"F!VB8a[>	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14730	14730	114		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14731	14731	115		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14732	14732	116	+K-"O#V,6a[	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14733	14733	117	yK-"2WV,6a[7	42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51
14734	14734	118		443	IPPROTO_TCP	42497	184.24.195.51	2015-07-10	10.7.74.66
14735	14735	119		42497	IPPROTO_TCP	443	10.7.74.66	2015-07-10	184.24.195.51

Şekil 5.14. Kaydedilen paketlerin veritabanı görüntüsü

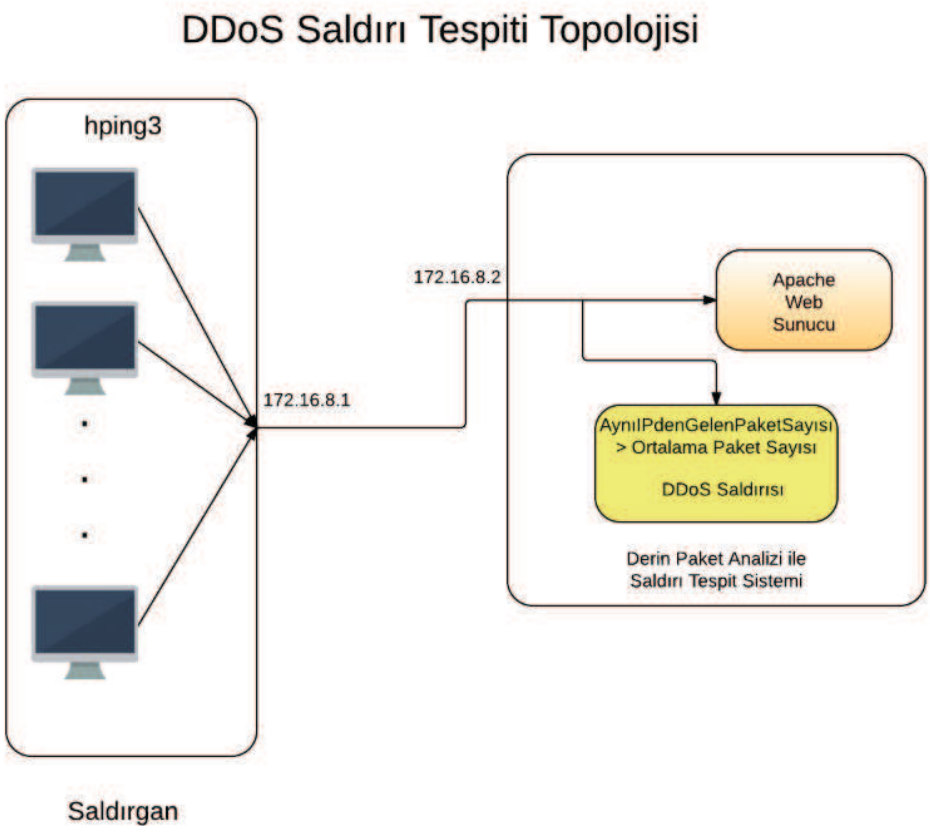
Payload kısmı kaydedilirken yazdırılamayan karakterler “.” olarak alınmıştır. Payload kısmı kaydedilirken karakterler bir dizinin içine aktarılmıştır ve döngü bittiğinde dizinin içeriği temizlenmiştir. Eğer payload kısmı yok ise veritabanında “NULL” olarak gösterilmiştir. Şekil 5.15’te herhangi TCP paketinin payload görüntüsünün veritabanına kaydedilmiş şekli gösterilmiştir:



Şekil 5.15. TCP payload görüntüsü

5.4. DDoS Uygulaması

Geliştirilen DDoS uygulamasının topolojisi şekil 5.16’te görülmektedir:



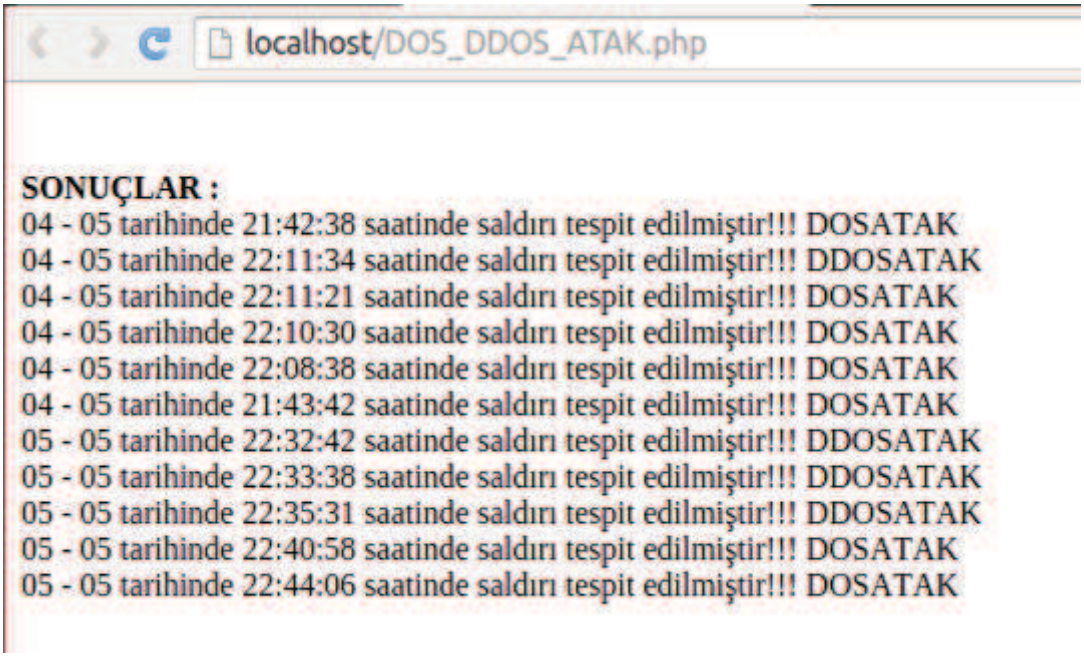
Şekil 5.16. DDoS Uygulamasının topolojisi

Paketler anlık olarak kaydedilmektedir. Uygulamada ortalama paket sayısına bakılarak saldırı tespiti yapılmaktadır. Ortalama paket sayısı 'PaketSayisi.txt' dosyasının içerisindeki veriden çekilerek hesaplanmaktadır. Saldırı tespitinde yakalanan paketler bir önceki paketle kıyaslanmaktadır. Sistemde herhangi sayaç ataması yapılmıştır. DoS saldırı tespitinde kaynak adres(From) ve hedef adres(To) aynı olup olmadığına dikkat edilmiştir. İki adreste aynı olduğu durumlarda sayaç birer birer artırılmıştır. Sayaç miktarı ortalamadan fazla ise sistem durdurulmuştur ve DoS saldırısı tespit edilmiştir yazısı sistemde gösterilmiştir. DDoS saldırı tespitinde kaynak adres(From)'in farklı ve hedef adres(To)'in aynı olup olmadığına dikkat edilmiştir. İki adreste aynı olduğu durumlarda sayaç birer birer artırılmıştır. 2 saniye içinde sayaç miktarı PaketSayısı.txt verisi içindeki değerden fazla ise sistem durdurulmuştur ve DDoS saldırısı tespit edilmiştir yazısı sistemde gösterilmiştir. Paket sayısı değeri 100 olarak belirlenmiştir. Bu değer, aslında önceki istek kayıtlarından hesaplanarak bulunması daha doğru olur. [25] numaralı

makalede 0.5-2 saniye arası değerler verilerek DDoS saldırılarının tespiti edilebileceği belirtilmiştir.

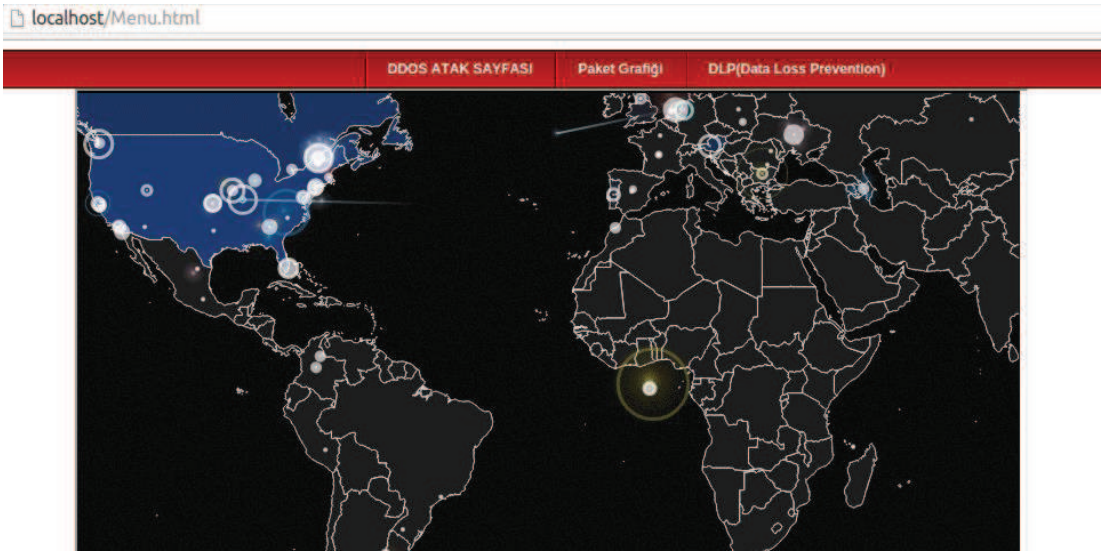
Paketleri oluşturmak için DDoS saldırı araçlarından biri olan 'hping3' kullanılmıştır. Hping3 istenilen türde tcp/ip paketleri oluşturmak için kullanılan bir araçtır.

Oluşan saldırı tespitleri veritabanına kaydedilmektedir. Web arayüzünde veritabanından çekilen saldırılar gösterilmiştir. Saldırılar hangi gün ve saatte olarak ekrana getirilmektedir. Saldırılar gösterilirken veritabanından veriler çekilmektedir. Her iki saniyede bir sistemde saldırı olup olmadığı tespit edilmektedir.



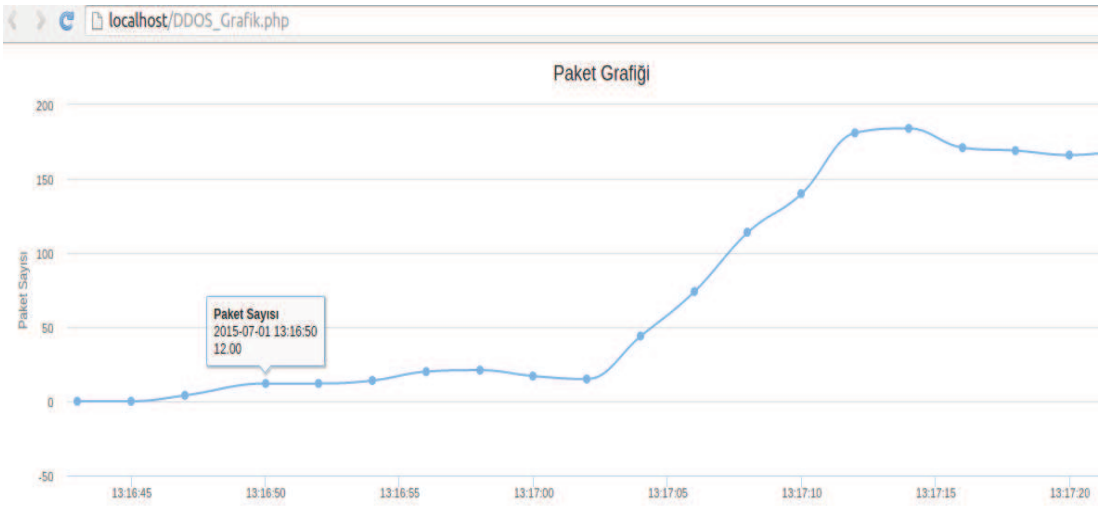
Şekil 5.17. Saldırılar

Uygulamanın bu kısmı php diliyle geliştirilmiştir. Bu uygulama veritabanından bilgileri çekerek tasarlanmıştır. Web arayüzünde üç adet uygulama tasarlanmıştır:



Şekil 5.18. Web ana ekran

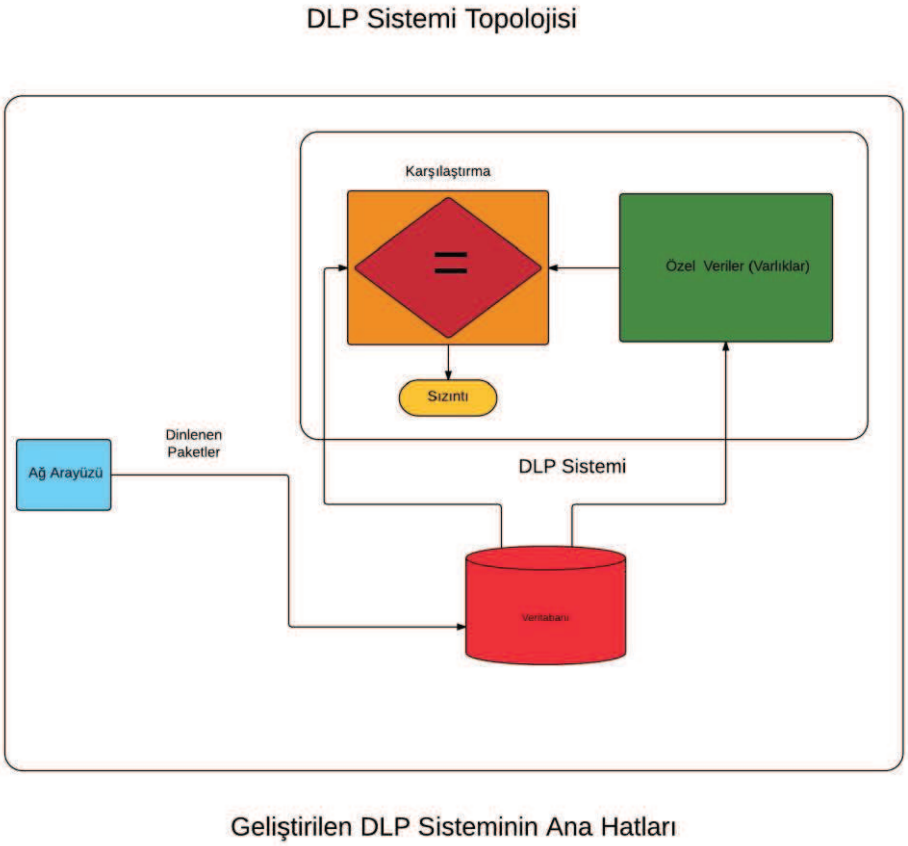
Paket grafiği sayfasında veriler çevrimiçi olarak sürekli güncellenmektedir. Her iki saniyede güncel olarak paket grafiği çizilmektedir. X ekseninde zaman kavramı, y ekseninde paket sayısı gösterilmektedir.



Şekil 5.29. Paket grafiği

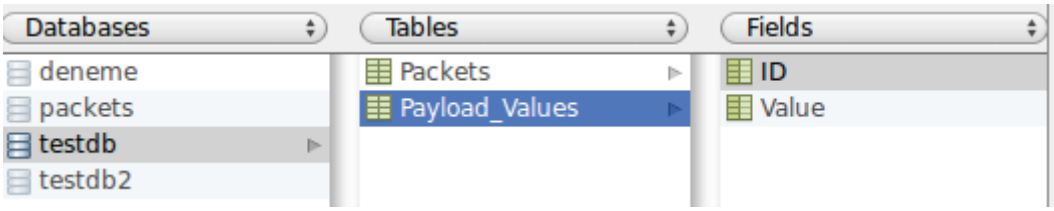
5.5. DLP Uygulaması

Geliştirilen DLP sisteminin blok şeması şekil 5.20'de görülmektedir:



Şekil 5.20. DLP sisteminin blok şeması

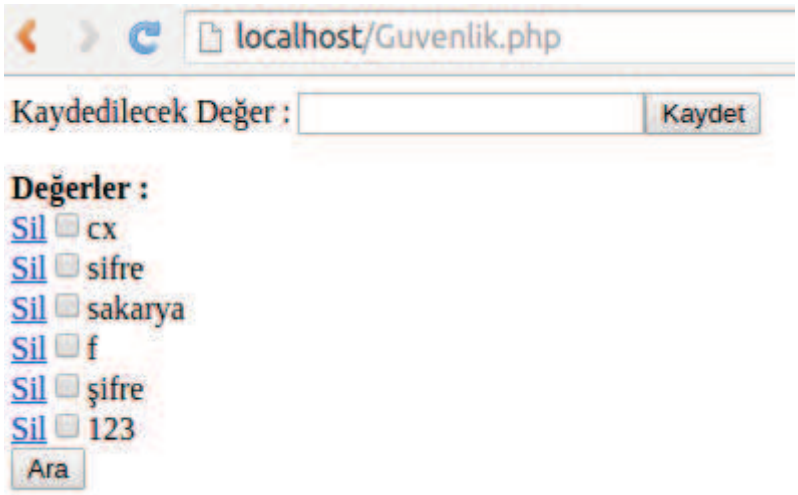
Bu sayfa oluşturulurken ayrı bir veritabanı daha oluşturulmuştur. Payload içeriğinde veri taraması yapılmaktadır. Veritabanındaki veriler sürekli güncellenmektedir. Eklediğimiz veriler payload içerisinde taranmaktadır ve veri ihlali olup olmadığı tespit edilmektedir. Şekil 5.21'de oluşturulan veritabanını gösterilmektedir.



Şekil 5.23. DLP sayfası için oluşturulan veritabanı

DLP sayfasında veriler Payload_Values tablosundan verileri çekmektedir. Eğer Payload_Values içerisinde değerler güncellenirse DLP sayfası da çevrimiçi olarak güncellenmektedir.

Şekil 5.22'de DLP sayfası gösterilmiştir.



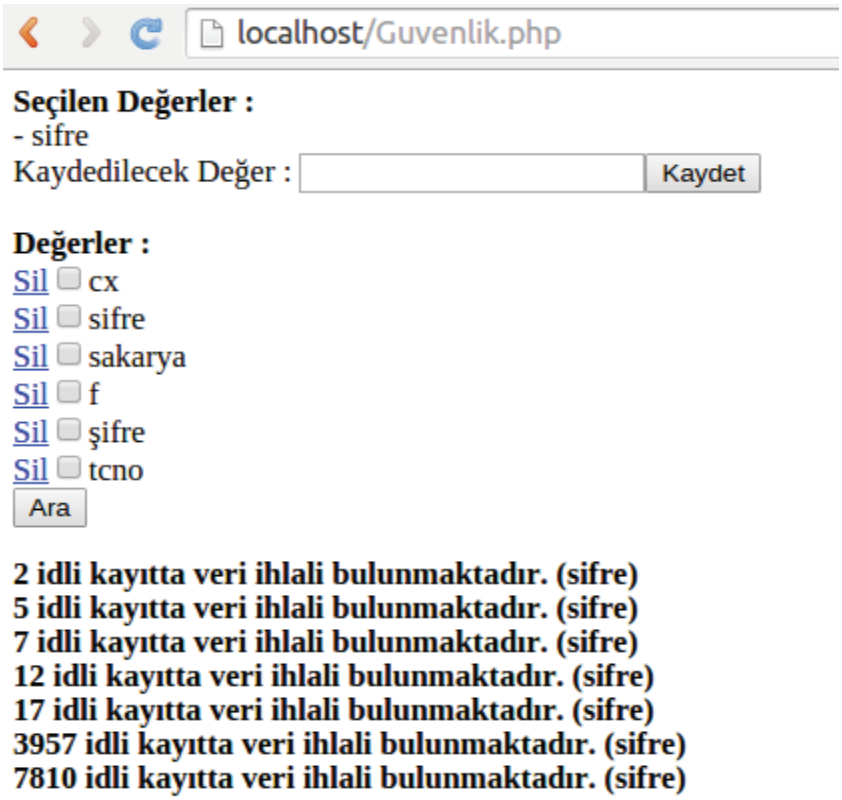
Şekil 5.22. DLP sayfası

Sayfa güncel olarak kullanılmaktadır. Örnek olarak yeni bir değer eklediğimizde hem veritabanında hem de sayfa da güncellenerek eklenmektedir. Değerlerden herhangi birini silmek için değerlerin sol tarafında bulunan sil butonuna basılmalıdır.

Bu sayfada yapılan iki adet örnek açıklanmıştır:

1. Örnek 1:

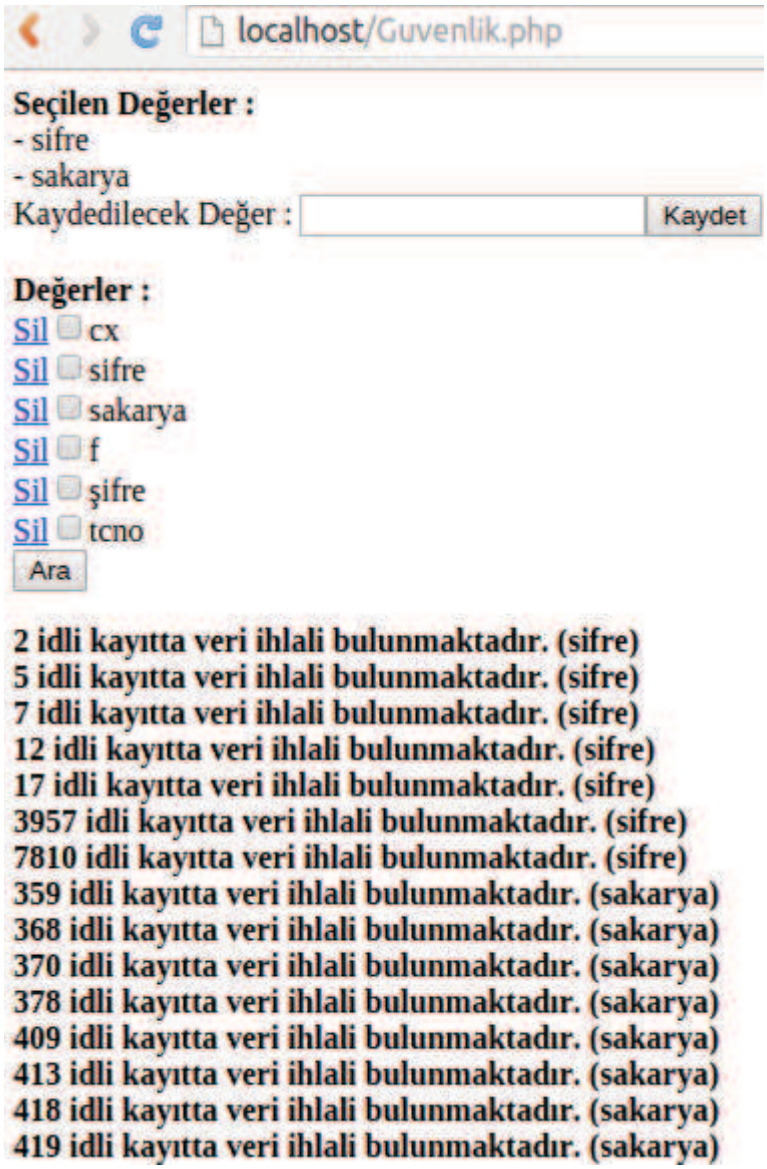
Şeçilen değer "sifre" olsun. Payload içerisindeki değerler tarandıktan sonra veri ihlali olup olmadığı tespit edilmiştir. Sistem kendini her beş saniyede bir güncellemektedir. Herhangi bir ihlal söz konusu olursa yeni ihlaller eklenmektedir.



Şekil 5.23. Örnek 1

2. Örnek 2

Seçtiğimiz değerler hem “sifre” hem de “sakarya” olsun. Payload içerisindeki değerler tarandıktan sonra veri ihlali olup olmadığı tespit edilmiştir.



Şekil 5.24. Örnek 2

BÖLÜM 6. SONUÇ

Bu tez çalışmasıyla birlikte Derin paket analizi yapılmıştır. Ağ üzerinden geçen paketler yakalanmıştır, paketler türüne ve sayısına göre filtrelenmiştir. Gerçek zamanlı olarak ağ üzerinde DoS veya DDoS saldırısı olup olmadığı tespit edilmiştir. Yakalanan paketler veritabanına kaydedilirken aynı zamanda iki saniye aralıkta paket sayıları grafiğe dökülmüştür. Verilerin payload kısmı incelenerek güncel olarak veri ihlali olup olmadığı tespit edilmiştir.

Sonuç olarak siber güvenliğin önemi gün geçtikçe artmaktadır. Siber tehditler inanılmaz boyutlara ulaşmış, şirketlerin ve devletlerin bu konuya yatırımı artmıştır. Şirketler için daha büyük riskler doğmakta, gizlilik ihlalleri, iş kaybı, hizmet kesintisi yaşanmaktadır. Bu yüzden risklerin azalması ve gizlilik ihlallerinin yaşanmaması için altyapıya yönelik çalışmaların daha fazla yapılması gerekmektedir.

KAYNAKLAR

[1] Siber Dünya ve Bilgi Güvenliği, http://www.bilgiustam.com/siber-dunya-ve-bilgi-guvenligi/, Erişim Tarihi:23.06.2015.

[2] Öğün M, Kaya A,Siber Güvenliğin Milli Güvenlik Açısından Önemi ve Alınabilecek Tedbirler, 2013.

[3] Scarfone K, Mell P, Guide to intrusion detection and prevention system(IDPS), 2007.

[4] Newman R, Computer security: protecting digital resources jones & bartlett learning, 2009.

[5] Whitman M, Mattord H, Principles of information security cengage learning EMEA, 2009.

[6] Vacca J, Managing information security syngress, 2010.

[7] Tutkun H, Network sistemleri sistem yöneticisinin el kitabı, Seçkin Yayınevi, 2012.

[8] Çölkesen R, Network TCP/IP unix el kitabı, Papatya Yayınevi, 2001.

[9] Çetin G, Metin B, Linux ağ yönetimi, Seçkin Yayınevi, 2003.

[10] TCP/IP taşıma ve uygulama katmanı, Mesleki eğitim ve öğretim sisteminin güçlendirilmesi projesi bilişim teknolojileri, 2008.

[11] Ketenci S, OSI referans modeli ağ katmanı(network layer), 2011.

[12] Yılmaz D, TCP segmentinin yapısı,2005.

[13] UDP, http://tr.wikipedia.org/wiki/UDP, Erişim Tarihi: 04.04.2015.

[14] Vig A, Preventing denial of service attacks, 2004.

[15] Gezgin DM, Buluş E, Kablosuz ağlar için bir DoS saldırısı tasarımı, 2013.

[16] Denial-of-service attack, http://en.wikipedia.org/wiki/Denial-of-service_attack, Erişim Tarihi: 01.05.2015.

[17] Kumar K, Joshi RC, Singh K, An integrated approach for defending against distributed denial-of-service (DDoS) attacks

[18] Mydoom virüsü, http://www.bbc.co.uk/turkish/news/story/2004/02/040204_mydoom_nedir.shtml, Erişim Tarihi: 01.03.2015

[19] Mydoom virüsü, http://www.turkishajan.com/trojan-spyware-ve-virusler/mydoom-nedir-33365.html, Erişim Tarihi: 01.03.2015

[20] Stachheldraht, http://arsiv.ntv.com.tr/news/45315.asp?cp1=1, Erişim Tarihi: 01.03.2015

[21] Smurf Attacks, http://ekinoks.cu.edu.tr/internet/konu_46.htm, Erişim Tarihi :01.03.2015

[22] Fraggle, http://www.ids-sax2.com/articles/PreventDosAttacks.htm, Erişim Tarihi:01.03.2015

[23] Fraggle, http://in1.csie.ncu.edu.tw/~cs102085/DDoS/amplification/fraggle/description.htm, Erişim Tarihi:01.03.2015

[24] Sargar S, Joshi J, Tipper D., "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," Communications Surveys & Tutorials, IEEE , vol.15, no.4, pp.2046,2069, Fourth Quarter 2013

[25] Jin G, Yang J, Wei W, Dong Y, "Resisting Network DDoS Attacks by Packet Asymmetry Path Marking," Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on , vol., no., pp.1205,1209, 22-24 Aug. 2007.

[26] Yen-Hung H, Hongsik C, Choi, H.-A., "Packet filtering to defend flooding-based DDoS attacks [Internet denial-of-service attacks]," Advances in Wired and Wireless Communication, 2004 IEEE/Sarnoff Symposium on , vol., no., pp.39,42, 26-27 Apr 2004.

[27] Thapngam, T , Shui Yu , Wanlei Zhou, Beliakov, G., "Discriminating DDoS attack traffic from flash crowd through packet arrival patterns," Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on , vol., no., pp.952,957, 10-15 April 2011.

[28] You-ye Sun; Cui Zhang; Shao-qing Meng; Kai-ning Lu, "Modified Deterministic Packet Marking for DDoS Attack Traceback in IPv6 Network," Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on , vol., no., pp.245,248, Aug. 31 2011-Sept. 2 2011.

[29] Data Loss Prevention (DLP), http://sistemuzmaniyiz.blogcu.com/data-loss-prevention-dlp/2123409, Erişim Tarihi:22.06.2015.

[30] DLP (Data Loss Prevention – Veri Sızıntısı Önleme), http://www.biznet.com.tr/tr/dlp, Erişim Tarihi:22.06.2015.

[31] DLP nedir? Örnek çözümler, ürünler, http://innovaktif.com/dlp-nedir-ornek-cozumler-urunler/, Erişim Tarihi:22.06.2015.

ÖZGEÇMİŞ

Erman ÖZER, 01.03.1989'da Mardin'de doğdu. İlk, orta ve lise eğitimini Mardin'de tamamladı. 2007 yılında Mardin Anadolu lisesinden mezun oldu. 2007 yılında başladığı Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği bölümünü 2012 yılında bitirdi. 2013 yılında Recep Tayyip Erdoğan Üniversitesi'nde ÖYP araştırma görevlisi olarak göreve başladı. Ekim 2014 tarihinden itibaren Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği'nde araştırma görevlisi olarak çalışmaktadır.