ÖNSÖZ

Geçmiş 20-30 yılda teknoloji akıl almaz derecede hızlı gelişme göstermiştir. Bu gelişme sürecinin hızlı olmasının başlıca sebeplerinden biri de tartışmasız internettir. En başlarda küçük bir araştırma ağı mevcut iken, günümüzde kendi evimizdeki herhangi bir cihazdan, kilometrelerce ötedeki bir cihazla iletişime geçilebiliyor.
Konu ile ilgili olarak bu bitirme çalışmasında ise , ağ üzerindeki trafiği dinleyerek, ağı kullanan kullanıcıların kaydının tutulması amaçlanmıştır.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

LAN	: Local Area Network
WAN	: Wide Area Network
PAN	: Personal Area Network
MAN	: Metropolitan Area Network
OSI	: Open System İntercorrection
HTTP	: Hyper Text Transfer Protocol
HTTPS	: Hyper Text Transfer Protocol Secure
FTP	: File Transfer Protokol
SMTP	: Simple Mail Transfer Protocol
TCP	: Transfer Control Protocol
UDP	: User Datagram Protocol
IP	: Internet Protocol
IPv4	: Internet Protocol version 4
IPv6	: Internet Protocol version 6
ARP	: Adress Resolution Protocol
DNS	: Domain Name System
DHCP	: Dynamic Host Configuration Protocol
IDS	: Intrusion Detection Systems
TTL	: Time to Live

ŞEKİLLER LİSTESİ

ÖZET

Anahtar kelimeler: TCP/IP, Sniffing, Pcap, ICMP, C#, Ağ

Bilgisayar Ağı her geçen gün büyüyen bir alandır. Ağ oluşturma hayatı kolaylaştırmıştır ve dünya çapında bilgisayar ağına 3 milyardan fazla insan erişmektedir. Bu kadar büyüme hızı bize bu internet dünyasının ne kadar karmaşık olduğunu da göstermektedir. İlk başlarda ABD'nin savunma araştırması ile ortaya çıkan ARPANET, Daha sonra iki farklı bilgisayarı birbirine bağlamak için protokoller oluşturdu. Bu yaratılan şey TCP / IP protokolüdür ve internet bu şekilde doğmuştur. Bilgisayarlar farklı bir ağdaki topolojiler ve ağ protokolleri sayesinde iletişim kurarlar. Ev ağı, ofis ağı, bilgisayarların yerel alan ağı (LAN) daha sonra Geniş alan ağı (WAN) ile bağlantılı hale gelir.

Bu tezde yapılan uygulama ile ağ dinlemesi sonucu nelerle karşılaşılacağı gözlemlenip, internet adına bilgi sahibi olunulacaktır. Hatta ağ daki açıklar dahi bulunup bunun üzerine ne gibi önlemler alınabiliceği düşünülecektir.

BÖLÜM 1. GİRİŞ

1.1. Amaç

Ağ dinleme olayı, Ağın yapılandırması hakkında birçok yararlı bilgi sağlar. Hatta yanlış yapılandırılan rotaların ve filtreleri ortaya çıkarabilir. Fakat kötü amaçlı kişiler bu özelliği kişilerin özel verilerine erişmek için kullanabilir veya verileri anlamasalar bile bundan kazanç sağlayabilirler. Bunun yanında Ağ yöneticisi ise ağlarındaki açıkları bularak, daha güvenli hale getirmek amaçlı kullanabilirler.

1.2. Motivasyon

Mevcut çoğu ağ dinleme uygulamaları paket paket analiz etmeye odaklanır ve yakalanan verileri şifreli bir şekilde sunar. Kullanıcının gördüğü ilk şey, yakalanan paketlerin sürekli uzayan, ondalık sistemde birkaç şekilde ayrıştırılmış ve özetlenmiş bir listesidir. Yinede verilerin akışını takip etmek için seçenek vardır.

1.3. Uygulanabilir Çözümler

Muhtemelen en meşhur ve en çok kullanılan ağ dinleme programları tcpdump ve Wireshark.

Tcpdump neredeyse her ağ yöneticisi için bir zorunluluktur. Ancak sadece dinleme aracıdır. Paket Başlıklarını ayrıştırabilse de yakalanan verileri analiz etmez, yalnızda bunları konsola veya dosyaya döker. Amacı, cihazda belirli bir iletişim olup olmadığını kontrol etmektedir.

Wireshark, günümüzde ağ analizinde standart haline gelen büyük çaplı bir projedir. Bir çok platformda çalışır, yüzlerce protokolü destekler ve analiz kodları bir milyondan fazla satır kod içermektedir. Öte yandan halihazırda yakalanmış verileri çevrimdışı olarak analiz yapabilir.

BÖLÜM 2. VERİ AĞI

Veri ağı, bilgisayar ağı olarak da bilinir. Bir gruptaki bağlı bilgisayarların bilgi veya veri paylaşmak için kablolar veya radyo dalgaları yoluyla iletişim kurmasına veri ağı denir. Bir ağdaki bilgisayarlara düğümler denir. Bilgisayar ağı e-posta alışverişi, internet, video konferans ve daha pek çok avantaj sağlar. Bilgisayar ağlarına örnek verecek olursak, kişisel alan ağı(PAN), yerel alan ağı(LAN), metropolitan alan ağı (MAN), geniş alan ağı(WAN). Genel olarak topoloji, iki bilgisayarı bir yapı, tasarım veya düzenlemede bağlamanın bir yoludur. Topoloji örnekleri Point-to-Point, bus topolojisi, yıldız topolojisi, mesh topolojisi ve fazlası.
Ağ oluşturma, yazılım, kablolar, fiziksel cihazlar nedeniyle oldukça karmaşıktır. Sonuç olarak, ağ iletişimi katmanlara bölünür. Katmanlar birbirlerinin üzerine kurulmuştur. Bu çoklu katmanlar veri alışverişinde bulunur, her katman belirli bir görevi yerine getirir ama birbirinden bağımsızlardır. Her katman iletişim için bir dizi protokol veya kural içerir.

2.1. OSI Modeli

Bilgisayar Ağları kullanılarak bilgisayarların birbirleri ile haberleşmeye başladıkları ilk yıllarda iki bilgisayarın birbiri ile haberleşmesi için aynı marka/model kullanmaları gerekiyordu. Farklı üreticiler tarafından üretilen cihazlar birbirleri ile haberleşmeleri sorunsuz olması için çeşitli standartlar geliştirilmiştir.

Bunu üzerine OSI (Open Systems Interconnection) referans modeli oluşturuldu. 7 katmandan oluşan bir model.

KATMANLAR

7.Application---------->HTTP,HTTPS,SMTP,FTP,TFTP,UUCP,NNTP,SSL....

6.Presentation---------->ISO 8822,ISO8823,ISO8824,ITU-T T.73...

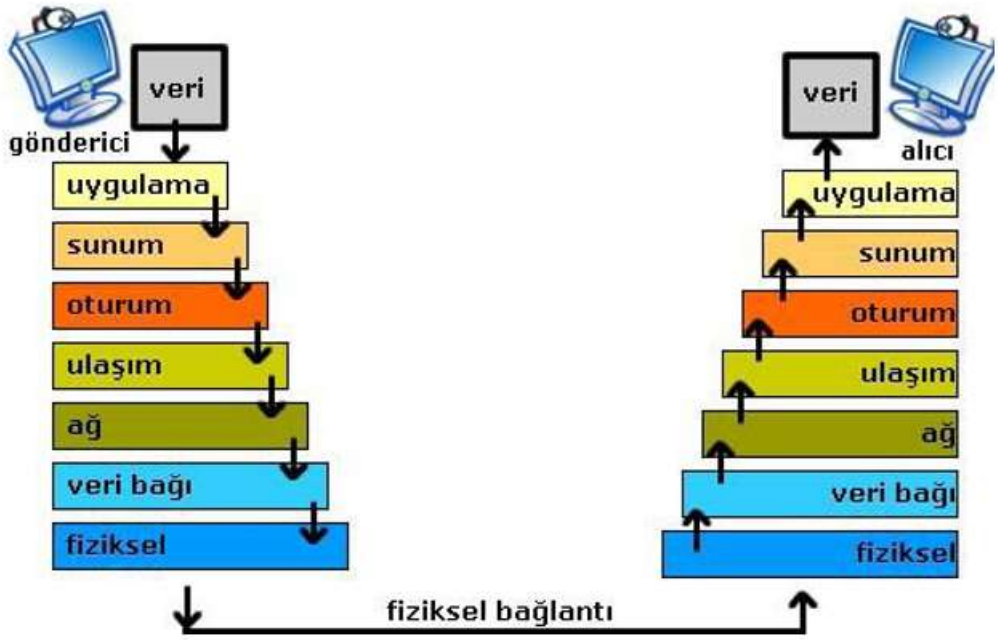
5.Session(Oturum)---------->SMB,ISO 8326,NFS,ISO8327....

4.Transport(Ulaşım)---------->TCP,UDP.....

3.Network---------->IP,Ipv4,Ipv6,ICMP,ARP,IGMP.....

2.Data Link---------->Ethernet,HDLC,Wi-Fi, Token ring,...

1.Physical---------->ISND,RS-232.....



Şekil 2.1. OSI Modeli

OSI modeli iletişim standartlarını belirlemeye yöneliktir ve TCP/IP daha uygulanabilir bir model olduğu için daha çok uygulamaya yöneliktir.

2.2. TCP/IP Mimarisi

4 Katmandan oluşan bir referans modelidir.

Veri Kapsülleme (Data Encapsulation): Veri gönderilirken, verinin uygulama katmanından fiziksel katmana doğru ilerlemesiyle, her bir katmanda yeni başlık bilgilerinin eklenmesi sürecine denir.

Segment = Taşıma katmanındaki hali
Datagram = Ağ katmanındaki hali
Frame = Veri bağı katmanındaki hali

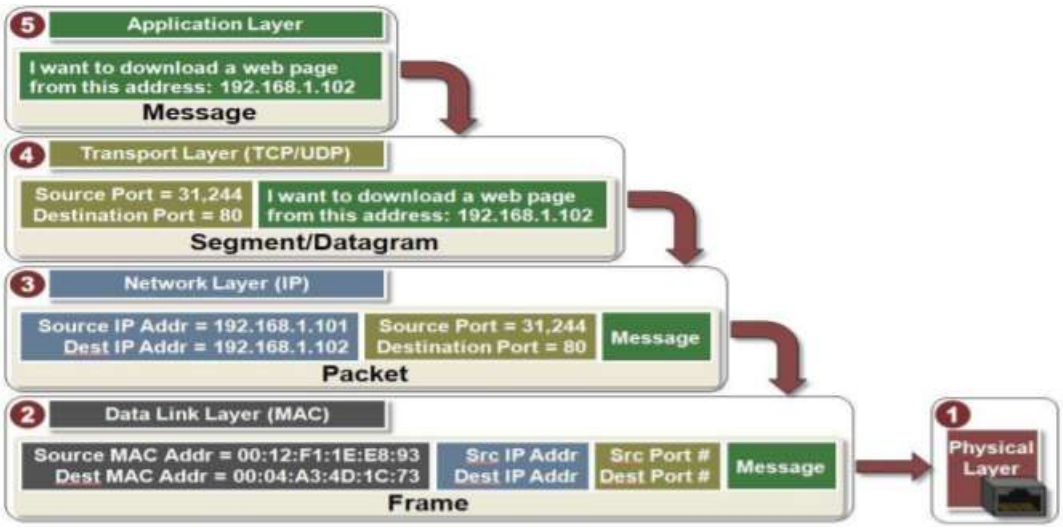
4.Application (Uygulama)---------->FTP, HTTP, SMTP, DNS, NFS

3.Transport (Ulaşım)---------->TCP, UDP

2.Network (Ağ)---------->ICMP, IP, ARP, DHCP, RARP

1.Network Access(Ağ Arayüzü)--->Ethernet, Token Ring, Lan, Wan

2.3. Ağ Katmanları



Şekil 2.3. Ağ Katman Modeli

2.3.1. Uygulama Katmanı

Bu katmanda gönderilecek veri tipi ve veriyi işleyen uygulamalar bulunur. Örneğin bir HTML web sayfası isteğine karşılık HTTP protokolü uygulanır. E posta gönderimi için SMTP (Simple Mail Transfer Protocol) protokolü bulunur. Dosya gönderimi için FTP kullanılır. Taşıma katmanı ile portlar yardımıyla haberleşir. (HTTP:80, FTP:21,SMTP:25 HTTPS:443 DNS:53)

2.3.2. Taşıma Katmanı

Bu katmanda verinin nasıl gönderileceği belirlenir. TCP ve UDP bu katmandadır. TCP klasik veri aktarımında, UDP ise medya aktarımında kullanılır. TCP, UDP (User Datagram Protocol) ye göre daha güvenli fakat daha yavaştır. Çünkü TCP'de gönderilen her veri paketinin ardından yerine doğru şekilde ulaşıp ulaşılmadığı kontrol edilir. TCP ve UDP iletim esnasında veriye içinde bazı kontrol bilgilerinin bulunduğu başlık (header) ekler. TCP protokolü, gönderilen veriler için özel bir TCP kabul paketi (TCP ACK) gönderilir ve gelmiş olan paketlerin doğruluğu kontrol edilir.

2.3.3. Ağ Katmanı

IP katmanı olarak adlandırılan bu katman verilerin gideceği adres veriye eklenir yani veri bu katmandan gönderilir ve yönlendirilir.

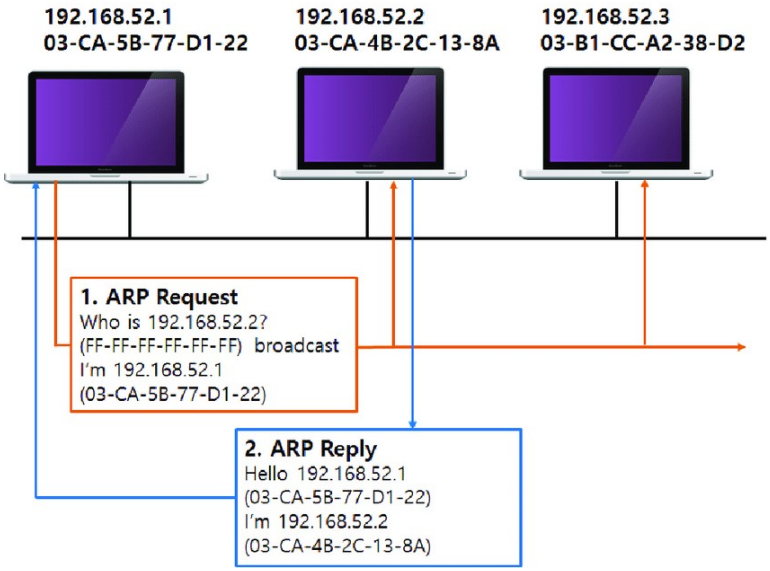
2.3.4. Fiziksel Katmanı

Bu katman verinin hangi yolla gönderileceği belirlenir. İletişim ortamının özelliklerini, haberleşme hızını ve kodlama şeması belirler.

2.4. Uçtan uca prensibi

Bilgisayar ağlarında, bir ana bilgisayar diğer bilgisayarla anca fiziksel adresini biliyor ise iletişim kurabilir. ARP (Adress Resolution Protocol) 2. katman da olan ve genelde MAC adresi olarak bilinen fiziksel adresi öğrenmek için kullanılan bir protokoldür. Ne zaman kaynak ve hedef IP adresli bir paket 2 katmana ulaşırsa, bağlantı katmanı paketi kaynak ve heder MAC'e ihtiyaç duyduğu çerçeve içinde iletmesi gerekir.

İlk başta ARP, heder MAC adresi için ARP önbelleğini kontrol eder. ARP önbellek tablosu, IP adresini MAC adresi ile eşler. Önbellek içinde bulunmazsa, ana bilgisayar ARP isteğini yayın olarak, yerel ağına bağlı her bilgisayara gönderir. Hedef MAC adresi FF:FF:FF:FF:FF:FF şeklinde doldurularak ağa yollanır.



Şekil 2.4. ARP Örneği

BÖLÜM 3. AĞ DİNLEME VE KULLANILAN TEKNOLOJİLER

3.1. Ağ Dinleme Nedir?

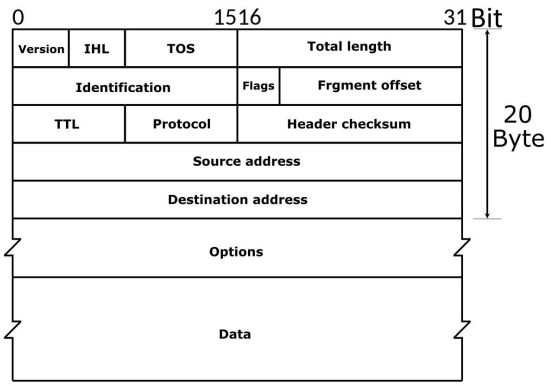
Koklama (Sniffing), telefon dinlemenin bilgisayar ağı eşdeğeridir. Sniffing aslında telden geçen veri paketlerini okumaktan başka bir şey değildir. Dinleme yapılan veriler bir şekilde telefon dinlemesinden daha karmaşık ve rasgeledir. Bu nedenle Sniffing araçları, verilerin kodunu çözme özelliğine sahiptir.

Bir Sniffer, Koklama (Sniffing) için kullanılan herhangi bir yazılım aracıdır. Sniffer uygulamaları baz olarak sistem yöneticilerinin ağlarını ve sistemlerini korumak ve sürdürebilmek amacı ile kullanılabilir.

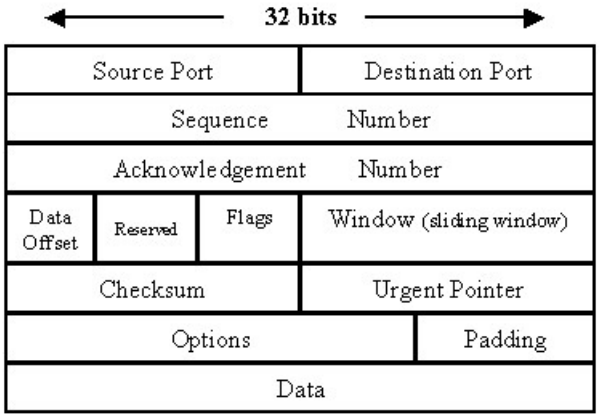
Anahtarlanmamış bir ağda, amaçlanan normal senaryo, veriler ağdaki tüm makinelere yayınlanır. Her ağ arabirim kartı pakete bakar ve hedef değil ise paketi atar, aksi takdirde paketi işleme sokar.

3.2. Paket Yapısı ve İçeriği

IP paketleri olarak tanımlanan koklanmış veriler daha da ayrıştırılır. Başlıkları, her bağlantı için benzersiz bir küme çıkarmak için kullanılır. Bu set, kaynak IP adresini, hedef IP adresini, protokolü, kaynak bağlantı noktasını ve hedef bağlantı noktası şeklindedir.



Şekil 3.2. IP paket yapısı



Şekil 3.2.1. TCP paket

Şekil 3.2.2. UDP paket

3.3. Uygulama Alanları

Dinleme yazılımlarının birçok farklı uygulama alanlarından iki tanesi, ağ analizi ve hata ayıklama, diğeri ise izinsiz giriş tespit etme.

3.3.1 Ağ Analizi ve Hata Ayıklama

İyi bir ağ dinleme yazılımı, ağ üzerinde gerçekten neler olup bittiğini anlamak için en iyi araçtır. Ağı analiz etmenin iki düzeyi vardır, makro ve mikro seviye.

Makro düzeyde, bir ağ segmentindeki trafik toplu olarak incelebilir; uzun süreli izleme yapılabilir ve trafik miktarı, bant genişliği gibi sorunlar, gün boyunca ağ trafiğinin değişmesi, mevcut ağ protokolleri, yayın trafiği miktarı, ağ hataları ve ağın en ağır kullanıcısı öğrenilebilir.

Mikro düzeyde, bir ağ segmentinde akan tüm veri çerçeveleri yakalanır ve yakalanan veriler, algılayıcı analiz moduna getirilerek analiz edilir. Analiz modunda, her bir veri çerçevesinin içeriği görüntülenebilir.

Ağ dinleme uygulamaları ayrıca grafiksel gösterimler ve istatistikler sağlayabilir. Etkileşimdeki trafik ve sistemlerin hacmi, eş harita tarafından tanımlanır. Veriler, hızlı ve üst düzey bir etkinliği hesabı sağlar. Örneğin ayrıntılı istatistikler belirli bir protokole (FTP,HTTP, vb) atfedilen ağ trafiğinin tam yüzdesi ayrıca temin edilmektedir.

Bir uygulamada gecikmeye neden olanı belirlemek için istemci ve sunucu arasındaki konuşmanın analizi, istemci ve sunucu arasındaki görüşmenin analizi belirlemek için paket düşüşleri nedeniyle yeniden iletimin varlığı, olayların belirlenmesi TCP/IP ağ konuşmalarında donmuş pencelereler (büyük olasılıkla tampon dolu anlamına gelir her iki taraftaki durum), istenmeyen yayınların kaynağının belirlenmesi, IP çok noktaya yayın veri akışı, aşırı ICMP yönlendirmeleri, yönlendirme tablosu hatalarının belirlenmesi, ağdaki bir güvenlik ihlalinin analizi ve belirli bir ağ uygulamasının çalışma şeklinin belirlenmesi, bunların ağın analizinde kullanım örnekleri olarak düşünülebilir.

3.3.2. İzinsiz Giriş Tespiti

İzinsiz Giriş Tespit Sistemi (IDS), bir saldırganın izinsiz girmesini veya bir kullanıcının sistem kaynaklarını kötüye kullanmasını tespit etmeye çalışır.

İzinsiz giriş tespitinin birincil varsayımları şunlardır: kullanıcı ve program etkinlikleri gözlemlenebilir ve daha da önemlisi, normal ve izinsiz giriş

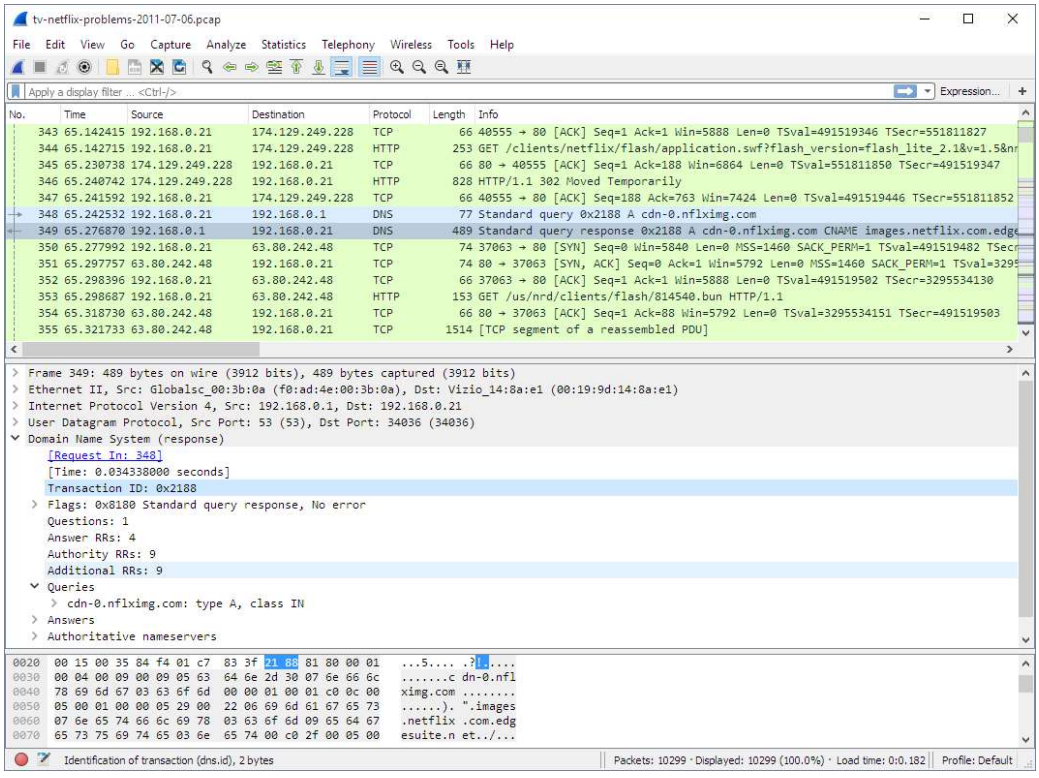
aktivitelerindeki farklı davranış hareketliliği. Bu nedenle, izinsiz giriş tespiti aşağıdaki temel unsurları içerir:

- Korunacak kaynaklar (Kullanıcı hesapları, ağ hizmetleri, işletim sistemi çekirdekleri vb.)
- Bu kaynakları içeren faalietlerin normal veya meşru davranışını karakterize eden modeller.
- Gözlemlenen faaliyetleri yerleşik modellerle karşılaştıran teknikler.

3.4. Ağ Analiz Programları

3.4.1 Wireshark

Wireshark bir ağda hareket eden veri paketlerini yakalayan ve görüntüleyen açık kaynaklı ücretsiz bir uygulamadır. Her paketin içeriğini ayrıntılı bir şekilde inceleme imkanı sunduğundan ağ sorunlarını tespit etmek ve gidermek için yaygın olarak kullanılır. Çapraz platform bir program olarak Windows, macOS, Linux ve UNIX işletim sistemlerinde kullanılabilir. Ağ uzmanları, güvenlik uzmanları, geliştiriciler ve eğitimciler bu programı sıklıkla kullanır. Açık kaynak olarak serbestçe kullanılabilir.



Şekil 3.4.1. Wireshark GUI

3.4.2. tcpdump

Tcpdump Linux/UNIX sistemlerde de-fakto paket yakalama ve analiz aracıdır.

Tcpdump pcap paket yakalama kütüphanesini(libpcap) kullanır ve ağ birimlerinden geçen paketleri (TCP/IP protokollerini) kaydedip, pcap destekli herhangi bir araç kullanarak kaydedilmiştir. Paketleri okuma işine yarar.
Özellikle ağ üzerinden yakaladığı paketleri pcap formatındaki sniffer araçlarının okuyabileceği formatta kaydetme özelliği, yoğun trafiğe sahip ağlarda sorunsuz paket yakalama becerisi tcpdump'ı ağ güvenliği yöneticilerinin vazgeçilmezi kılmaktadır.

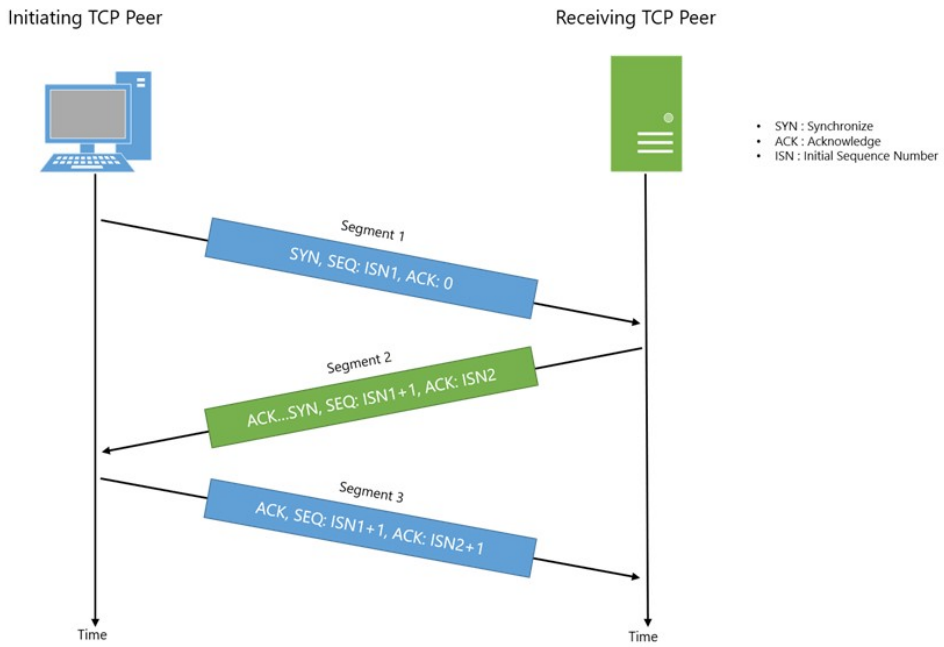
3.5. Protokollerin Analizi

Bilgisayar Ağlarında ağ tehditlerini ve güvenlik sorunlarını izlemek her zaman için zordur. Ancak, iyi bir ağ mimraisi ve ağ trafiği bilgisi ağdaki trafik sıkışıklığını, bant genişliği ve güvenlikle ilgili birçok sorunu çözebilir. Bu bölümde, ağ iletişiminde yaygın olarak kullanılan bazı protokoller analiz edildi.

3.5.1. TCP

İletim Kontrol Protokolü, iki kişi arasında güvenilir iletişim sağlar, sonra erer ve güvenilir veri dağıtımını sağlar. Bu taşıma Protokolü, verilerin sıralanmasını ve hata kontrolünü destekleyen OSI modelidir. Bu protokol paket bilgilerini düşürmeden veya yerleştirmeden verileri hedefe iletir. TCP iletişimi, kaynak ve hedef bağlantı noktaları kullanılarak kurulur. 1-1023 arası geçici portlar, 1024-65535 arası işletim sistemlerinin benzersiz iletişim için seçebileceği standart portlar vardır.
TCP iletişimi üç yönlü el sıkışma ile başlar. Örneğin, bilgisayar X diğer bilgisayar Y ile iletişim kurmak istiyorsa, öncelikle X bilgisayarı içinde veri olmayan, sadece ilk sıra numarası ve maksimum segment boyutu bulunan SYN bayrak paketini yollar.

Y bilgisayarı bu pakete SYN ve ACK bayrak setleri ile cevap verir. Son olarak X bilgisayarı ACK bayrak paketini yollar ve bu işlemler sorunsuz işlerse, iki bilgisayar iletişim kurmaya başlar. Ancak TCP iletişim oturumu FIN bayraklarını kullanarak 4 paket ile biter.

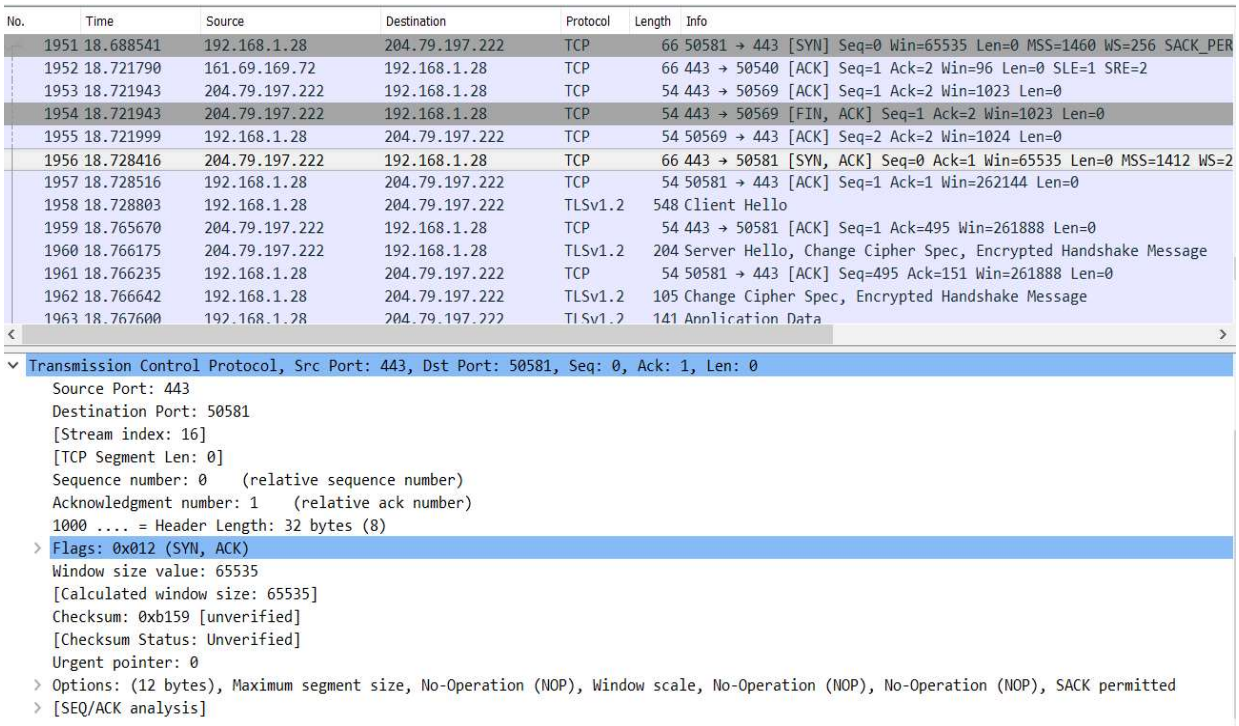


Şekil 3.5.1. TCP üç yönlü el sıkışma

TCP paketinin içeriği:

- **Kaynak port:** Gönderenin cihazı tarafından paketi iletmek için kullanılan bağlantı noktası.
- **Hedef port:** Paketin iletildiği bağlantı noktası.
- **Sıra numarası:** TCP, bir oturum sırasında her segmenti izlemek için sıra numaralarını kullanır. Bu alan, güvenilir bir TCP oturumu için önemlidir. Bu değer, her bir segmenti takip eder ve herhangi bir verinin eksik olmamasını sağlar.
- **Acknowledgement numarası:** Bu alan, bir sonraki paketteki sıra numarası olarak beklenecek iletişim değerini sağlar.

- **Bayraklar:** Bu alan, TCP paketinin türünü tanımlamak için işaretler olarak değer sağlar. Kullanılan işaretlerin örnekleri URG, ACK, PSH, RST, SYN ve FIN'dır.
- **Pencere boyutu:** Bu alan, TCP alıcı arabelliğinin boyutu için değer sağlar.
- **Checksum:** Bu alan, TCP başlığı ve verilerinin içeriğinin güvenli şekilde varışını belirtir.
- **Acil İşareti:** Bu alan, URG bayrağı ayarlandığında etkindir ve bir paketteki veri okuma noktası hakkında CPU için bilgi sağlar.



Şekil 3.5.1.1. TCP başlık yapısı

3.5.2. IP

İnternet Protokolü, OSI modelinin ağlar arası iletişimine izin veren 3. katman protokolüdür. IP'nin iki sürümü vardır IPv4 ve IPv6 dır. Bu protokol ağdaki konum ve karmaşıklıktan bağımsız olarak cihazlar arası veri taşıyabilir. IP adresleri 32 bitlik benzersiz adreslerle tanımlanır. Örneğin, 192.168.5.53, bir IP adresidir. 3. katmanda,

internet protokolü bilgileri, 4. katmandaki taşıma protokolleri kullanılarak hedefe teslim edilen paket şeklindeki verileri bağlar. IPv4 başlığı, bir paketle ilgili aşağıdaki alanlara sahiptir:

- **Versiyon:** Bu alan kullanılan IP protokolünün versiyonunu belirtir. (IPv4,IPv6)
- **Başlık Uzunluğu:** Bu alan IP başlığının uzunluğunu belirtir. (20 byte)
- **Servis tipi:** Bu alan, trafiğin öncelik seviyesini hizmet bayraklarına göre belirler. Bu değer, yönlendiriciler tarafından trafikle ilgilenmek için kabul edilir.
- **Toplam uzunluk:** Bu alan, IP paketinde mevcut olan başlığının uzunluğu ile verilerin toplamıdır.
- **Kimlik:** Bu alan, IP paketi için pakete verilen benzersiz kimlik numarasıdır.
- **Bayraklar:** Bu alan, paketin parçalanmış paketlerin parçası olup olmadığını tanımlar.
- **Parça ofseti:** Bu alan, parçalanmış bir paketin yeniden birleştirilmesini sağlar.
- **Yaşama Zamanı:** Bu alan, 'yönlendiricilerin üzerinden atlama/ saniye' şeklinde ölçülen paketin ömrünü belirtir. Yaşama zamanı (TTL) her yönlendiriciden geçtiğinde 1 azalır.
- **Protokol:** Bu alan, bir sonraki gelen paketin tipini belirtir.
- **Başlık Checksum:** Bu alan, hataları bulur ve paketin zarar görmemiş olduğunu doğrular.
- **Kaynak IP adresi:** Bu alan, paketin gönderildiği kaynak adresini belirtir.
- **Hedef IP adresi:** Bu alan, paketin gönderilmek istendiği adresi belirtir.

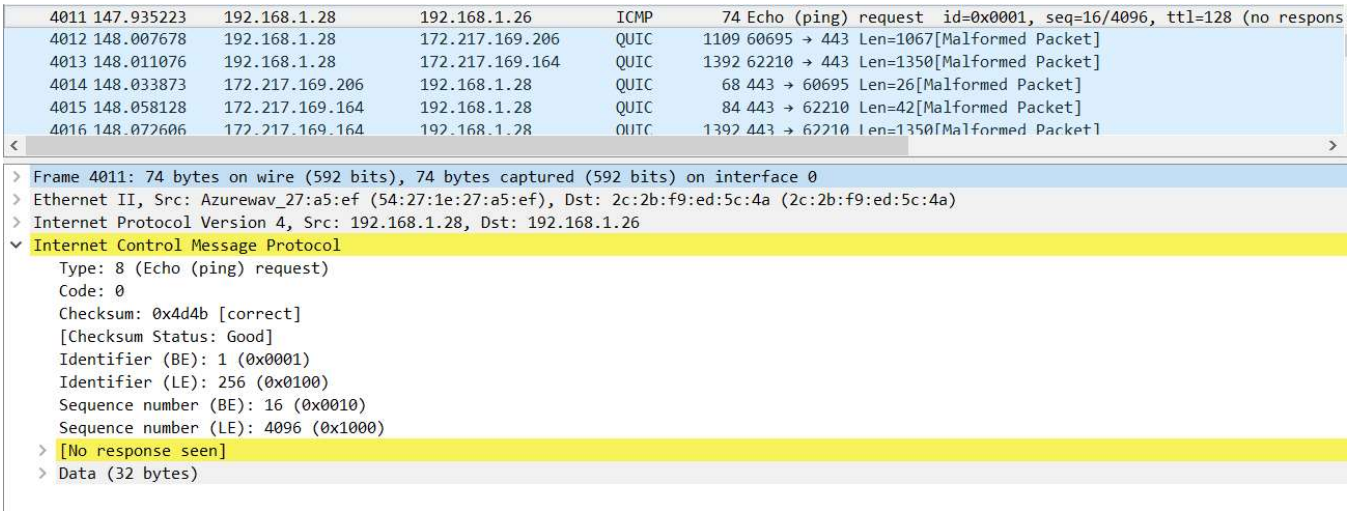
3.5.3. ICMP

İnternet Kontrol Mesajı Protokolü, ağdaki cihazların durumunu ve yolların bilgisini sağlar. Bu protokol, ağ iletişiminde sorun gidermek açısından önemli bir özelliktir. Ping, bu protokolün özelliklerinden biridir. Genel olarak, ping komutu bir seferde bir paket gönderir ve cevap paketinin iki cihaz arasındaki bağlantıyı belirlenmesini

bekler. Bu protokol ayrıca ulaşılamayan yerlere ve bağlantı noktalarıyla ilgili sorunları çözer.

ICMP başlık yapısı:

- **Tip:** Bu alan, ICMP mesajının türü için değer verir. Örneğin, bu değer için 8 verildiği zaman ping isteği, 0 ise ping yanıtıdır.
- **Kod:** Bu alan, ICMP mesajının alt sınıflandırmasıdır.
- **Checksum:** Bu alan, ICMP başlığının içeriğini ve varsa verileri kontrol eder.



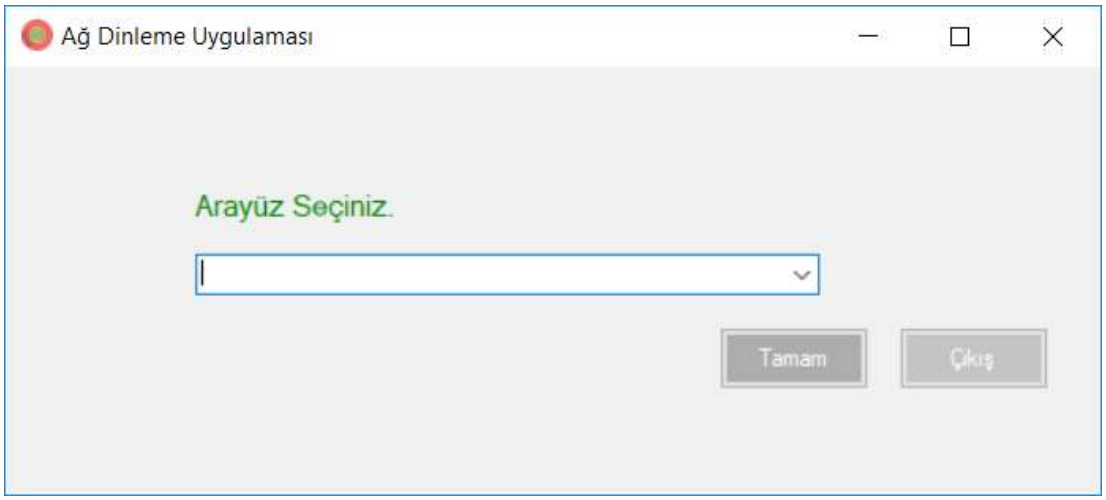
Şekil 3.5.3. ICMP başlık yapısı

BÖLÜM 4. PROJENİN GERÇEKLEMESİ

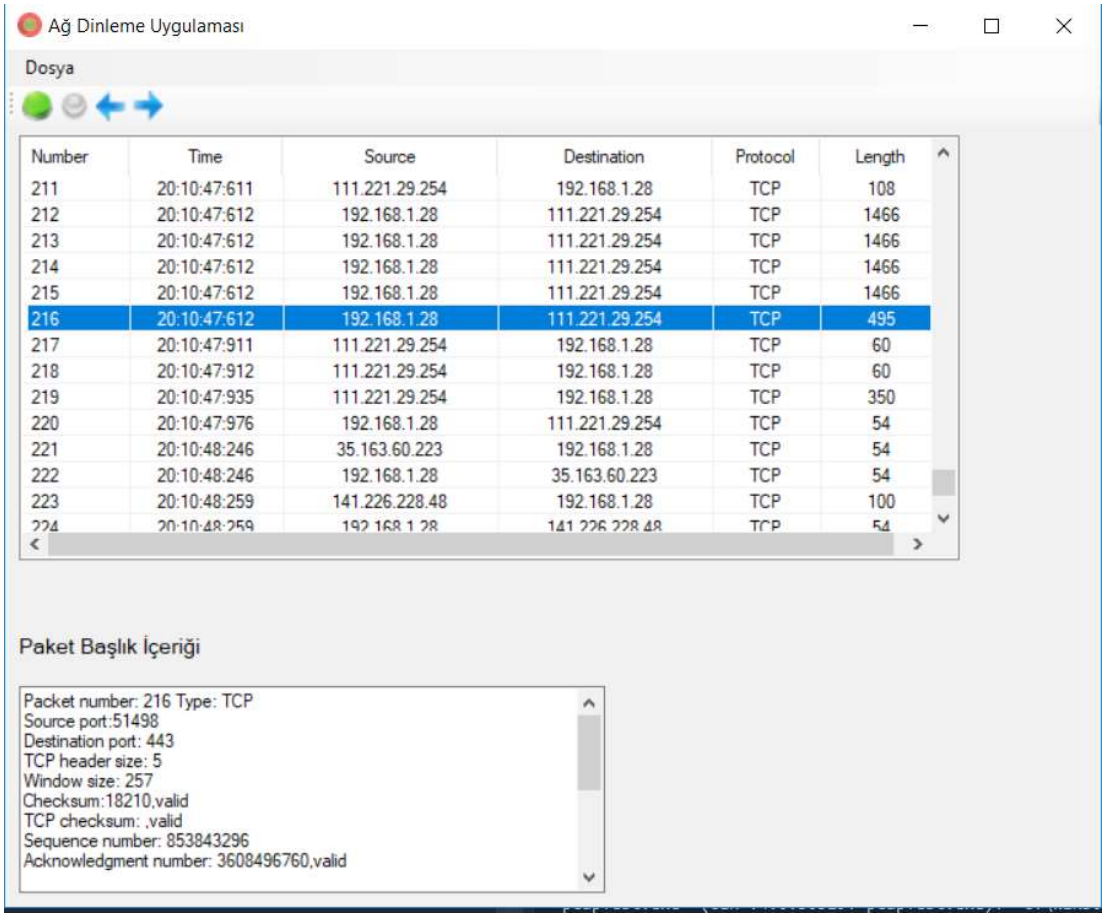
4.1. Tasarım ve Kodlar

Projeyi c# form şeklinde yapmak istedim. Çünkü arayüzü, kullanıcıların rahat bir şekilde kullanabileceği tasarıma sahip olmalıdır. Tasarımda gerektiğince sade ve kolay anlaşılır olması için özen gösterilmiştir. C# için ağ dinleme uygulaması için yaptığım araştırmada kullanabileceğim paketler buldum. PacketDotNet, Pcap.net ve SharpPcap. Bu paketlerin içinde programcının işini kolaylaştıran ve işine yarayan bir çok kütüphane bulunmakta olup, ağ dinlemede büyük yardımları bulunmaktadır.

Giriş ekranı olarak karşımıza bu ekran çıkmaktadır:



Bu ekranda dinlemek istenen bağlantı şekli seçilmektedir (WiFi, ethernet vs.)



Bu örnek için Wifi bağlantısı seçilmiştir. Yukarıda bulunan yeşil butona basıldığı vakit sniffing olayı başlamış bulunuyor. Yakalanan paketlerin Adet sırası, süresi, Kaynak IP adresi, Hedef IP adresi, kullandığı protokol ve uzunluğu ekranda belirtiliyor. İstenilen bir paketin üstüne gelindiği vakit, Aşağıda bulunan alanda paketin detaylı olarak başlık içeriği gözükmektedir. Yeşil butonun yanında bulunan kırmızı buton ile de sniffing işlemi durduruluyor.

Kod kısmına gelecek olursak:

```
private void toolStripButton1_Click(object sender, EventArgs e)// Dinleme Başlatılır
{
    if(YenidenDinleme == false) //ilk seferde
    {
        System.IO.File.Delete(Environment.CurrentDirectory + "dosya.pcap");
        wifi_device.OnPacketArrival += new PacketArrivalEventHandler(Device_OnPacketArrival);
        sniffing = new Thread(new ThreadStart(sniffing_Proccess));
        sniffing.Start();
        toolStripButton1.Enabled = false;
        toolStripButton2.Enabled = true;


    }
   else if (YenidenDinleme)
    {
        if (MessageBox.Show("Paketler dosyaya yazıldı", "Confirm", MessageBoxButtons.OK, MessageBoxIcon.Warning) == DialogResult.OK)
        {

            System.IO.File.Delete(Environment.CurrentDirectory + "dosya.pcap");
            listView1.Items.Clear();
            capturedPackets_list.Clear();
            packetNumber = 1;
            textBox2.Text = "";
            wifi_device.OnPacketArrival += new PacketArrivalEventHandler(Device_OnPacketArrival);
            sniffing = new Thread(new ThreadStart(sniffing_Proccess));
            sniffing.Start();
            toolStripButton1.Enabled = false;
            toolStripButton2.Enabled = true;

        }
    }
    YenidenDinleme = true;
}
```

Dinleme işlemini yapan en kritik kod parçası olmakla beraber yapılan iş bir thread oluşturarak dinleme olayını işleme sokmamıza yaramaktadır.

```
private void listView1_ItemSelectionChanged(object sender, ListViewItemSelectionChangedEventArgs e)
{
    string protocol = e.Item.SubItems[4].Text;
    int key = Int32.Parse(e.Item.SubItems[0].Text);
    Packet packet;
    bool getPacket  = capturedPackets_list.TryGetValue(key, out packet);

        switch (protocol) {
        case "TCP":
            if(getPacket)
            {
                var tcpPacket = (TcpPacket)packet.Extract(typeof(TcpPacket));
                if (tcpPacket != null)
                {
                    int srcPort = tcpPacket.SourcePort;
                    int dstPort = tcpPacket.DestinationPort;
                    var checksum = tcpPacket.Checksum;

                    textBox2.Text = "";
                    textBox2.Text = "Packet number: " + key +
                                    " Type: TCP" +
                                    "\r\nSource port:" + srcPort +
                                    "\r\nDestination port: " + dstPort +
                                    "\r\nTCP header size: " + tcpPacket.DataOffset +
                                    "\r\nWindow size: " + tcpPacket.WindowSize +
                                    "\r\nChecksum:" + checksum.ToString() + (tcpPacket.ValidChecksum ? ",valid" : ",invalid") +
                                    "\r\nTCP checksum: " + (tcpPacket.ValidTCPChecksum ? ",valid" : ",invalid") +
                                    "\r\nSequence number: " + tcpPacket.SequenceNumber.ToString() +
                                    "\r\nAcknowledgment number: " + tcpPacket.AcknowledgmentNumber + (tcpPacket.Ack ? ",valid" : ",invalid") +

                                    "\r\nUrgent pointer: " + (tcpPacket.Urg ? "valid" : "invalid") +
                                    "\r\nACK flag: " + (tcpPacket.Ack ? "1" : "0") +
                                    "\r\nPSH flag: " + (tcpPacket.Psh ? "1" : "0") +
                                    "\r\nRST flag: " + (tcpPacket.Rst ? "1" : "0") +

                                    "\r\nSYN flag: " + (tcpPacket.Syn ? "1" : "0") +


                                    "\r\nFIN flag: " + (tcpPacket.Fin ? "1" : "0") +
                                    "\r\nECN flag: " + (tcpPacket.ECN ? "1" : "0") +
                                    "\r\nCWR flag: " + (tcpPacket.CWR ? "1" : "0") +
                                    "\r\nNS flag: " + (tcpPacket.NS ? "1" : "0");
                }
            }
            break;
```

Dinleme işleminden elde edilen paketlerden birine tıklandığı vakit bize o paketin başlık yapısını detaylı bir şekilde göstermektedir. Burada da seçilen paketin protokolü TCP olduğu durumda, bu dökümanın içinde de bahsettiğimiz (TCP başlık yapısı) gibi gerekli bilgileri atamaktadır.

```
case "UDP":
    if (getPacket)
    {
        var udpPacket = (UdpPacket)packet.Extract(typeof(UdpPacket));
        if (udpPacket != null)
        {
            int srcPort = udpPacket.SourcePort;
            int dstPort = udpPacket.DestinationPort;
            var checksum = udpPacket.Checksum;

            textBox2.Text = "";
            textBox2.Text = "Packet number: " + key +
                            " Type: UDP" +
                            "\r\nSource port:" + srcPort +
                            "\r\nDestination port: " + dstPort +
                            "\r\nChecksum:" + checksum.ToString() + " valid: " + udpPacket.ValidChecksum +
                            "\r\nValid UDP checksum: " + udpPacket.ValidUDPChecksum;
        }
    }
    break;
```

Yukarda belirttiğim gibi bu seçilenin UDP, ARP, ICMP veya IGMP olduğu takdirde gerekli bilgilerle header içeriğini bize göstermektedir.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Bu tezde genel olarak ağ sistemi araştırılıp bilgi sahibi olundu. Gerek paket yapıları, gerekse kullanılan protokoller öğrenildi. Bilgisayar ağları hakkında edinilen bilgi kullanılarak, sniffing yapılan bir ağda, bize verdiği çıktıyı anlayıp değerlendirecek seviyeye gelindi. Piyasadaki Wireshark veya tcpdump gibi hazır ağ dinleme programları test edilerek ne tür çıktı verildiği gözlemlendi. Benzer sistem c# kullanılarak tasarlandı.

Bu gelinen noktadan daha ileriye gitmek adına zararlı yazılım tespit edilmesi üzerine uğraşılabilir.

KAYNAKLAR

[1] Charles University in Prague Faculty of Mathematics and Physics BACHELOR THESIS J´an Vesel´y Passive connection monitoring on IP based networks. Institute of Formal and Applied Linguistics 2009

[2] https://stackoverflow.com

[3] I. J. Computer Network and Information Security, 2018, 7, 12-22 Ethical Network Surveillance using Packet Sniffing Tools: A Comparative Study

[4] https://www.bgasecurity.com/

[5] BACHELOR'S THESIS | ABSTRACT TURKU UNIVERSITY OF APPLIED SCIENCES Degree Programme In Internet Techonology 2012| 42 Instructor: Patric Granholm

ÖZGEÇMİŞ

Mehmet Ali Demir, 26.11.1996 da İstanbul'da doğdu. İlk ve orta eğitimini Bostancı'da tamamladı. Lise eğitimini Ataşehir ilçesinde bulunan Dr. Nureddin ERK Anadolu Meslek Lisesi'nde tamamladı. 2014 yılında Lise'den mezun olup Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı. 2016 yılında BGA Security firmasında yazılım stajını yapmıştır. 2018 yılında İspanya ülkesinde University of Castilla-La Mancha'da eğitim gördü.

BSM 498 BİTİRME ÇALIŞMASI
DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU :
ÖĞRENCİLER (Öğrenci No/AD/SOYAD):

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: