F.Ü.MÜHENDİSLİK FAKÜLTESİ

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Labaratuvarı

DENEY NO 1

VERİ SIKIŞTIRMA YÖNTEMLERİ VE HUFFMAN KODLAMA İLE VERİ SIKIŞTIRMA

1. Deney Amacı

Veri sıkıştırma sadece bilgisayar bilimlerinde uygulama alanı olmayıp değişik disiplinler içinde, kullanılan haberleşme kanalı üzerindeki veri iletim miktarını arttırmak için kullanılmıştır. Bu deneyde bilgisayar ve haberleşme dünyasında sıklıkla kullanılan veri sıkıştırma yöntemleri ve tipleri tanıtılıp, Huffman kodlama kullanan bir sıkıştırma uygulaması yapılacaktır.

2. Veri Sıkıştırma Yöntemleri

Veri Sıkıştırma tanım olarak, bir bilginin orijinal halinden daha az yer kaplayacak şekilde kodlanması olarak tanımlanabilir. Bir iletim kanalından iletilecekse genelde kodlama, diğer durumlarda veri sıkıştırma olarak adlandırlır.

Veri sıkıştırılmada temel prensip, sıkıştırılacak dosya içinde gereksiz (unnecessary) ya da tekrar eden kısımların (redundancy) bulunmasıdır. Bu tür kısımların olmadığı bir dosyayı sıkıştırmak bilinen yöntemlerle mümkün olmayacaktır.

Veri Sıkıştırma çeşitli kaynaklara göre farklı kategorize edilmesine rağmen genel olarak kayıplı ve kayıpsız olmasına göre iki grupta incelenebilir. Kayıplı ve kayıpsız sıkıştırma yöntemlerinin ne demek olduğu deney föyünün ilerleyen sayfalarında açıklanmıştır. Farklı şekilde uygulama alanlarına (resim, ses, video vb) göre de kategorize edilebilir. [1] e sıkıştırmada kullanılan yöntemlerin benzerliğine göre kategorilerde inceleniştir.

Kayıplı sıkıştırmaların bir kısmı [4]

Jpeg, MPEG-2 (DVD), MPEG-4 (DivX, Xvid..) , H.265, MPEG-1 VCD, H.264 Blu-ray, HD DVD, MP3, Vorbis, JPEG 2000

Kayıpsız sıkıştırmaların bir kısmı [4]

Lempel-Ziv (LZ) , Lempel-Ziv–Renau LZR (ZIP), DEFLATE (PKZIP, Gzip, PNG), LZW (Lempel–Ziv– Welch) GIF için, document compression standard DjVu, Aritmetik Kodlama WMA 9 Lossless, FLAC (free lossless audıo codec), ALAC Apple, DVD-Audio, Dolby TrueHD, JPEG 2000

Kullanılan Yöntemlere ve Uygulama Alanlarına Göre Kategorileri[1]

- İstatiksel Yöntemle
  - o Huffman Coding
  - o Facsimile Compression
  - o Arithmetic Coding

- o Adaptive Arithmetic Coding
- Sözlük Kullanan Yöntemlero LZ77 (Sliding Window) Lempel-Ziv
  - o LZSS
  - o LZ78
  - o LZW Lempel–Ziv–Welch
- Resim Sıkıştırma
  - o Progressive Image Compression
  - o JPEG
  - o JPEG-LS
- Wavelet (dalgacık) yöntemleri
  - o Averaging and Differencing
  - o The Haar Transform o Subband Transform o Filter Banks
  - o DWT
  - o The Daubechies Wavelets
  - o SPIHT
- Video Sıkıştırma
  - o MPEG-2
- Ses Sıkıştırma
  - o MPEG-1 Audio

Veri sıkıştırma yöntemlerini tanıtmaya başlamadan önce bu alanda sıklıkla kullanılan bazı terimler ve anlamları üzerinde durulacaktır.

Kayıplı Kayıpsız Sıkıştırma, Bazı sıkıştırma yöntemleri kayıplıdır. Bu şekilde bazı bilgiler kaybedilerek daha iyi sıkıştırma elde edilir. Bu tür yöntemlerde sıkıştırılan veri, tekrardan açıldığında orijinal veri ile aynısı elde edilmez. Bu şekilde bir sıkıştırma yöntemi ancak resim, video ve ses verileri üzerinde uygulandığında anlam ifade etmektedir. Eğer kayıp az ise kullanıcı tarafından fark edilmeyecektir. Buna karşılık bilgisayar dosyalarındaki 1 bitlik bir bilgi kaybı dahi o dosyayı kullanılamaz hale getirebilir. Bu tür dosyaların sıkıştırılmasının o yüzden kayıpsız olması gereklidir. Sıkıştırılmış veriler açıldığında, orijinal hale dönüldüğünde bu tür yöntemlere kayıpsız sıkıştırma yöntemleri denir. Metin dosyaları sıkıştırılırken iki konuya dikkat edilmelidir. 1 sıkıştırılacak veri, bir programlama dilinin kaynak koduna ait ise içindeki boşluklar zaten derleyici tarafından ihmal edileceği için, bu kısımlar sıkıştırmada kullanılabilirler. 2 bir kelime işlem programının çıktısı metin belgesi olarak kaydedilmek istenirse, font bilgisi gibi bilgiler ihmal edilebilir.

Sıkıştırıcı kodlayıcı, giriş olarak tekrarın çok olduğu (redundency) veri alınıp düşük tekrarlı (low redundency) dosya oluşturan koddur. Kodlama anlamı çok genel olmasına rağmen burada veri sıkıştırma olarak kullanılacaktır.

Adaptif olmayan sıkıştırıcı, bu yöntem verileri sıkıştırırken kullandığı tablo parametrelerini ya da metodunu sıkıştırılacak veriye göre değiştirmezler. Bazı sıkıştırma algoritmaları ham veriyi inceleyip çalışmasını ona göre değiştiren yöntemlere adaptif sıkıştırıcı yöntemler denir. Huffman kodlama bu tip bir yöntemdir. Bazı sıkıştırma algoritmaları iki fazlıdır. İlk fazda sıkıştırılacak veri hakkında istatistiksel bilgi toplanır, diğer fazında bu verilere elde edilen parametre ve kodlara bağlı olarak sıkıştırma gerçekleştirilir. Bu yöntemlere yarı Adaptif yöntemler olarak ifade edilir.

Simetrik sıkıştırma yöntemlerinde, sıkıştırma ve sıkışmış verinin açılması aynı temel algoritmayı

<zıt> yönlerde çalıştıran yöntemlerdir.

Sıkıştırma performansı için değişik büyüklükler kullanılır. Bu faktörler sırası ile anlatılacaktır:

1. En sık kullanılanı sıkıştırma oranı (compression ratio) dur denklem 1 deki gibi ifade edilir.

$$\text{Sıkıştırma Oranı}=\frac{\text{Çıktı dosyasının boyutu}}{\text{Girdi dosyasının boyutu}} \quad (1)$$

örneğin 0.6 değerinin anlamı sıkıştırmadan sonra veri orijinal verinin %60 ı kadar yer kaplıyor olacaktır. Aynı şekilde 1 den büyük değerler sıkıştırılan dosya büyüklüğünün orijinal veriden daha fazla olacağı anlamındad olup negatif sıkıştırma olarak adlandırılır. Sıkıştırma oranı bpb (bit per bit) birimle de ifade edilebilir. Resim sıkıştırmada ise benzer bir birim kullanılır bpp (bits per pixel). Metin dosyaları için bu ifade bpc olur (bits per character: Bir karakteri sıkıştırmak için ortalama gerekli bit miktarı).

Sıkıştırma oranı ile iki terimden de söz edilmesi gereklidir. Bunlardan ilki bitrate bpb ve bpc için genel bir terimdir. Bundan dolayı veri sıkıştırmadaki temel hedef girilen herhangi bir veriyi düşük bitrate'lerde ifade etmektedir. Bit budget sıkıştırılmış bir dosyada bir bitin görevini ifade eder.

2. Sıkıştıra oranının tersi ise Sıkıştırma Faktörü (compression Factor)

$$\text{Sıkıştırma Faktörü}=\frac{\text{Girdi dosyasının boyutu}}{\text{Çıktı dosyasının boyutu}} \quad (2)$$

Bu durumda 1 den büyük değerler sıkıştırmayı, küçük değerler genişlemeye işaret edecektir.

3. 100x(1-compression ratio) da anlamlı bir ölçüm performans göstergesidir. 60 değeri çıkış dosyasının orijinal dosyanın %40 ı kadar yer kapladığı anlamındadır. Yada sıkıştırma ile %60 lık tasarruf edilmiştir.
4. Resim sıkıştırmada bpp sıkılıkla kullanılmaktadır. Bu Bir pikseli sıkıştırmak için ortalama gerekli olan bit miktarını vermektedir.

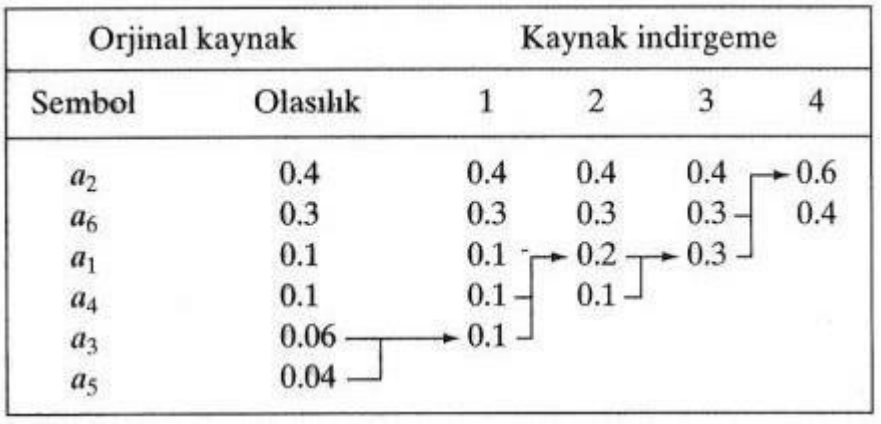
Olasılık modeli, istatistiksel veri sıkıştırma metotlarında önemli bir kavramdır. Bazen sıkıştırma algoritması iki kısımdan oluşmaktadır probability model ve sıkıştırmanın kendisini ifade etmektedir.

Entropi 1948 yılında Claude Shannon Tarafından Bell Laboratuarında Informasyon Teorisi oluşturulmuştur. Veri sıkıştırmayı anlamak için bilinmesi gereken en önemli enformasyon teorisi kavramı entropidir. Olasığı P olan bir a sembolünün entropisi $-P\log_2 P$ olarak ifade edilir. Örneğin olasılığı 0.5 olan a sembolünün entropisi 0.5 çıkacaktır. a1 den an'e kadar tüm sembollerin Entropisi ise, P1 den Pn e o sembollerin olasılıkları olacak şekilde ifade edilirse $\sum_n -P_i \log_2 P_i$ eşitliği ile bulunur. Çoğunlukla bir sistemdeki rastgelelik ve düzensizlik olarak tanımlanan bu istatistikten teknolojide birçok alanda yararlanılır.

3. Huffman Kodlama

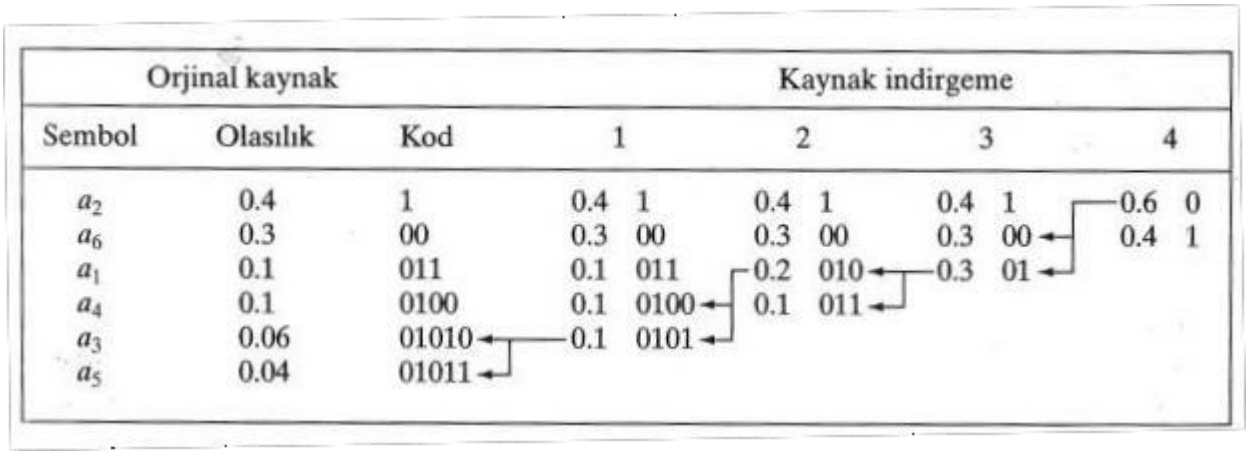
Bilgi kaynağını içindeki sembolleri kodlarken Huffman kodlama kaynak başına en küçük sayıdaki kod sembolü üretecektir [3]. Huffman kodlamanın ilk aşamasında var olan sembollerin olasılıkları bulunarak sıralanır. Bu

semboller arasında en düşük olasılığa sahip olan ikisi kodlanmak üzere yeni bir sembolde birleştirilir. Sonradan oluşan indirgenmiş kolar tekrardan sıralanır ve en düşük olasılığa sahip ikisi toplanır. Bu işlem toplanan olasılıklar sonucu 1 olana kadar devam edilir. Şekil 1 de örnek verilen 6 sembol için olasılıkları en büyük olanı en üstte olacak şekilde en solda sıralanmıştır. En düşük olasılığa sahip a5 ve a3 sembolleri olasılıkları toplanıp indirgenmiş birleşik sembol elde edilir.



Şekil 1. Huffman kaynak sembol indirgenmesi

İkinci aşama bunların kodlanması olacaktır. En küçük kaynaktan başlayarak asıl sembol kaynaklarına doğru yerleştirme yapılır. İkili sbir kodlama için kullanılacak en az uzunluk elbette ki 0 ve 1 olacaktır. Şekil 2 de bu kodlamanın nasıl uygulandığı gözükmektedir. Şekilde gözüktüğü gibi bu iki sembole Şekil 2 nin en sağında atanmıştır. 0.6 aslında bir önceki adımda ki iki olasılığın toplamıydı, 0.6 yı kodlamak için kullandığımız 0 şimdi onu oluşturan iki sembol için de ön kod olarak kullanılacak şekilde, altta kalan sembollere yeniden 0 ve 1 sembolleri verilir. Bu her bir indirgenmiş sembol için devam ederek en sonunda orijinal sembollere kadar devam edilir. Burada dikkat edilecek husus dallanmada kodlama yaparken, 0 kodunu büyük olan tarafa vermişsek diğer dallarda da aynı prensibe dikkat etmeliyiz.



Şekil 2. Huffman code atama prosedürü

4. Huffman Kod Çözümlenmesi

Kod doğru bir şekilde üretildikten sonra kodun çözülmesi bir tablo bakma sayesinde hatasız bir şekilde gerçekleştirilecektir. Block Code dur çünkü, her kaynak kod belirli bir kod serisine eşleştirilmiştir. Kodlar soldan sağa doğru yaklaşımla çözülmelidir. Şekil 2 de elde edilen Huffman Kodları ile aşağıdaki kodlanmış veri dikkate alındığında

010100111100

Kod çözülmüş hali aşağıdaki gibi olacaktır

a3a1a2a2a6

Referanslar

[1] web sitesi: http://ww3.ticaret.edu.tr/mckasapbasi/files/2015/09/Compression-Huffman-.pdf

[2] David Salomon “A Guide To Data Compression Methods” 2001 Springer

[3] Entropi http://tr.wikipedia.org/wiki/Entropi

[4] Rafael C. Gonzalez, Richard E. Woods Digital Image Processing 2008 (third Edition) Pearson International

[5] Data Compression http://en.wikipedia.org/wiki/Data_compression

F.Ü. MÜH. FAK.

BİLGİSAYAR MÜH. BÖL.
BİLGİSAYAR
SİSTEMLERİ LAB.

DENEY NO : 2

BİLGİSAYAR AĞ CİHAZLARININ YAPILANDIRILMASI

1. GİRİŞ

Bu deneyin amacı Cisco 1760 Router ve Cisco Catalyst 2950 switch kullanarak:

- Router ve Switch üzerindeki arabirimleri ve portları tanımak, işlevlerini anlamak
- Router ve Switch konfigürasyonlarının nasıl yapılacağını öğrenmek
- Yerel Alan Ağı (LAN, Local Area Network) oluşturmak
- Statik, default ve dinamik yönlendirme kavramlarını öğrenmek
- RIP, IGRP, EIGRP gibi yönlendirme algoritmalarını öğrenmek
- Bu algoritmaları routerlar üzerinde yapılandırmaktır.

Deneye gelmeden önce aşağıdaki konuların ve özellikle yönlendirme algoritmaları, protokolleri ve metriklerinin kesinlikle bilinmesi gerekmektedir.

- Ağ arabirim kartı (Network interface card)
- IP adresleri ve Ağ maskeleri
- Ağ topolojisi
- Ağ protokolleri
- OSI modeli ve katmanları
- TCP/IP modeli
- Kablo türleri ve özellikleri
- Routing algoritmaları, metrikleri ve protokolleri (RIP, IGRP, EIGRP)
- Statik ve Dinamik yönlendirme
- Uzaklık Vektörü Algoritması (Distance Vector Routing)
- Bağlantı Durumu Algoritması (Link State Algorithm)

2. ROUTER

2.1. Router'ın Çalışması

Aynen PC'ler gibi Router'larda ilk açıldıklarında POST işlemini gerçekleştirir. Yani CPU, hafıza, arabirim devreleri gibi sistem donanımlarını kontrol eder. Tüm donanımın düzgün çalıştığından emin olunduktan sonra POST işlemi ROM'da tutulan bootstrap yazılımını çalıştırır. Bootstrap programı Flash'ta bulunan IOS'u bulur, sıkıştırmasını açar ve bu IOS'u Flash'dan RAM'e yükler. Bazı router'lar yeterli hafızaya sahip olmadıkları için IOS'u RAM'e yüklemeden doğrudan Flash'dan çalıştırırlar. Eğer router herhangi bir geçerli IOS bulamazsa RAM'daki RXBoot olarak adlandırılan mini IOS'u yükler. Eğer bu işlemde başarısız olursa ROM Monitor (ROMMON) moduna düşer. IOS yüklendikten sonra NVRAM'da bulunan başlangıç konfigürasyonlarını (startup configuration) yükler. Eğer herhangi bir sebepten ötürü konfigürasyon dosyası bulamazsa IOS, "NVRAM invalid" mesajını verir ve IOS otomatik olarak "setup dialog" olarak adlandırılan konfigürasyon işlemini başlatır. Router'ların üzerinde **IOS (Internetwork Operating System)** işletim sistemi çalışır. Bu işletim sisteminde temel olarak iki farklı komut modu vardır.

User exec
Privileged exec

Router'a bağlanıp, yönetmek için değişik seçenekler mevcuttur. Birincisi router'a doğrudan konsol portundan bağlantı yapılır. İkincisi uzaktan modem yoluyla router'ın AUX portuna bağlanılır. Üçüncü seçenek ise Router aktif olan LAN veya WAN portunda telnet aracılığı ile bağlanılır. Fakat telnet ile bağlantı kurulacak Router'ın bazı öncelikli ayarlarının yapılması (örneğin interface'lerin up duruma getirilip adreslerinin atanmış olması) gerekir. Router'a ilklogin olunduğunda user exec modda olunur. Bu modda sadece bilgi görüntülenebilir. Yani herhangi bir konfigürasyon değişikliği yapılamaz. Herhangi bir değişiklik yapılması istenilirse privileged exec modun kullanılması zorunludur. User exec moddan privileged moda geçmek için **enable** komutu kullanılır.

2.2. Hyperterminal

Ağ cihazlarını konfigüre etmek için kullanılan bir terminal emülasyon yazılımıdır. Bu program kullanılarak router'a bağlanabilmek için PC'nin herhangi bir seri portuna takılan (COM1 veya COM2) DB-9-RJ45 dönüştürücüye rollover kablo takılır. Hyperterminal programı çalıştırılır, bağlantı ismi verilir, bağlantının kurulacağı seri port seçilerek portun özelliklerinin belirlendiği pencereye uygun değerler girilerek bağlantı kurulur.

2.3. Cisco Packet Tracer

Ağ cihazlarının yapılandırılmasını, simülasyon olarak gerçekleştirmek amacıyla kullanılan platformlar arası bir görsel simülasyon aracıdır. Cisco Systems tarafından tasarlanan Cisco Packet Tracer programı, kullanıcıların ağ topolojileri oluşturmalarına ve modern bilgisayar ağlarını taklit etmelerine olanak tanır [1].

2.4. Router'ın Kurulması

Router'ın açılması sırasında router konfigürasyon dosyasını arar. Eğer herhangi bir konfigürasyon dosyası bulamazsa sistem konfigürasyon işlemi başlar. Bu işlem sırasında aşağıdaki sorulara "Yes" diye cevap verilirse router soru temelli konfigüre edilebilir.

Continue with configuration dialog? [yes/no]
Would you like to see the current interface summary? [yes/no]

Bu konfigürasyon türünde router bir takım sorular sorar ve bu soruların cevaplarını ister. Sorulan soruların varsayılan cevapları soru sonundaki köşeli parantezlerin (*+) içinde verilmiştir. Varsayılan cevapları kabul etmek için Enter'a basılmasıdır. Eğer soru cevap tabanlı konfigürasyondan çıkılmak istenirse, **Ctrl+C** tuşlarına basılması yeterlidir. Eğer yukarıda sorulan sorulara "No" diye cevap verilirse router komut satırından konfigüre edilecek demektir. Bu durumda komut satırı aşağıdaki şekildedir.

Router>

İlk karşılaşılan mod "user exec" moddur. Varsayılan olarak konfigüre edilmemiş tüm Router'ların adı Router'dır ve "privileged exec" moda geçmek için herhangi bir şifre tanımlanmamıştır. Router üzerinde herhangi bir konfigürasyon değişikliği yapmak istenilirse privileged moda geçilmesi gerekir. Bunun için komut satırına aşağıdaki komut yazılır.

Router>enable
Router#

Fast-Ethernet arabirimini konfigüre etmek için, global kanfigürasyon mod içerisindeki aşağıda verilen komutların kullanılması gerekir.

Router#configure terminal

Router(config)# interface fastethernet 0/0 (fast Ethernet arabirimi için konfigürasyon moda girme)

Router(config-if)# ip address 20.20.20.20 255.255.255.0 (Fast Ethernet arabirimi için IP adres ve alt ağ maskesi atama)

Router(config-if)# no shutdown (Fast Ethernet arabirimini aktif hale getirme)

Router(config-if)# exit (Fast Ethernet arabirimi için konfigürasyon modundan çıkma)

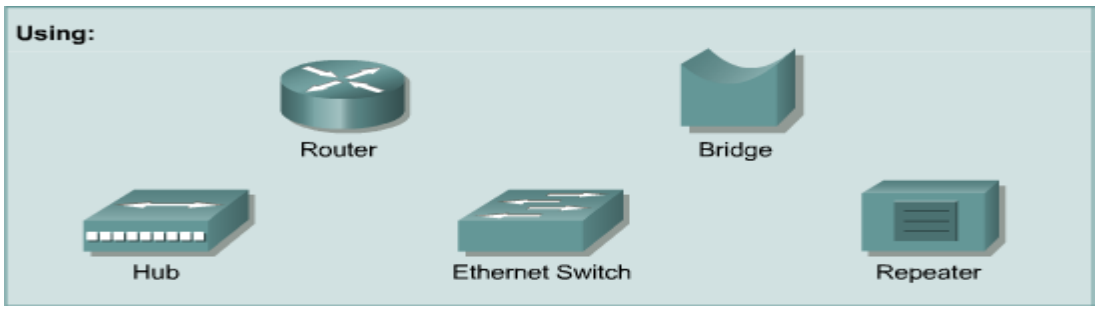
Konfigurayon ayarlarını kontrol etmek için;

Router# show running-config

3. YEREL ALAN AĞLARI

Yerel alan ağları, aynı çalışma ortamında birbirleriyle ilgili işlerde çalışan bir topluluk içinde kaynakların paylaşılması amacıyla kurulur. LAN'larda temel özellik, sistemlerin aynı ortamdaveya birbirlerine yakın mesafede olmasıdır. LAN uygulamasında kablolama alt yapısı oldukça önemlidir; kablo türü, seçilecek teknolojiyi, ağın yayılabileceği fiziksel genişliği ve portlar arasındaki iletişim hızını belirlemede baskın parametrelerdir. LAN'lar aşağıdaki bileşenlerden oluşur.

- Bilgisayarlar
- Ağ arabirim kartları
- Paralel cihazlar
- Kablolar
- Ağ cihazları



Şekil 1. Ağ cihazları.

LAN uygulamalarında Ethernet, Token Ring ve FDDI sıkça kullanılan teknolojilerdir. ETHERNET en sık kullanılan LAN teknolojisidir ve bu deneyde de ETHERNET LAN teknolojisi kullanılacaktır. Şekil 1'de LAN'larda sıkça kullanılan ağ cihazları verilmiştir.

4. YÖNLENDİRME

Bu bölümde yönlendirmenin ne olduğu tanımlanarak statik ve dinamik yönlendirmearasındaki farklar açıklanacaktır. Bir router hedef ağa paketleri sevk etmek için yönlendirmeyi kullanır. Router, paketin hedef IP (destination IP) adresine bakarak karar verir. Yol boyunca tüm cihazlar paketi doğru hedefe ulaştırmak için hedef IP adresini kullanır. Doğru kararlar alabilmek için, Routerlar hedef ağlara nasıl ulaşacaklarını öğrenmek zorundadırlar. Routerlar dinamik yönlendirmeyi kullandıkları zaman, uzak ağlara ulaşmak için gerekli yönlendirme bilgisini diğer routerlardan öğrenir. Statik yönlendirmeyi kullandıkları zaman ise, bu bilgi ağ yöneticisi tarafından yapılandırılır.

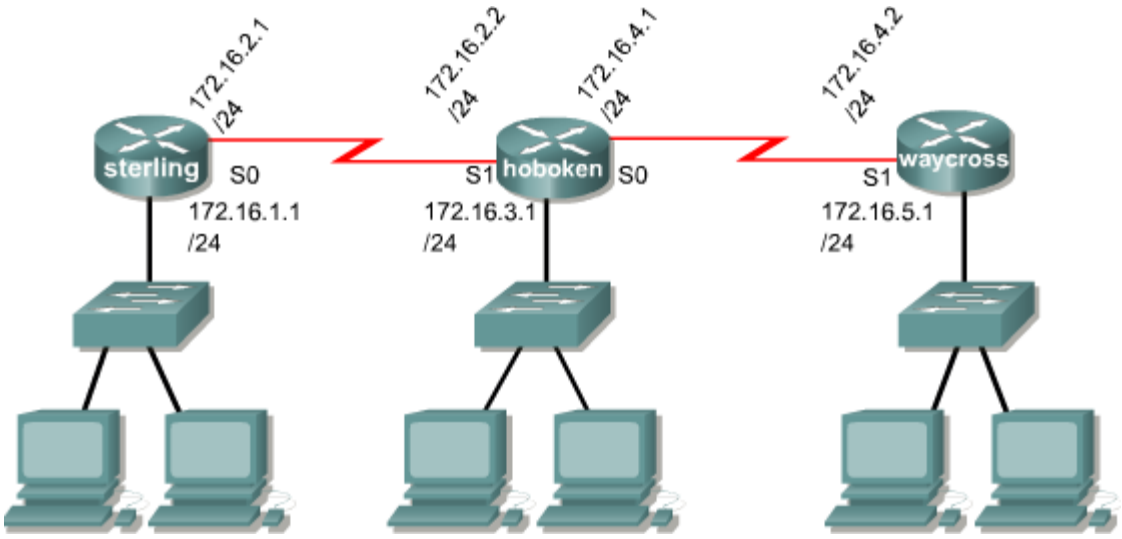
Statik yönlendirme el ile yapılandırıldığından, ağ yöneticileri herhangi bir topoloji değişikliğini yansıtmak için statik route'ları ekleyip silmek zorundadırlar. Büyük ağlarda yönlendirme tablolarını (routing tables) el ile sürdürme çok büyük yönetim zamanı gerektirir. Birkaç olası değişimin olduğu küçük ağlarda statik yönlendirme daha sık kullanılır. Statik yönlendirme ekstra yönetici ihtiyacından dolayı dinamik yönlendirme kadar kullanışlı değildir. Bazen büyük ölçekli ağlarda belirli bir amacı gerçekleştirmek için statik router'lar dinamik yönlendirme protokolleri ilebirlikte yapılandırılır [2].

4.1 Statik Yönlendirme

Bu bölümde statik yönlendirmenin nasıl yapılacağı anlatılacaktır. Statik yönlendirme üç kısım içerisine bölünebilir.

- Ağ yöneticisi yönlendirmeyi yapılandırır.
- Router yönlendirmeyi yönlendirme tablosuna kurar.
- Statik yönlendirme, paketleri yönlendirmek için kullanılır.

Bir yönetici statik yönlendirmeyi el ile yapılandırmak için **ip route** komutunu kullanır. Bu komut için doğru sözdizimi Şekil 2'deki yapı için aşağıda verilmiştir.



Şekil 2. Ağ şeması

Şekilde Hoboken routerın ağ yöneticisi diğer routerlar üzerindeki 172.16.1.0/24 ve 172.16.5.0/24ağlarına statik yönlendirmeyi yapılandırma ihtiyacı duyar. Yönetici bunu başarmak için aşağıdaki iki komutu kullanmak zorundadır.

```
Hoboken(config)#ip route 172.16.1.0 255.255.255.0 s1
                  command  destination   sub mask     gateway
                           network
Hoboken(config)#ip route 172.16.5.0 255.255.255.0 s0
                  command  destination   sub mask     gateway
                           network
```

Komut söz diziminde çıkış arabirimleri s1 ve s0 geçit yolları (gateway) ile belirtilmiştir.Aynı statik yönlendirme için diğer bir komut söz dizimi ise aşağıdaki gibidir.

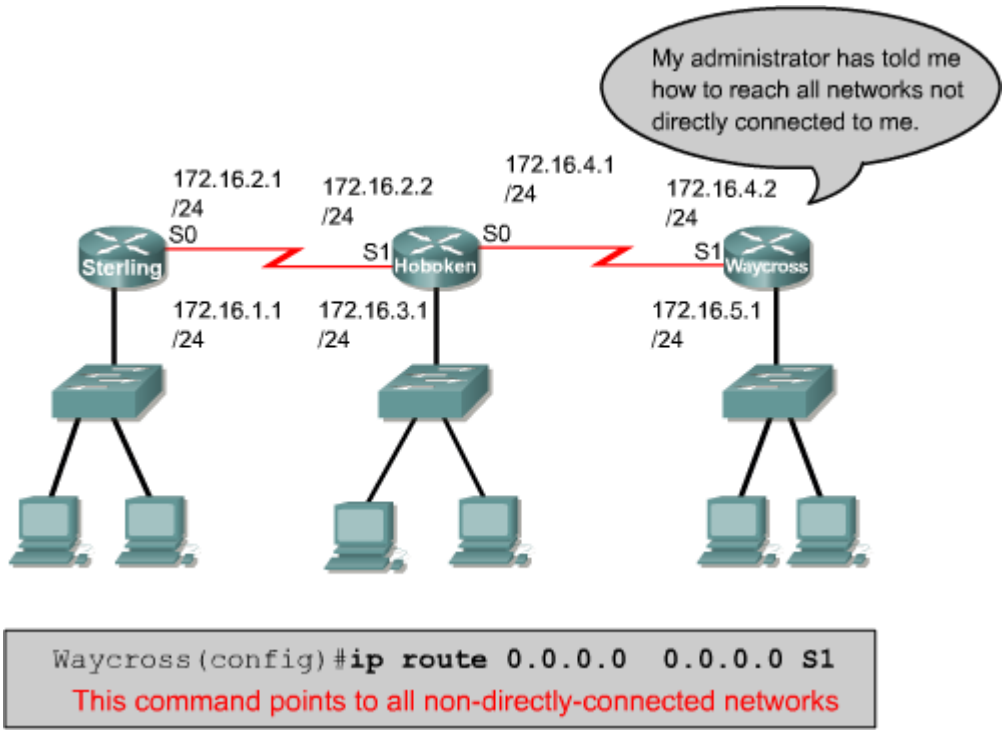
```
Hoboken(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.1
                  command  destination   sub mask     gateway
                           network
Hoboken(config)#ip route 172.16.5.0 255.255.255.0 172.16.4.2
                  command  destination   sub mask     gateway
                           network
```

4.2 Default-Statik Yönlendirme

Bu bölümde default statik yönlendirmenin nasıl yapılacağı anlatılacaktır. Default yönlendirme, hedef IP yönlendirme tablosundaki hiçbir giriş ile eşleşmediği durumda kullanılır.

Default yönlendirme söz dizimi aşağıdaki gibidir. Şekil 3, Waycross router için default yönlendirmeyi göstermektedir.

ip route 0.0.0.0 0.0.0.0 [gelecek -hop-adres | çıkış arabirimi]



Şekil 3. Default-statik yönlendirme

4.3 Dinamik Yönlendirme

Bu bölümde bir yönlendirme protokolünün nasıl yapılandırılacağı açıklanacaktır. Bir router üzerinde IP yönlendirme protokolünü yapılandırmak için global ve yönlendirme parametrelerinin yapılandırılması gerekir. Global parametreler olarak RIP, IGRP, EIGRP veya OSPF gibi yönlendirme protokollerinin seçimi gereklidir. Yönlendirme yapılandırma modundaki (global configuration mode) başlıca görev IP ağ numarasının belirtilmesidir. Bir yönlendirme protokolü aşağıdaki gibi yapılandırılır.

Command
`Router(config)#router protocol [options]`

Defines a routing protocol

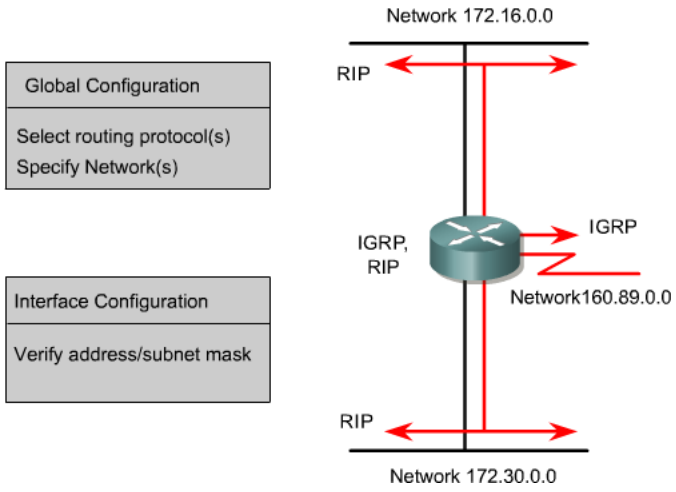
Command
`Router(config-router)#network network-number`

The network subcommand is a mandatory configuration command for each routing process

router Command	Description
protocol	IGRP, EIGRP, OSPF, or RIP
options	IGRP and EIGRP require an autonomous number. OSPF requires a process ID. RIP does not require either.

network Command	Description
network number	specifies a directly connected network

router komutu bir yönlendirme protokolünü tanımlar. **network** alt komutu ise her bir yönlendirme işlemi için zorunlu bir yapılandırma komutudur ve yönlendirme tablolarını hangi arabirimden gönderilip alınacağını belirler. Ağ numarası (network number) doğrudan bağlantılı ağı belirlemek için kullanılır. IGRP ve EIGRP yönlendirme algoritmaları bir otonom numarası (Autonomous number) gerektirirken, OSPF bir işlem ID'si gerektirir. RIP hiçbirini gerektirmez. Bir yönlendirme yapılandırma örneği Şekil 4'teki yapı için aşağıdaki gibi tanımlanır.



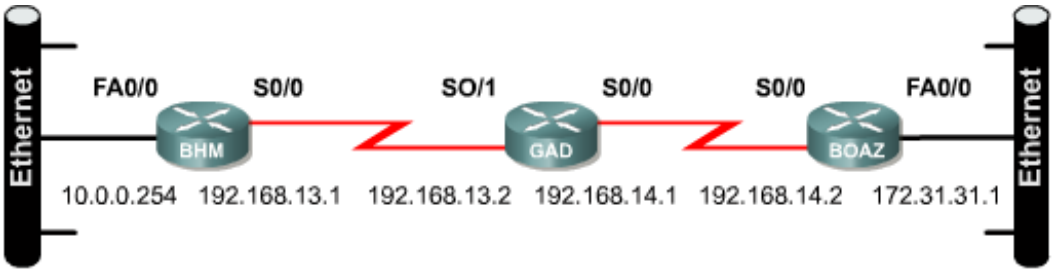
Şekil 4. Dinamik yönlendirme için örnek yapı.

GAD(config)#router rip

GAD(config-router)#network 172.16.0.0

4.3.1 RIP Yönlendirme Algoritması

Bu bölümde RIP [3] yönlendirme algoritmasının nasıl yapılandırılacağı açıklanmıştır. **router rip** komutu yönlendirme protokolü olarak RIP'i yetkilendirir. **network** komutu hangi arabirim üzerinde RIP algoritmasının çalışacağını routera söylemek için kullanılır. Yönlendirme işlemi ağ adresleri ile belirli arabirimleri ilişkilendirir ve bu arabirimler üzerinden RIP güncellemelerinin (update) alınıp gönderilmesini başlatır. RIP, yönlendirme güncelleme mesajlarını (routing- updates messages) düzenli aralıklarla gönderir. Router herhangi bir değişimi gösteren bir güncelleme aldığında, yeni güncellemeyi yansıtmak için kendi yönlendirme tablosunu günceller. Şekil 5'teki ağ için RIP dinamik yapılandırılması aşağıda verilmiştir.




```
BHM(config)#router rip
BHM(config-router)#network 10.0.0.0
BHM(config-router)#network 192.168.13.0
```

```
GAD(config)#router rip
GAD(config-router)#network 192.168.14.0
GAD(config-router)#network 192.168.13.0
```

```
BOAZ(config)#router rip
BOAZ(config-router)#network 192.168.14.0
BOAZ(config-router)#network 172.31.0.0
```

Şekil 5. RIP yönlendirme algoritması ve router üzerindeki yapılandırılması

4.3.2 IGRP Yönlendirme Algoritması

Bu bölümde IGRP yönlendirme algoritmasının nasıl yapılandırılacağı açıklanmıştır. IGRP yönlendirme işlemini yapılandırmak için **router igrp** yapılandırma komutu kullanılır. IGRP

yönlendirme işlemini sonlandırmak için bu komutun **no** formu kullanılır. Aşağıda IGRP yönlendirme algoritması için komut söz dizimi verilmiştir.

RouterA(config)#router igrp as-number

RouterA(config)#no router igrp as-number

```
RouterA(config)#router igrp 101
RouterA(config-router)#network 192.168.1.0
RouterA(config)#no router igrp 101
```

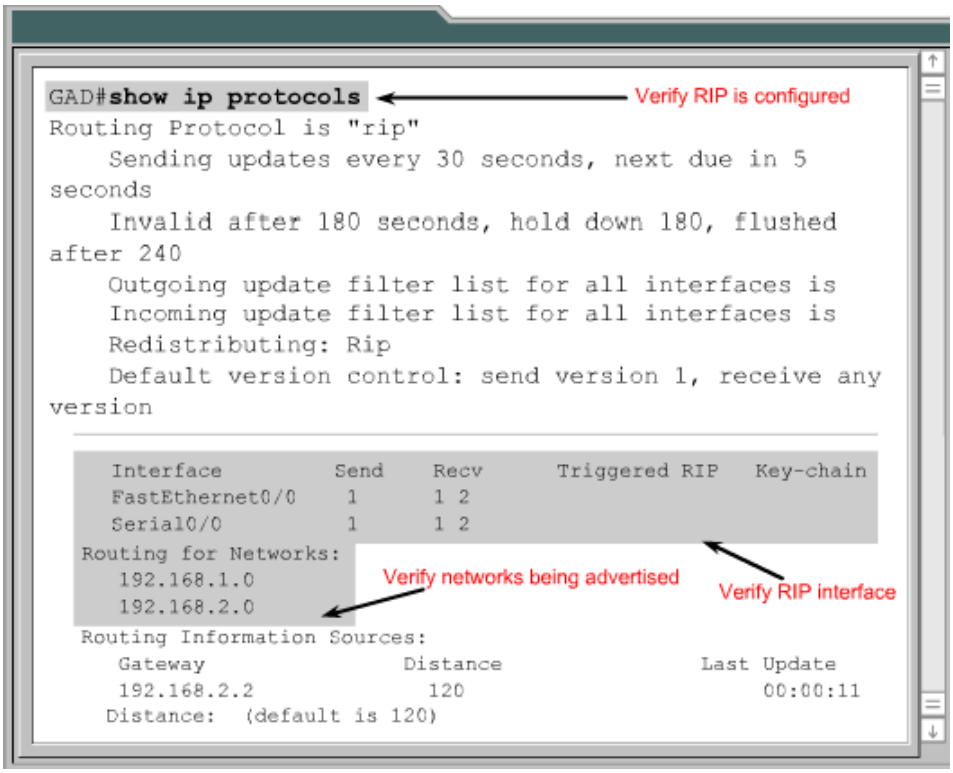
Burada AS (Autonomous system) numarası IGRP işlemini tanımlar. Aşağıda AS no 101 için IGRP'nin nasıl yapılandırılacağı gösterilmiştir.

```
RouterA(config)#router igrp 101
RouterA(config-router)#network 192.168.1.0
RouterA(config-router)#network 192.168.2.0

RouterB(config)#router igrp 101
RouterB(config-router)#network 192.168.2.0
RouterB(config-router)#network 192.168.3.0
```

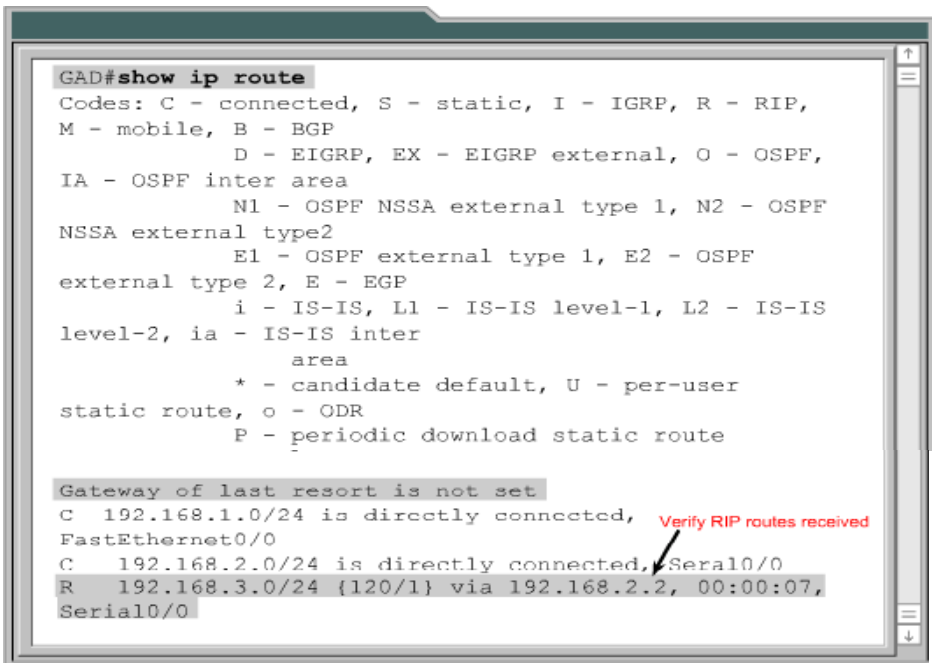
5. RIP VE IGRP YAPILANDIRMASINI DOĞRULAMA

Bu bölümde RIP, IGRP ve diğer yönlendirme işlemlerinin doğru yapılandırılıp yapılandırılmadığını öğrenmek için gerekli komutların söz dizimi ve amaçları tanımlanacaktır. İki komut **show ip route** ve **show ip protocols** mevcuttur. **show ip protocols** komutu router üzerinde IP trafiğini hangi yönlendirme protokolünün taşıdığını gösterir. Şekil 6, **show ip protocols** komutunun çıktısını göstermektedir.



Şekil 6. show ip protocols komutunun çıktısını

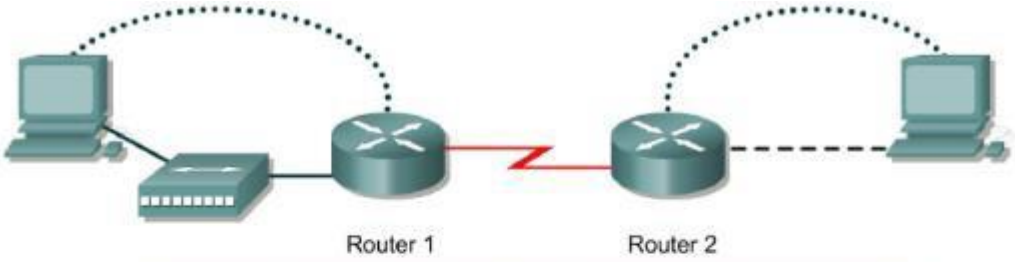
show ip route komutu yönlendirme algoritması vasıtasıyla komşu routerdan alınan yönlendirmebilgilerinin doğrulanması için kullanılır. Şekil 7'de **show ip route** komutunun çıktısı verilmiştir.



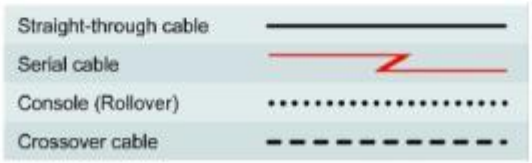
Şekil 7. show ip route komutunun çıktısı

DENEYLER

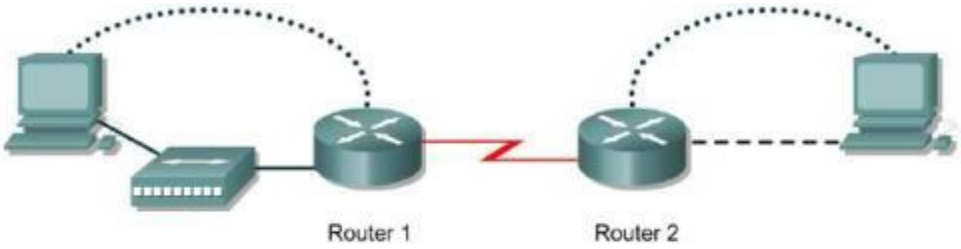
1. Aşağıdaki verilen ağ yapısını RIP yönlendirme algoritmasını kullanarak yapılandırınız?



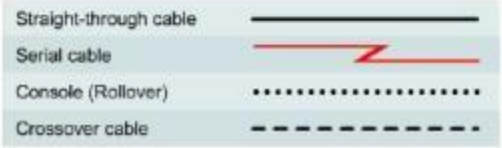
Router Designation	Router Name	Fast Ethernet 0 Address	Interface type	Serial 0 Address	Subnet mask for both interfaces
Router 1	GAD	172.16.0.1	DCE	172.17.0.1	255.255.0.0
Router 2	BHM	172.18.0.1	DTE	172.17.0.2	255.255.0.0



2. Aşağıdaki verilen ağ yapısını EIGRP yönlendirme algoritmasını kullanarak yapılandırınız?



Router Designation	Router Name	Fast Ethernet 0 Address	Interface type	Serial 0 Address	Subnet mask for both interfaces
Router 1	GAD	192.168.20.1	DCE	192.168.22.1	255.255.255.0
Router 2	BHM	192.168.25.1	DTE	192.168.22.2	255.255.255.0



Kaynaklar

1. Garima Jain, Nasreen Noorani, Nisha Kiran, Sourabh Sharma, Designing & simulation of topology network using Packet Tracer, International Research Journal of Engineering and Technology (IRJET), 2(2), 2015
2. Chang, V., Kamireddy, A., Xu, Q., Li, J., Psarros, C., & Chong, P. L. (2022). Simulate and compare routing protocols for smart green systems. *International Journal of Business and Systems Research*, *16*(3), 302-329.
3. Wu, Bing, “Simulation Based Performance Analyses on RIP, EIGRP and OSPF Using OPNET"

F.Ü. Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Laboratuvarı

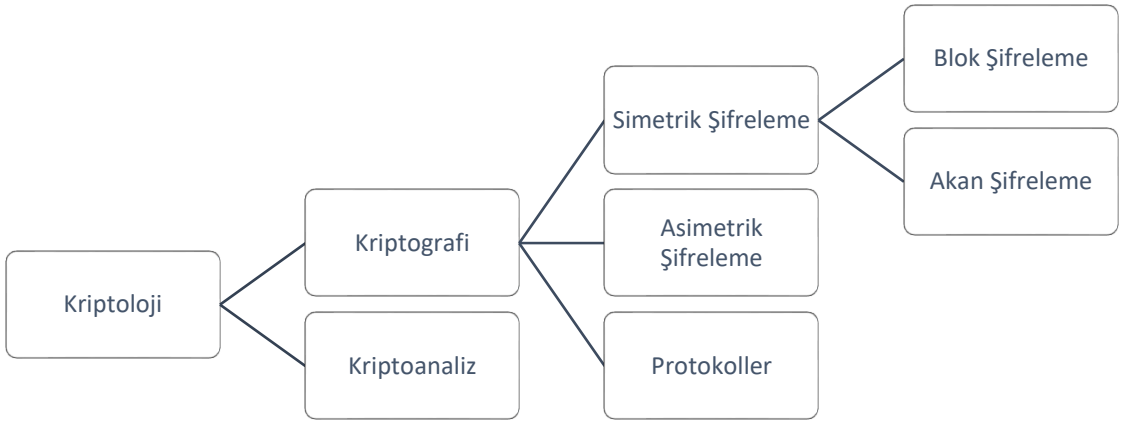
DENEY NO : 3

DENEY ADI : ŞİFRELEME YÖNTEMLERİ

Deneyin amacı:

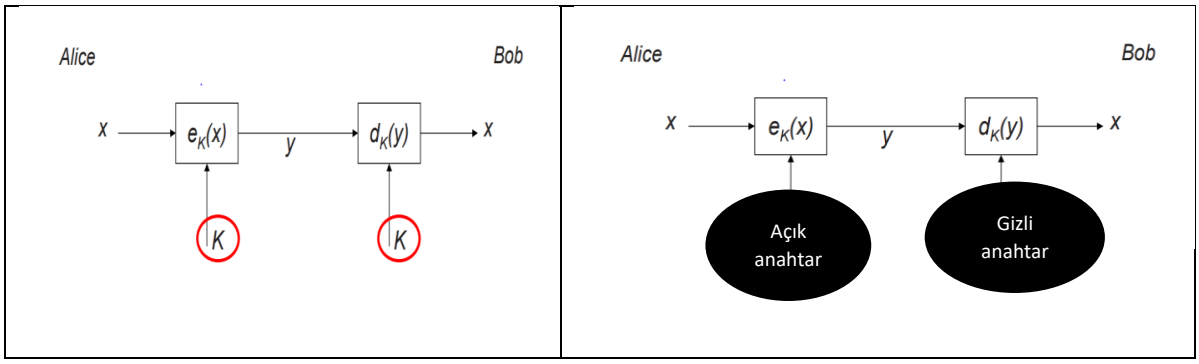
Kriptoloji, kriptografi, kriptoanaliz kavramlarının öğrenilmesi, simetrik şifreleme algoritmalarından DES ve asimetrik şifreleme algoritmalarında RSA algoritmalarının öğrenilmesidir.

Kriptoloji Bilimi



Şekil 1.Kriptoloji bilimi [1]

- Kriptoloji: Haberleşmede veri güvenliğini sağlayan şifreleme cihazlarını, bu cihazlarda kullanılan algoritmaların tasarımını ve bu algoritmaların güvenilirliğini araştırır.
- Kriptografi: iletilen bilginin istenmeyen şahıslar tarafından anlaşılmayacak bir biçime dönüştürülmesinde kullanılan tekniklerin bütünüdür. Güvenliği sağlayan protokolün ortaya konulmasıdır.
- Kriptoanaliz: Kriptolojinin, kriptografik sistemlerin şifrelenmiş metinlerini çözebilmek için bu sistemlerin güvenliklerini inceleyen - zayıf yanlarını bulmaya çalışan dalıdır.



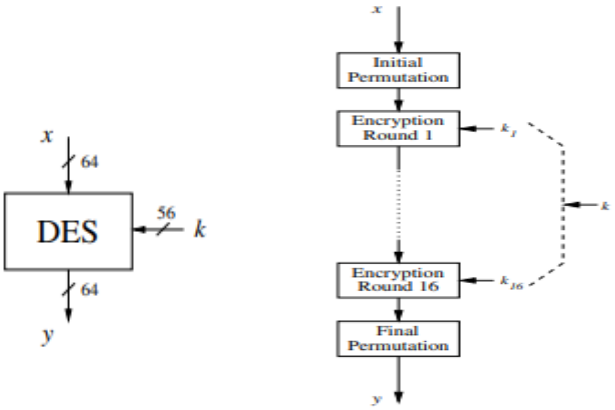
Şekil 2. Simetrik ve asimetrik şifreleme mekanizması [1]

DES Algoritması

Simetrik blok şifreleme algoritması olan DES 1997'de resmi olarak bilgi şifreleme standartı olarak kabul edilmiştir. Ancak 2000 yılında yerini AES'e bırakmıştır. DES algoritması 64 bitlik blok şifreleme algoritmasıdır. Anahtar uzayı ise 56 bit uzunlundadır. Anahtar uzayının 56 bit olması bu algoritmanın güvenli olarak nitelendirmemesine neden olmaktadır. Bu nedenle DES'in güvenilirliğini artırmak için 3DES yöntemi geliştirilmiştir. Bu yöntemde, şifrelenen veri farklı anahtar(lar) ile tekrar geri çözülür ve DES şifrelemesi 3 sefer ardarda yapılır [1].

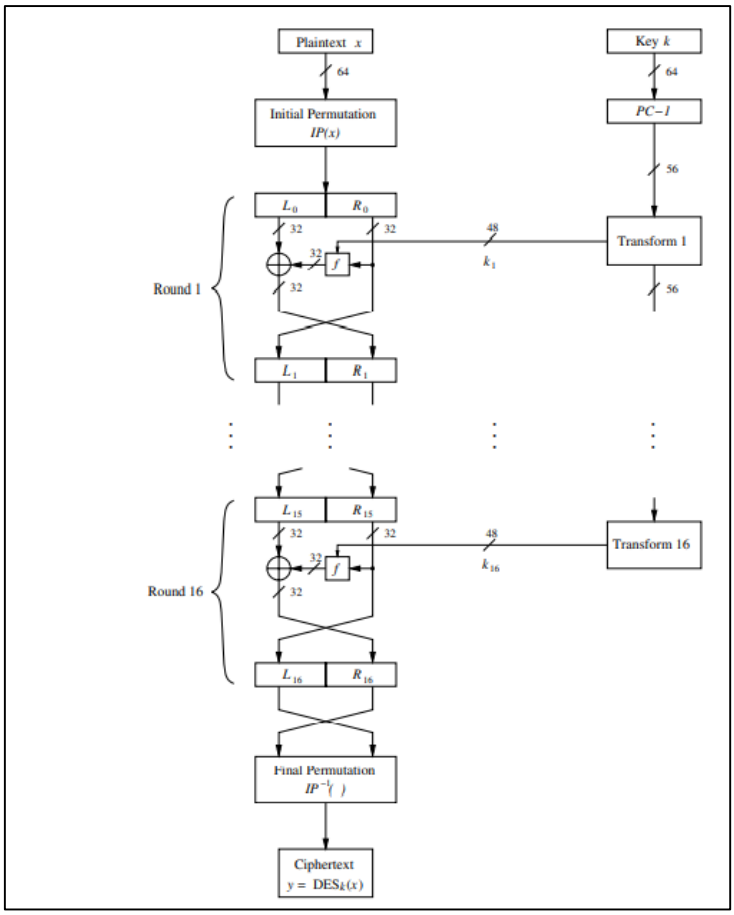
$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$

DES algoritması simetrik şifreleme algoritması olması bakımından hem şifreleme hem de şifre çözme aşamasında aynı anahtarı kullanmaktadır. Şekil 3'de simetrik şifreme yaklaşımı gösterilmiştir. DES 16 adımda veriyi şifrelemektedir. Her bir adımda farklı anahtar kullanmaktadır [1].



- Veri blokları 64 bittir.
- Anahtar uzunluğu 56 bittir.
- Simetrik şifreleme: şifreleme ve şifre çözme sürecinde aynı anahtarı kullanır.
- Her bir adımda yeni anahtar kullanılır.

Şekil 3. DES algoritmasın genel yapısı [1]



Şekil 4. DES Algoritması [1]

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

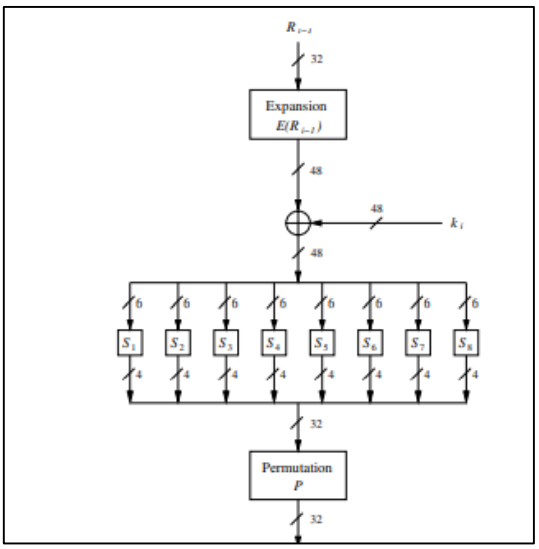
Başlangıç ve bitiş permütasyonları: Şekil 5'de başlangıç permütasyonu (IP) ve bitiş permütasyonu (IP^{-1})gösterilmektedir. Bu permütasyonları kullanmadaki amaç, başlangıçta verilen 64 bitlik veri bloğunu karıştırmaktır. Örneğin, ilk baştaki veri bloğundaki 2 bitin yerine 8 biti koyulmuştur. IP^{-1}'de ise bu işlemlerin tersi yapılmaktadır.

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Şekil 5. Başlangıç ve bitiş permütasyonları [1]

F fonksiyonu Giriş olarak R_{i-1} ve K_i alır ve dört adımdan oluşan işleme tabi tutar bunlar E'nin genişletilmesi, tura ait anahtar ile XOR'lama, S-BOX yer değiştirme ve permütasyondur.



Şekil 6. F fonksiyonunun blok diyagramı [1]

S-Box'lar 4 satır 16 sütündan oluşan matrislerdir. S-box veriyi 6 bitten 4 bite indirgeme yapılmaktadır. Bunun için sayının ilk ve son biti satır sayısını, ortadaki biti ise sütün sayısını gösterecek şekilde tablodaki sayı alınır. Örneğin sayı, 6 bitlik 100101 olsun, MSB ve LSB biti 11 S-Box daki satır sayısını, ortadaki 0010 ise sütündaki sayısı ifade eder. Böylece yeni 4 bitlik sayı 08'dir.

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Permütasyon aşamasında ise, 32 bitten oluşan veri, permütasyon tablosundan yararlanılarak karıştırma yapılmaktadır.

RSA Algoritması

- Ron Rivest, Adi Shamir ve Leonard Adleman tarafından 1978'de geliştirilmiştir.
- Yaygın kullanılan asimetrik şifreleme algoritmasıdır.
- RSA algoritması, anahtarın taşınması ve dijital imza olmak üzere iki amaç için kullanılır.
- Açık anahtar (n,e) ile mesaj şifrelenir. Alıcı tarafta gizli anahtarla (d) şifre çözülür.

$$y = e_{k_{pub}}(x) \equiv x^e \bmod n$$

$$x = d_{k_{pr}}(y) \equiv y^d \bmod n$$

Anahtar üretimi

Açık anahtar:$k_{pub} = (n, e)$ gizli anahtar: k_{pr}=d

1. İki asal sayı seç p,q
2. n=p*q
3. Φ(n)=(p-1)*(q-1)
4. GCD(e,Φ(n))=1 olacak şekilde e asal sayısının seçimi
5. d*e≡1 mod Φ(n)
6. return $k_{pub} = (n, e)$ gizli anahtar: k_{pr}=d

Örnek:

A	B
Mesaj x = 4	1. p = 3 ve q = 11 2. n = p · q = 33 3. Φ(n)=(3−1)(11−1) = 20 4. e = 3 5. d ≡ e^{-1} ≡ 7 mod 20
	$k_{pub} = (33,3)$ ←
$y = x^e \equiv 4^3 \equiv 31 \bmod 33$	
	y =31 ⇒
	$y^d = 31^7 \equiv 4 = x \bmod 33$

Kaynaklar

1. Paar, C., & Pelzl, J. (2009). *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media.
2. Heys, H. M. (2002). A tutorial on linear and differential cryptanalysis. *Cryptologia*, *26*(3), 189-221.

F.Ü. Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Laboratuvarı

DENEY NO: 4
PENETRASYON TESTİ

Deneyin Amacı:

Etik saldırının ve ağ güvenliğinin yapısının ve adımlarının anlaşılması ve bu saldırıda kullanılan yöntemlerin ve araçların incelenmesi ve kullanımıdır.

Penetrasyon Testi nedir?

Penetrasyon Testi; pen test, etik hack, beyaz şapkalı hack olarak da adlandırılabilir. Bu test; bilgisayar ağının ya da sisteminin içeriden ya da dışarıdan gelebilecek saldırılara karşı ne kadar güvende olduğu konusunda fikir edinilmesini sağlar. Bu test sistemin güvenli olmasını sağlamak amacıyla, yasal ve izinli olarak açıklarının aranmasıdır. Bu sayede sistemin zayıf noktalarının tespiti ve saldırı düzenlenerek sistemin gerçek bir hacker'ın saldırısına karşı savunmasız olup olmadığı belirlenir. Sonuç olarak sistemin ve verilerin ne kadar güvende olduğu sorusu spesifik bir sorudur ve cevabı görecelidir. Penetrasyon testi bu sorulara objektifcevaplar bulmaya yardımcı olur.

Penetrasyon testi; dos, application (uygulama), internet, intranet, process gibi farklı bölümler içerebilir. Bu test bu bölümlerin birini veya bir kaçını kaplayacak şekilde yapılabilir.

- Dos testinde şirketin herhangi bir saldırıda hizmetlerinin çöküp çökmeyeceği ve bu saldırıya karşı dayanıklılığı test edilir.
- Application (uygulama), kısmında kullanılan uygulamalar ve kodlarında açık taraması yapılır. Kullanılan uygulamaların güncelliğinin korunması, iç ağın durumunun test edilmesinde bu testler kapsamındadır.
- Dış ağın penetrasyon testinde(intranet) yapılanmanın dış dünyadan gelebilecek herhangi bir tehdide karşı durumu tespit edilir. Örnek olarak web siteye gelebilecek herhangi bir saldırı bu kapsamdadır.

Şirketler pentest ile her türlü saldırıya karşı kendilerini hazırlayabilir ve saldırıya gerekli tedbiri alabilirler. Böylelikle bir saldırı durumunda ya hiç etkilenmezler ya da en az zararla saldırıyı atlatabilirler. Dışarıdan saldırgan bakış açısıyla güvenlik açıklarının kontrolü ve raporlanması güvenlik açısından önemlidir. Şirketlerin kendi içlerindeki güvenlik tedbirleri çoğunlukla yeterli olmamakta ve önlemler güncelliğini koruyamamaktadır. Ayrıca kötü niyetli hackerların sayısının artması ve bilgi düzeylerinin genellikle birçok şirket çalışanından önde olması pentestin önemini ortaya koymaktadır. Pentest bir şirketin bilişim sistemleri için iç ve dış tehditlere karşı güncel önlemler alınmasını ve zafiyetlerin giderilmesini sağlar. Bununla birlikte bu testlerin faydaları aşağıdaki gibi sıralanabilir.

- Saldırılara karşı daha dirençli bir bilişim altyapısı
- Kullanıcı bazlı olarak bilgi güvenliği farkındalığının artması
- Sistemlerin durdurulma veya kaynak doldurmaların engellenmesi
- Yasal olarak uyum sağlama
- Kurum prestijinin ve marka değerinin korunması sağlanır.
- BT kaynaklı risklerinin azalması

Pen testinde Hack sanatında üç terimin sıklıkla kullanılır. Bunlar Beyaz şapkalı (White hats), siyah şapkalı (black hats) ve gri şapkalı (gray hats). Beyaz şapkalılar art niyetli olmayan ve sistemlerin savunmalarını güçlendirmek amacıyla pen test yapan hacker'lardır. Siyah şapkalılar aynı zamanda crackers olarak da adlandırılırlar ve art niyetli amaçlar için hack'i kullanırlar. (Kredi kartı bilgilerinizi çalarak satmak ya da firma sırlarını çalmak gibi…) Bu amaçla beyaz şapkalıların siyah şapkalıların kullandıkları araç ve yöntemleri bilmeleri önemlidir böylece siyah şapkalıların bir adım önünde kalarak art niyetli eylemlerini engelleyebilirler. Gri şapkalılara ise beyaz ve siyah şapkalıların kombinasyonudur ve genellikle yapabildikleri için ya da meydan okumayı sevdikleri için hack'lerler.

Penetrasyon Testinin Adımları

- **Bilgi toplama:** Sistem hakkında ön bilgi edinmek. internet, rakip bilgileri, WHOIS, DNS, ağ, web sitesi, e-posta, Google yoluyla olabilir. Whois, bize aradığımız hedefin alan adı bilgilerini ve dns sunucularını verir. Alan adları ve yapıları bize hedef hakkındaönemli bilgiler verir.
- **Tarama:** Sistemi tarayarak bilgi edinmek. (Sistemdeki bilgisayarlar ve network

cihazlarını tespit etmek, sistemdeki cihazlarda bulunan açık portların tespiti, sistemdeki açıklarının taranması vs.)

- **Erişim Sağlama:** Sistemde bulunan açıklar kullanılarak sisteme izinsiz erişim sağlanması. (yetki yükseltme, sistem durdurma, kaynakların doldurulması, yetkisiz erişim sağlama)
- **Erişim Koruma:** Elde edilen erişimin korunması. (sistem üzerinde elde edilen erişim haklarını kalıcı kılma, sistem üzerinde yetkili bir kullanıcı oluşturma, arka kapı açma).
- **İzleri yok etme:** Hedef sistem üzerinde ilk dört adımda yapılan işlemlerin bıraktığı izler (log kaydı gibi) temizlenir veya kirletilir. (APT-Advanced Persistent Threat).

Penetrasyon Test Araçları

Bilgi güvenliği, özellikle e-ticaret ve e-devlet uygulamalarının yaygınlaşmasıyla birlikte oldukça önemli bir hâle gelmiştir. Bilginin güvenli bir şekilde iletilmesi, işlenmesi ve saklanması bilişim uzmanlarının başlıca görevlerinden birisi olmuştur. İletilen bilginin veya bilgiyi ileten sistemin gerekli güvenlik özelliklerini sağlayıp sağlamadığını test etmek ve denetlemek için ağ güvenliği test ve denetim araçları kullanılmaktadır. Bu araçlardan bazıları ücretsizdir, bazıları ise belirli bir ücretlendirmeye tabidir. Ağ güvenliği test ve denetim araçlarının birçoğu Backtrack altında toplanmıştır. Backtrack, Linux işletim sistemi üzerine kuruludur ve CD'den boot edilerek kullanılmaktadır.

Ağ güvenliği test ve denetim araçları aşağıdaki başlıklar altında gruplandırılabilir:

1. Ağ dinleme araçları
2. Port tarayıcılar
3. Şifre kırma araçları
4. Web güvenliği test araçları
5. Genel Amaçlı Güvenlik Açığı Tarayıcılar

1. Ağ Dinleme Araçları

Ağ ve sunucu trafiğini izlemek için ve ağ dinlemek için kullanılan araçlardır. Ağ dinleme araçları arasında en çok kullanılan ve en yaygın olanı Wireshark programıdır. Wireshark açık kaynak kodlu bir yazılımdır ve internetten ücretsiz olarak indirilebilir. Hem Windows hem de Linux işletim sistemleri üzerinde çalışmaktadır. Wireshark trafiği kaynak adres, hedef

adres, kaynak port, hedef port gibi belirli kriterlere göre yakalayabilmektedir. Ayrıca izlenen trafik sonradan incelenmek üzere kaydedilebilir. Bu program aynı zamanda kablosuz ağları da dinleyebilmektedir. Wireshark'ın kurulumu kullanımı ile ilgili geniş anlatıma sahip bir kayağa*http://www.enderunix.org/docs/wireshark.pdf*

Ping ve Ping Sweeps

Ping özel bir network paketi olup ICMP packet olarak adlandırılır. Ping sweep(pingtaraması) yapmanın en kolay yolu FPing adlı aracı kullanmaktır. FPing'i kullanmak için terminali açarak;

fping –a –g 172.16.45.1 172.16.45.254>hosts.txt

2. Port Tarayıcılar

Hedef makine de ne kadar çok açık port varsa, açıklık potansiyeli de o kadar fazla olmaktadır. Bu yüzden kullanılmayan portların kapatılmış olması gerekir. Hedef bilgisayar üzerinde açık olan portlar, port tarayıcı yazılımlar ile tespit edilmektedir.

En yaygın olarak kullanılan port tarayıcı program Nmap yazılımıdır Nmap, açık kaynak kodlu bir yazılım olup ücretsizdir. Hem Windows hem de Linux üzerinde çalışabilmektedir. Nmap programının en önemli özellikleri şunlardır:

- TCP ve UDP port taraması yapabilmektedir.
- İşletim sistemi tespiti yapabilmektedir.
- Çalışan servisleri tespit edebilmektedir.
- Yazılımların sürümünü tahmin edebilmektedir.
- Bir ağdaki canlı bilgisayarları tespit edebilmektedir.
- Raporlama yeteneği bulunmaktadır. Test sonucunda HTML formatında raporlar çıkarmaktadır.
- Nmap, komut satırıyla çalışan bir programdır. Ancak, Zenmap isminde kullanıcı arayüzüne sahip olan sürümü de çıkmıştır.

Kullanımı için;

Nmap –p 192.168.56.101

-p hedef makinedeki tüm portların taranması anlamını taşır.

HTTrack

Pen testimize hedef siteyi gözden geçirerek başlamak isteriz. Bu amaçla HTTrack adlı araç kullanılarak web sitesinin sayfa sayfa kopyası çıkarılabilir. HTTrack web sayfasının sayfa sayfa ofline kopyasını çıkaran ücretsiz bir programdır. Kopyalanan web sitesi tüm sayfaları, linkleri, resimleri ve orijinal web sitesinin kodlarını içerir ancak tüm bunlar sizin lokal bilgisayarınızda bulunur. HTTrack gibi bir araç kullanarak siteye offline erişim sağlanması şirketin web sunucusunda uzun zaman geçirerek dikkat çekmenin önün geçer. Ve sitenin içeriklerine geniş ulaşım imkânı sağlar. HTTrack işini bitirince hedef sitenin tam bir offline kopyası inceleme için bilgisayarınızda hazır bulunacaktır.

3. Şifre Kırma Araçları

Hedef cihazda çalışan bir servise ait kullanıcı adını ve parolayı kırmak için şifre kırma araçları kullanılmaktadır. Örneğin bir yönlendiricinin yönetimini ele geçirmek için şifre kırma araçları kullanılabilir. Bu araçlar vasıtasıyla yönlendiriciyi yönetmek için kullanılan kullanıcı adı ve parola elde edilebilir. Bu araçlara örnek olarak Cain and Abel, Brutus ve Hydra programları verilebilir.

Bu araçlardan Cain and Abel, ücretsiz bir yazılımdır (Linkleri Görebilmek İçin Üye Olun veya Giriş Yapın.). Sadece Windows işletim sistemleri üzerinde çalışabilmektedir.

Medusa

Medusa, brute force (kaba kuvvet) yöntemini kullanarak şifreyi tahmin etmeye çalışan ve uzak servislere erişim sağlayan bir araçtır. Medusa FTP, http, MySQL, Telnet, VNC, Web Form ve daha pek çok servise saldırı yapma yeteneğine sahiptir. Medusanın kullanılabilmesi için bellibilgilere ihtiyaç duyulur bu bilgiler;

- Hedef IP adresi
- Girişi yapmak için kullanılacak kullanıcı adı veya kullanıcı adı listesi
- Şifre ya da şifre olarak kullanılacak bir sözlük
- Giriş yapmak istediğini servisin adı

Medusa, Backtrack 5'de yüklü olarak gelmektedir. Ancak başka bir sürüm ya da dağıtım kullanıyorsanız yüklemek için konsolda;

```
apt-get update apt-get install medusa
```

Daha önceki bölümlerde yapılan araştırmalardan elde edilen e- mail adresleri ya da hesap isimleri ve şifreler bu bölümde Medusaya girilir. Medusa gibi programlar bu kullanıcı isimlerini ve şifreleri kullanarak başarılı olana kadar denerler. Burada dikkat edilmesi gereken önemli bir nokta günümüz sistemlerinin belirli sayıdaki denemenin ardından saldırınızı fark edip IP'nizi kilitleyebileceğidir. Elbette ki dijital izleriniz kayıt altına alınacak ve sistem yöneticisi uyarılacaktır. Şifresini denediğiniz kullanıcı adının kilitlenmesi de ihtimal dâhilindedir.

Çeşitli şifre listeleri(sözlükler-dictionary) internette bulanabileceği gibi sıklıkla kullanılan şifreleri içeren bir liste Backtrak'da mevcuttur;

/pentest/passwords/wordlists/

Brute-force saldırıyı gerçekleştirmek için konsolda;

medusa –h target_ip –u username –P path_to_password_dictionary –M service_to_attack

-h Hedef hostun IP adresi

-u Medusanın kullanacağı tek kullanıcı adı

-P şifre listesi-Kelime sözlüğü-dictinary file- yolu

-M saldırılacak servis

4. Web Güvenliği Test Araçları

Günümüzde uygulama güvenliği diğer güvenlik araçlarının da önüne geçmiştir. Çünkü uygulamalar genellikle sınırlı bir ekip tarafından geliştirilmekte ve test edilmektedir. Bu da bilinen genel güvenlik yazılım ya da donanımlarına göre daha çok açıklık barındırmalarına sebep olmaktadır. Bu yüzden piyasada bu tür araçlar hızla artmaktadır. Web uygulama güvenliği alanında en önemli araçlardan bazıları şunlardır. Paros, açık kaynak kodlu bir yazılımolup platform bağımsız çalışmaktadır

Genellikle internet tarayıcı ara yüzünden girilmesine izin verilmeyen karakterlerin uygulama yazılımına gönderilmesi için kullanılır. Aynı şekilde uygulama yazılımına paketler gönderilirken yakalanarak içerikleri değiştirilip gönderilebilir. Ya da daha önceden yakalanmış olan paketler gönderilir. Bunların sonucunda uygulama devre dışı bırakılmaya zorlanabilir ya da uygulamanın yapısı hakkında bilgi toplanabilir. Paros kullanılarak ağın haritası çıkarılabilir. Buradan ağın haritasına bakılarak hangi sayfaların olduğu kolayca

görülebilir. Web testi kısmında ise injection, oturum numarası tahmin etme gibi birçok açıklığı uygulama üzerinde deneyebilir. FireBug, Mozilla Firefox'un bir uzantısı olarak çalışır Platformdan bağımsız olarak çalışır. Web sayfasının istenilen herhangi bir yerine gelindiğinde o kısımla ilgili kodu gösterebilir ve o kısımda inceleme yapılabilir. O kısmın kodu kolayca değiştirilebilir. Bu araç hem geliştiriciler hem de testçiler tarafından etkin olarak kullanılabilir.

Ticari bir yazılım olan Acunetix, Windows işletim sistemi üzerinde çalışmakta olup version check, CGI kontrol, parametre değişimi, dosya kontrolü, dizin kontrolü gibi testleri yapmaktadır Bu testleri yaparken istenilen testler için profiller oluşturularak sadece seçilen testlerin yapılması sağlanmaktadır. Uygulama açıklığı taraması yapmaktadır. İstenilen açıklıkları ekleyebilme yeteneği mevcuttur. Yapılan açıklıklarla ilgili detaylı raporlar üretmesinin yanında tek tuşla internetten güncellenebilmektedir.

Netsparker

Netsparker, bir web uygulaması güvenlik tarayıcısıdır. Otomatik olarak bir web sitesini uygulama seviyesindeki güvenlik açıklarına karşı analiz edip güvenlik açıklarını raporlar. Ek olarak raporlamanın bir adım da ötesine geçip güvenlik açıklarını kullanarak aynı bir saldırgangibi sistemden veri çıkartabilir ya da sisteme tam erişim sağlayabiliyor. Bu sayede SQL Injection, Cross-site Scripting gibi açıkları bulmayı sağlar.

Acunetix Web Vulnerability Scanner

Sadece web uygulamalarını denetlemekle kalmamakta, aynı zamanda web uygulamasının bulunduğu sunucu da tüm saldırı yöntemlerine göre denetlemektedir. Serverda bulunan güvenlik açıklarını bize belirtmektedir. Kendi Crawler'ı ile bize web sunucu tipini ve dilinigöstermektedir. Tüm kodlama dillerinin güvenlik açıklarını tarar.

Webinspect

WebInspect, web uygulamalarındaki güvenlik açıklarını, kodlama hatalarını bulup çözüm önerileri getirerek güvenlik duvarı ve saldırı tespit sistemleri için tamamlayıcı bir rol oynar. WebInspect çözümünün kolay yönetilen arayüzü, genişletilebilen fonksiyonları ile ister test ortamında, ister gerçek ortamında, web uygulamalarınızı ve web servislerinizi güvenlik değerlendirmesinden en doğru sonuçları elde ederek geçirebilir. WebInspect kullanıcılara, herhangi bir web uygulamasının ve/veya web servisin in teknolojilerinin uygulama günveliği

açısından denetleme ve olası riskleri bulma olanağı sunar.

5. Genel Amaçlı Güvenlik Açığı Tarayıcılar

Metasploit

Bu penetrasyon testi için en gelişmiş ve en popüler olanıdır. Güvenlik önlemlerini aşmak ve belli bir sisteme girebilecek "Exploit" kavramına dayalıdır. Eğer hedef bir makinede başarılı olduysa, penetrasyon testi için mükemmel bir kod çalıştırır. Web uygulamaları, ağlar, sunucularve benzerleri üzerinde kullanılabilir.

Qualys

Açık tespiti için kullanacağınız programların yanında doğrudan bu iş için tasarlanmış web sitelerine bağlanarak çevrimiçi açık tespiti yapabilirsiniz. Qualysguard Enterprise Intranet Scanner hizmeti böylesi bir açık hat web sitesidir

Nessus

Linux'ta sıkça kullanılan, kapsamlı bir güvenlik açığı tarama yazılımıdır. Kişisel ve hertür kurumsal olmayan kullanım için ücretsizdir. Genel amacı, bilgisayar sistemlerinde ve bilgisayar ağlarında potansiyel güvenlik açıklarını tespit etmektir. Nessus bir port tarama yazılımından çok daha üstün özelliklere sahiptir. Nessus, servislerdeki açıkları eklentilerinin güncelliğine bağlı olarak test edebilir. Çalışma prensibi istemci/sunucu biçimini kullanır ve test edilecek sistemde nessus sunucu yazılımının açık olması daha derinlemesine test ve analiz imkânı sunar.

Örnek Uygulamalar:

1) Ping sweep işlemini gerçekleştiriniz.
2) TCP ve UDP Port tarama işlemini gerçekleştiriniz.
3) Ağda görünen kullanıcı adı ve şifreleri bir log dosyasına yazan bir uygulama gerçekleştiriniz.
4) Ağ trafiğinin istatistiğini çıkaran bir uygulama gerçekleştiriniz.
5) Ağdaki DHCP trafiğini yakalayarak ağ bilgisi çıkaran bir uygulama gerçekleştiriniz.
6) SMTP protokolü için kaba kuvvet saldırısı (brute force attack) yapan bir uygulama gerçekleştiriniz.

7) FTP protokolü için kaba kuvvet saldırısı (brute force attack) yapan bir uygulama gerçekleştiriniz.

Kaynaklar:

1. http://backtracktutorials.com/backtrack-basics
2. "METASPLOIT The Penetration Tester's Guide" by David Kennedy, Jim O'Gorman,Devon Kearns, and Mati Aharoni
3. "The Basics of hacking and penetration Testing Ethical hacking and penetration TestingMade Easy" by Patrick Engebretson

F.Ü. Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Laboratuvarı

DENEY NO: 5
WEB API KULLANIMI

Deneyin Amacı:

Farklı hizmet sağlayıcıları tarafından sunulan hizmetleri projeye dahil etmek için API'ler günümüzde sıklıkla kullanılmaktadır. Bu deneyde, örnek API kullanımı incelenerek, Java ile API servislerine sorgu gönderme işlemi gerçekleştirilecektir.

API Nedir?

API (Application Programming Interface), uygulama programlama arayüzü anlamına gelmektedir. Uygulama programlama arayüzleri, farklı programların birbirleri ile iletişimini sağlayan protokollerdir. Günümüzde sosyal ağlardan alışveriş sitelerine kadar birçok site API'lerini yazarak bunları uygulama geliştiricilerin hizmetine açmıştır. Örneğin bir e-ticaret uygulamasında, geliştirdiğiniz uygulamayı e-ticaret platformları ile entegre etmek isterseniz o platform tarafından geliştirilen API'leri kullanmanız gerekir. API'leri kullanarak, e-ticaret platformuna giriş yapmadan geliştirdiğiniz program üzerinden ürün açma, ürün silme, stok güncelleme, ürünleri listeleme gibi izin verilen işlemleri gerçekleştirebilirsiniz. Veya uygulamanızda hava durumu bilgisi gerekiyorsa, bu bilgiye hava durumu servis sağlayıcısı tarafından sunulan API'yi kullanarak ulaşabilirsiniz. Sosyal ağlar tarafından sağlanan API'leri kullanarak izin verilen gönderiler çekilebilir, mesajlaşma işlemleri gerçekleştirilebilir ve izin verilen kişisel bilgilere (cinsiyet, yaş, arkadaş listesi gibi) geliştirdiğiniz uygulama aracılığıyla erişilebilir. Bazı zamanlar kritik ve veri güvenliğinin önemli olduğu işleri yapmakla görevli olan API'lerin güvenliğinin sağlanması önemli bir konudur. Bunun için API kullanımından önce geliştiricilere özel olarak üretilen bazı kodlar verilerek kimliklerinin doğrulanması sağlanmalıdır.

REST Nedir?

REST (REpresentational State Transfer), temsili durum transferi anlamına gelmektedir. API'lerin belirli standartlara oturması için geliştirilmiş olan bir terimdir. REST API'lerin belirli özellikleri sağlaması gerekir. Bu özellikler Stateless, Uniform Interface, Cacheable, Client-Server, Layered System, Code on Demand olup API geliştirilmesi aşamasında dikkat edilmesi gereken hususlardır. Tüm bu özellikleri sağlayan API'ler RESTFUL API olarak isimlendirilir. REST API'lerde genellikle JSON (JavaScript Object Notation) formatı kullanılarak bilgi alışverişi yapılır.

JSON Nedir?

JSON (JavaScript Object Notation), objelerin veya dizilerin serileştirilmesi ve bu sayede iletiminin kolaylaştırılmasını sağlar. Bir veri transfer edilecekse öncelikle JSON formatına dönüştürülür. JSON özel olarak yapılandırılmış string ifadelerdir. JSON aracılığıyla ulaşması gereken yere ulaşan veri gerektiğinde tekrar nesne veya dizi haline dönüştürülebilir. Örnek JSON formatı aşağıda verilmiştir. JSON'da objeler { }, diziler [ ] ile gösterilir. Dizi içinde obje, obje içinde de dizi kullanımına izin verilir.

```
{
 "no":1,
 "isim": "Ahmet",
 "kurum": {
   "kurumAdı": "Fırat Üniversitesi",
   "birim": "Mühendislik Fakültesi",
   "bölüm": "Bilgisayar Mühendisliği",
 },
 "telefon": [
   { "tur": "ev", "no": "0424 000 00 00" },
   { "tur": "iş", "no": "0424 237 00 00" }
 ],
 "sectigiDersler": [],
 "danışman": null
}
```

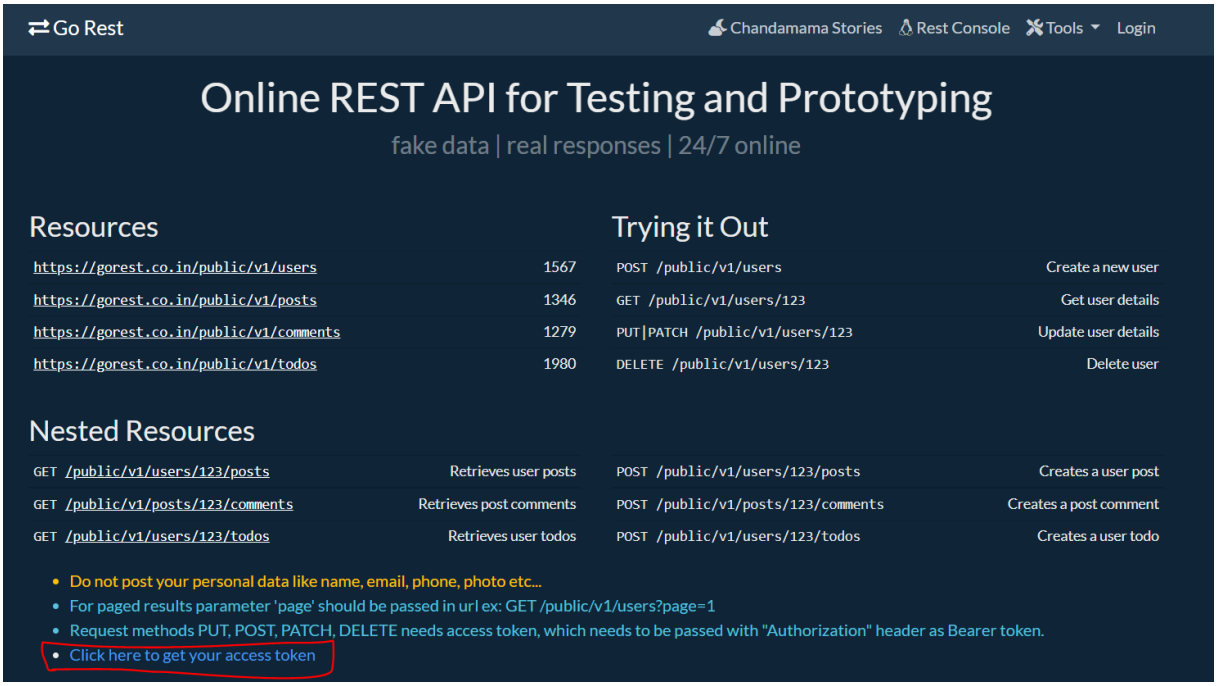
Örnek API Servisleri

Birçok büyük kuruluş API servisi vermektedir. Bunların yanı sıra geliştiricilerin sınırlı denemelerle veya örnek olarak kullanabileceği API servisleri de bulunmaktadır. Collectapi (https://collectapi.com/tr/) sayfasında farklı servis sağlayıcılarına ait API'ler bulunmaktadır. Bunlardan bazıları Tablo 1 ile verilmiştir. Kullanım detaylarına ve hizmet kapsamına ilgili URL'lerden ulaşılabilir. Collectapi sayfası altındaki API'leri ücretsiz olarak denemek için ücretsiz hesap açarak API Key almanız gerekmektedir.

Tablo 1. Collectapi tarafından sağlanan sınırlı ücretsiz kullanıma sahip API listesi

API Adı	URL	Ücretsiz Kullanım
Nöbetçi Eczane API	https://collectapi.com/tr/api/health/nobetci-eczane-api	100 istek / ay
Akaryakıt Fiyatları API	https://collectapi.com/tr/api/gasPrice/akaryakit-fiyatlari-api	100 istek / ay
Altın, Döviz ve Borsa API	https://collectapi.com/tr/api/economy/altin-doviz-ve-borsa-api	100 istek / ay
Haberler API	https://collectapi.com/tr/api/news/haberler-api	1000 istek / ay
Hava Durumu API	https://collectapi.com/tr/api/weather/hava-durumu-api	2500 istek / ay
Faiz Oranları API	https://collectapi.com/tr/api/credit/faiz-oranlari-api	30 istek / ay
Besin Bilgisi Analizi API	https://collectapi.com/tr/api/food/besin-bilgisi-analizi-api	10 istek / ay
Kitaplar API	https://collectapi.com/tr/api/book/kitaplar-api	100 istek / ay
Sözlük API	https://collectapi.com/tr/api/dictionary/sozluk-api	10 istek / ay

Örnek olarak kullanılabilecek bir başka API servisi de https://gorest.co.in/ adresinde sağlanmaktadır. Bir API'yi kullanmak için ilgili API dokümantasyonunun incelenmesi gerekir. Adrese giderek API kullanımı ile ilgili gerekli bilgileri edinebilirsiniz. Genel olarak REST API'ler güvenliği sağlamak için kullanıcı doğrulaması (authentication) gerektirir. Bu işlem API key veya tokenlar ile gerçekleştirilir. Gorest sayfasındaki sahte API'yi kullanmak için de Şekil 1'de gösterilen linkten token almanız gerekir.

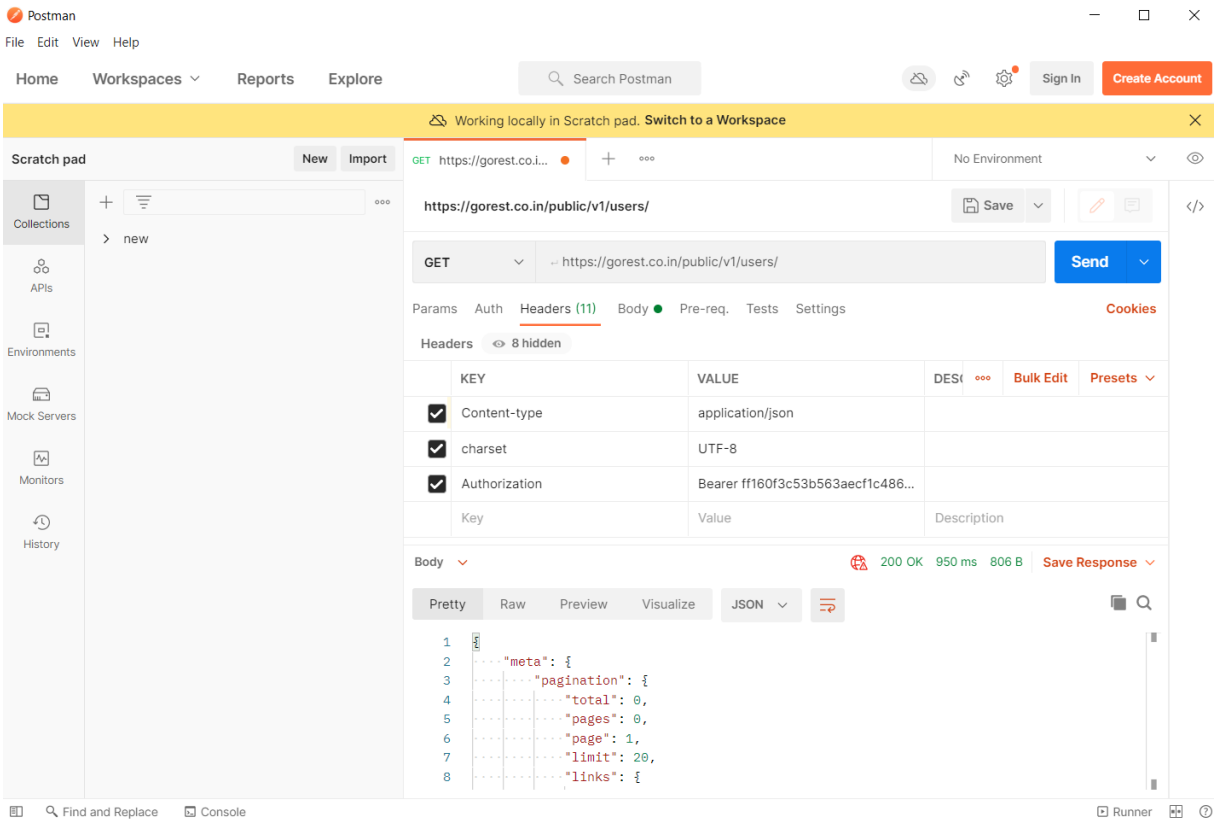


Şekil 1. https://gorest.co.in/ adresinden token alma

Yukarda bahsedilen API örneklerini internet aramaları ile çoğaltmak mümkündür.

API Test Araçları ve POSTMAN

Özellikle büyük çaplı projelerde hatanın kaynaklanma sebebinin bulunması karmaşık bir hale gelir. API kullanılması ile programdan veya API bağlantısından kaynaklanan problemler ortaya çıkabilmektedir. Bu yüzden öncelikle API'lere sorguların düzgün bir şekilde yapılması ve sonucun alınması gerekir. API araçları sayesinde sorguların API dokümantasyonunda yer alan bilgilere göre düzgün bir şekilde gerçekleştirilip gerçekleştirilemediği test edilir. POSTMAN bu amaçla kullanılan bir API aracıdır. POSTMAN programına ait arayüz şekil 2'de verilmiştir. Bu arayüz aracılığıyla API adresine (URI) ilgili header (üst bilgi), params (parametreler) bilgileri ile sorgular gerçekleştirilebilir. POSTMAN programı ile POST, GET, PUT gibi http metotlarını kullanarak sorgular gerçekleştirebilirsiniz. Sorgu sonucunda sunucudan dönen yanıt alt kısımda gözükecektir. Bu kısımda işlem başarılı ile işlem sonucu üretilmesi gereken cevap, başarısız ise de hatanın kodunu içeren mesaj görülmektedir. Sunucudan dönen JSON ifade Raw sekmesi altında olduğu gibi, Pretty sekmesi altında ise hizalanmış bir şekilde görüntülenebilir. Haberleşmede kullanılan orijinal JSON ifade Raw sekmesi altın görüldüğü gibi olmakla birlikte, okunurluğunu kolaylaştırmak için formatlanmış bir şekilde gösterilmesi tercih edilir.



Şekil 2. POSTMAN Arayüzü

Sık Kullanılan API Request Metotları

1. **GET**: Genellikle sunucudan veri almak için kullanılan http metodudur. GET metodu ile sorgu ile gönderilmesi gereken parametreler URL içinde gönderilebilir. Parametrelerin URL gibi görülebilir bir yolla gitmesi gizlilik gerektiren iletişim ihtiyacı için uygun değildir.
2. **POST:** Bu metot ile parametreler hem URL ile hem de body bilgisinde gönderilebilir. Body bilgisinde gönderilen parametreler gizlilik açısından daha etkili olmaktadır.
3. **PUT:** Genellikle veri güncellemek için kullanılan protokoldür.
4. **DELETE:** Genellikle veri silmek için kullanılan protokoldür.

HTTP Durum kodları

API'ler http protokolü üzerinden çalıştıklarından dolayı gönderdikleri hata kodları da http durum kodları ile aynı olmaktadır. http durum kodlarının sınıflandırılması aşağıda verilmiştir. POSTMAN gibi bir araç ile sorgu işlemi esnasında 4xx hata kodlarından biri meydana geliyorsa, bu sorguda (parametre, uri vs.) hata olduğu anlamına gelmektedir.

HTTP Durum Kodu Sınıfları

http durum kodları baştaki ifadeye göre sınıflandırılır. Örneğin durum kodu 1 ile başlayan kodlar aynı türevden benzer hatalar anlamına gelmektedir. http Durum kodu sınıfları aşağıda verilmiştir.

1xx: Gönderilen isteğin sunucuya başarılı bir şekilde ulaştığını ve işlemin başladığını bildiren durum kodlarını içerir (Bilgi içeren mesajlar).

2xx: Gönderilen isteğin sunucuya başarılı bir şekilde ulaştığını ve işlemin başarılı bir şekilde sonlandığını bildiren durum kodlarını içerir (Başarılı işlem mesajları).

3xx: Sunucunun başka bir adrese taşındığı durumlarda verilir (Yönlendirme mesajları).

4xx: Gönderilen sorguda eksik veya yanlış ifadelerin olması durumunda sunucudan dönen durum kodudur (İstemci hatası mesajları). Bu başlıkta gelen hata kodları genellikle sorgunun yanlış veya eksik olmasından kaynaklanır.

5xx: Sunucu tarafında bir hata olduğunu ifade eden durum mesajlarıdır (Sunucu hatası mesajları). Bu durumda servis sağlayıcısının durumu düzeltmesi beklenir.

Sık Karşılaşılan HTTP Durum Kodları:

200: İstemci ve sunucu arasındaki iletişimin başarılı olduğunu gösterir.

201: Gönderilen sorgu sonucunda sunucu tarafında bir verinin oluşturulduğu anlamına gelir.

400: İstek URI'sinin sözdiziminde bir yanlışlık olduğunu ifade eder.

401: Sunucu-istemci bağlantısı için kimlik doğrulaması gerekliliğini ifade eder.

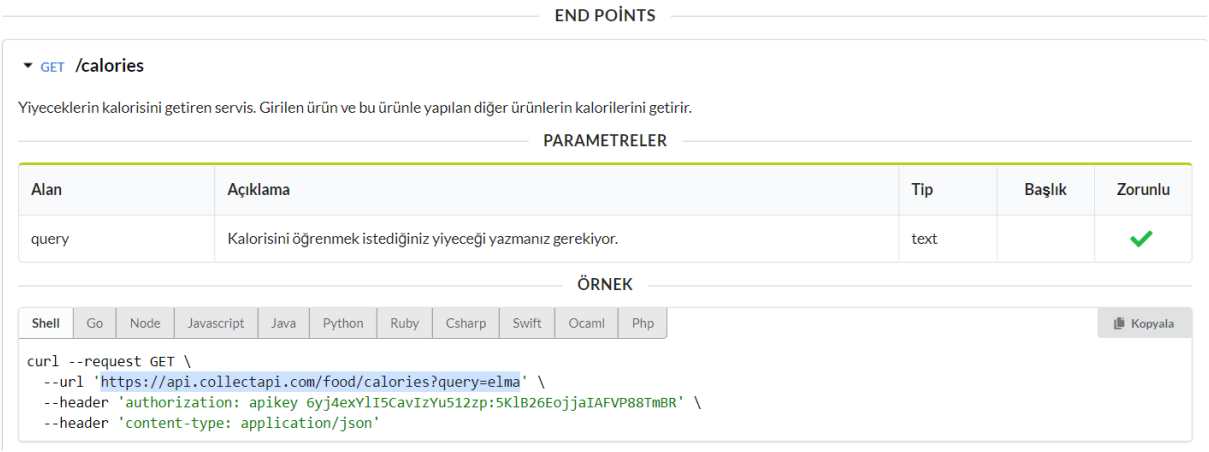
403: Kimlik doğrulaması başarılı olup, kimliği doğrulanan kişinin ilgili URI'ye erişim hakkının olmamasını ifade eder.

404: URI adresinin sunucu tarafından bulunamadığını ifade eder.

405: Gönderilen request metodunun sunucu tarafından desteklenmediğini ifade eder.

POSTMAN ile API Sorgusu Yazma

Collectapi sitesinde yer alan (tablo 1'de verilmiştir) Besin Bilgisi Analizi API'sini kullanarak uygulamamıza entegre etmeyi deneyelim. Besin Bilgisi Analizi API açıklamasında; "Besinlerin kalorilerine ulaşabilir, girilen ayın hangi meyve ve sebzelerin mevsimi olduğunu öğrenebilirsiniz" bilgisi yer almaktadır. Bu API'de GET /calories ve GET /whenFoods isminde iki temel *end point* bulunmaktadır. Giriş yapıp API key aldıktan sonra API dokümantasyonunda yer alan bilgiler ile POSTMAN aracılığıyla sorgu gerçekleştirebiliriz. Şekil 3'te GET /calories metoduna sorgu göndermek için gereken parametreler yer almaktadır.

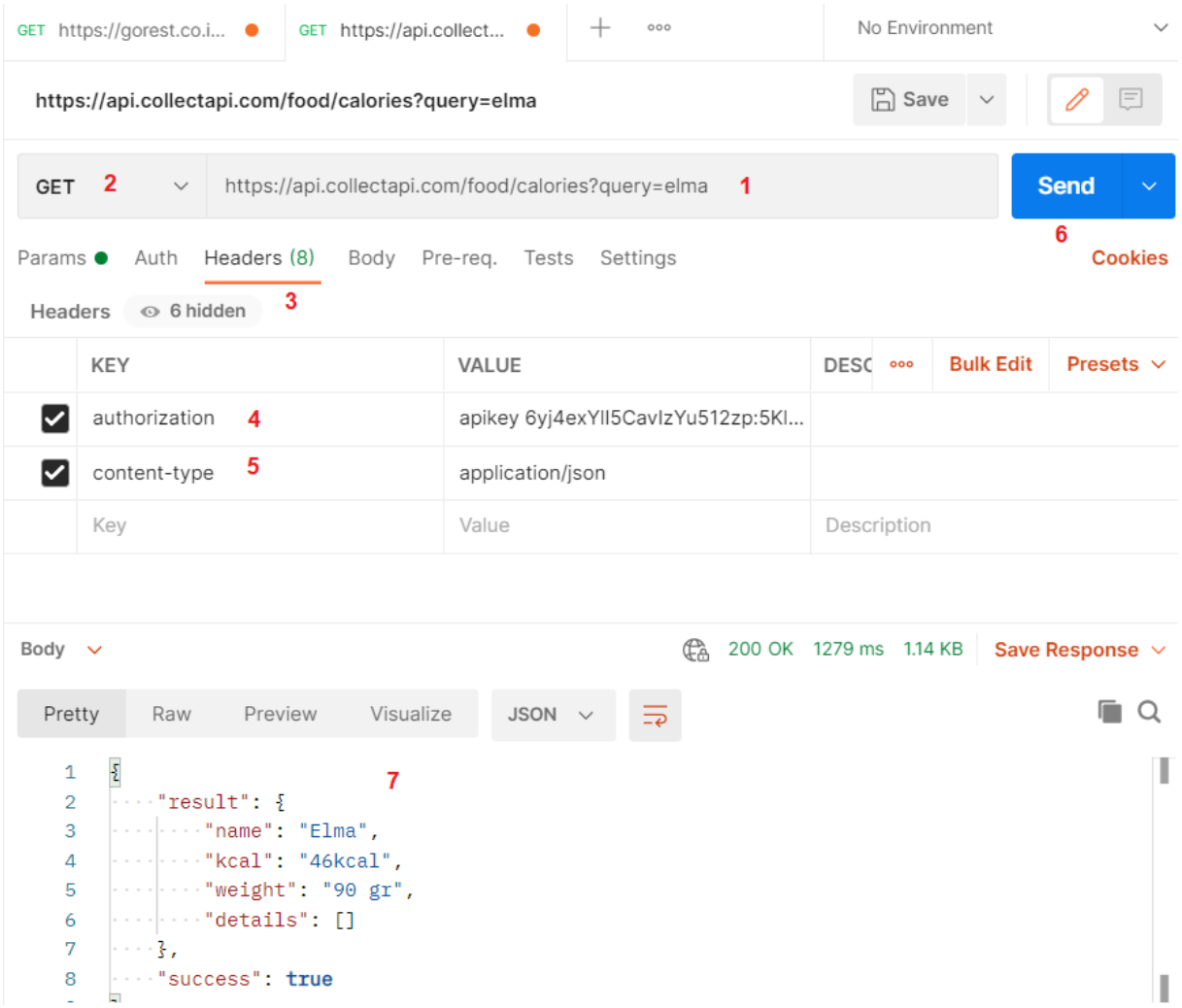


Şekil 3. Besin değerleri API dokümantasyonu.

POSTMAN programını açtıktan sonra file new diyerek yeni bir sorgu ekranı açılır:

1- Bu sorgu ekranında Enter Request Url yazan kısma API'nin erişim adresi ve URL'de göndermek istenilen parametreler yazılır. (https://api.collectapi.com/food/calories?query=elma).
2- Metodun dokümantasyonda belirtildiği şekilde (GET) olduğu kontrol edilir.
3- Header sekmesi açılır.
4- Header sekmesi altında key kısmına "authorization", value kısmına "apikey size_verilen_api_key" yazılır.
5- Yine header altında yeni satıra key yerine "content-type", value kısmına da "application/json" yazılır.
6- Send butonuna tıklanır
7- Sonuç sekmesinde cevap gösterilir.

Verilen adımlar sırası ile şekil 4'te gösterilmiştir.



Şekil 4. POSTMAN ile API sorgusu gönderme

Görüldüğü üzere yapılan sorgu 200 durum kodu ile birlikte dönmüştür. Bu durum kodu sorgu başarılı bir şekilde yürütüldü ve yanıt döndü anlamına gelmektedir. Dolayısı ile API sorgusunun bu parametreler ile doğru bir şekilde çalıştığı doğrulanmış olur. Bu aşamadan sonra programlama dili aracılığıyla aynı parametreler ile API sorgusu gerçekleştirilebilir.

JAVA ile Örnek API Bağlantısının Gerçekleştirilmesi

API'lerin amacı yazılım geliştiricileri için hizmet sunmaktır. Dolayısı ile API'ın bir programlama dili ile kullanılması gerekir. POSTMAN aracı ile doğrulanan API sorgusunun programlama dili ile gönderilmesi için ilgili programlama dillerinde API'lerin nasıl kullanıldıklarına bakılmalıdır. Bu deneyde Java programlama dili kullanılarak sorgular gerçekleştirilecektir.

Java ile http istekte bulunmak için client ve request nesneleri oluşturmak gerekmektedir. Request nesnesi uri ve parametre bilgilerini içermektedir. Örnek bir sorgunun yazılması aşağıda verilmiştir. Dönen cevap response nesnesi olup, body() fonksiyonu ile JSON formatında dönen bilgi yazdırılabilir. Kodda yer alan … kısmına site aracılığıyla size verilen API key yazılmalıdır. Burdaki query=armut kısmı dinamikleştirilerek bu bilgiyi kullanıcının girmesi sağlanabilir.

```
// client (istemci) oluşturma
var client = HttpClient.newHttpClient();
// request (sorgu) oluşturma
var request = HttpRequest.newBuilder()
  .uri(URI.create("https://api.collectapi.com/food/calories?query=armut"))
  .header("authorization", "apikey ...")
  .header("content-type", "application/json")
  .build();
var response = client.send(request, BodyHandlers.ofString());  //BodyHandlers.discarding()
System.out.println(response.body());
```

Java kodunun çalıştırılması ile konsolda yazılan ifade aşağıdaki gibi olmalıdır.

```
{"result":{"name":"Armut","kcal":"114kcal","weight":"200 gr","details":[]},"success":true}
```

Dönen cevap JSON formatında olup, JSONObject nesnesi ile ayrıştırılabilir. Dönen JSON cevabı yazılımın ihtiyacına göre Java nesnelerine dönüştürülebilir. JSON stringinden direkt olarak veriyi okumak için JSON formatını bilmemiz gerekmektedir. Bu örnekte result objesi altında name, kcal, weight gibi bilgiler yer aldığından, örneğin name bilgisini çekmek için aşağıdaki kod eklenmelidir. Yine daha öncesinden response nesnesi ile dönen durum kodunun incelenerek işlemin başarısı kontrol edilebilir.

```
JSONObject obj = new JSONObject(response.body()); //response.statusCode()
 System.out.println(obj.getJSONObject("result").getString("name"));
```

Yukardaki işlem Java net kütüphanesi kullanılarak gerçekleştirilmektedir. İstenirse bu işlem için 3. Part uygulamalar da kullanılabilmektedir. Collectapi sitesinde verilen kodu doğrudan çalıştırmak için ekstra kütüphaneler dahil edilmesi gerekmektedir. Unirest-java isimli kütüphaneyi kullanarak verilerin çekilmesi aşağıda gösterilmiştir. Burada dönen JSON verisini yazdırmak için getBody() fonksiyonu kullanılmalıdır.

```
kong.unirest.HttpResponse<String> response = Unirest.get("…?query=elma")
      .header("content-type", "application/json")
      .header("authorization", "apikey …")
      .asString();
System.out.println(response.getBody());
```

Kütüphanenin Maven kullanarak projeye dahil etmek için pom.xml dosyası içerisinde <dependencies></dependencies> etiketleri arasına aşağıdaki satırlar eklenmelidir.

```
<dependency>
   <groupId>com.konghq</groupId>
   <artifactId>unirest-java</artifactId>
   <version>3.12.0</version>
</dependency>
```

Deneyin Gerçekleştirilme Aşamaları

Deney için herhangi bir API servisi kullanarak istediğiniz programlama dili ile bu servise sorgu yapmanız istenmektedir. Deneyin gerçekleştirilme aşamaları aşağıda verilmiştir:

1- Problemin (hava durumu, nöbetçi eczaneler gibi) ve kullanılacak API servisinin belirlenmesi, ilgili servisten API key gibi gerekli erişim izinlerinin alınması.
2- API dokümantasyonu incelenerek end pointlere nasıl sorgu gönderileceği, hangi parametrelerin gönderilmesi gerektiğinin belirlenmesi.
3- POSTMAN programı ile belirlenen parametreler ile ilgili uri'lere sorgu gerçekleştirilerek, API'nin düzgün bir şekilde çalıştığının gösterilmesi.

4- Seçilen programlama dilinde http/api isteklerinin nasıl gerçekleştirileceği araştırılarak ilgili kodların yazılması.
5- Gelen cevaptaki http durum kodunun okunarak hatalı bir durum olup olmadığının kontrol edilmesi. Hatalı bir durum varsa hatanın yazdırılması.
6- Hatalı bir durum olmadığı durumda gelen JSON cevabının ayrıştırılarak ekranda/konsolda ilgili yerlerde gösterilmesi. İlgili butonlara tıklandığı/konsolda ilgili komut verildiği zaman API servise sorguda bulunulması.
7- Kullanıcıların hatalı parametreler girmesine veya http durum kodu hatalarına karşı gerekli önlemlerin alınarak programın çökmeden çalışmasının sağlanması.

Hazırlık Soruları

1- API kavramı nedir? Ne için ihtiyaç duyulur? Kullanım alanlarını araştırınız.
2- End point, uri, istemci, sunucu kavramlarını açıklayınız.
3- Arama motorları, sosyal ağlar ve e-ticaret siteleri gibi kategorilerde API hizmeti sunan firmalar hangileridir?
4- Bu firmalardan bir tanesinin API dokümantasyonunu inceleyerek ne tür sorgulara izin verdiğini öğreniniz.
5- JSON ve XML nedir? Birbirlerine göre avantajları dezavantajları nelerdir?
6- JSON formatında diziler ve nesneler nasıl ifade edilir. Araştırınız.
7- API Test Aracı nedir? En çok kullanılan API Test Araçları hangileridir?
8- Http durum kodları kategorileri nelerdir? Bütün durum kodlarının ne anlama geldiklerini araştırınız.
9- JSON string, JSON obje nedir? Farkları nelerdir? Kullandığınız programlama dilinde bunların dönüşümünü sağlayan hazır sınıflar var mıdır? Araştırınız.

Kaynaklar:

1. Zhou, W., Li, L., Luo, M., & Chou, W. (2014, May). REST API design patterns for SDN northbound API. In 2014 28th international conference on advanced information networking and applications workshops (pp. 358-365). IEEE.
2. Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: a case study. Caine, 9, 157-162.

3. POSTMAN, Build APIs together, https://www.postman.com/, erişim tarihi: 30.09.2022
4. Amazon, API nedir?, https://aws.amazon.com/tr/what-is/api/, erişim tarihi: 30.09.2022
5. Mulloy, B. (2013). Web API design. https://hashingit.com/elements/research-resources/2012-web-api-design.pdf
6. Berners-Lee, T., Fielding, R., & Frystyk, H. (1996). Hypertext transfer protocol--HTTP/1.0 (No. rfc1945).
7. API Marketplace, https://collectapi.com/tr, erişim tarihi: 30.09.2022
8. GraphQL and REST API for Testing and Prototyping, https://gorest.co.in, erişim tarihi: 30.09.2022

F.Ü. Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Laboratuvarı

DENEY NO : 6

DENEY ADI : GÖRÜNTÜ İŞLEME UYGULAMALARI

1. GİRİŞ

Elde edilen görüntünün bilgisayara aktarılıp üzerinde herhangi bir işlem yapılması ve ardından görüntüleyici çıkışa iletilmesine sayısal görüntü işleme adı verilir. Görüntü işleme teknikleri ile sayısal görüntüler iyileştirilerek nesne tanıma, hedef tanıma gibi işlemler gerçekleştirilebilmektedir. Görüntü işleme, amaca göre çeşitli işlemlerden oluşmaktadır. Görüntü işleme; tasarım, imalat, güvenlik, tıp, elektronik, makine ve jeodezi gibi alanlarda çok geniş bir uygulama alanı bulmuştur. Aşağıda görüntü işlemenin kullanıldığı alanlardan bazıları, uygulamalarıyla beraber verilmiştir.

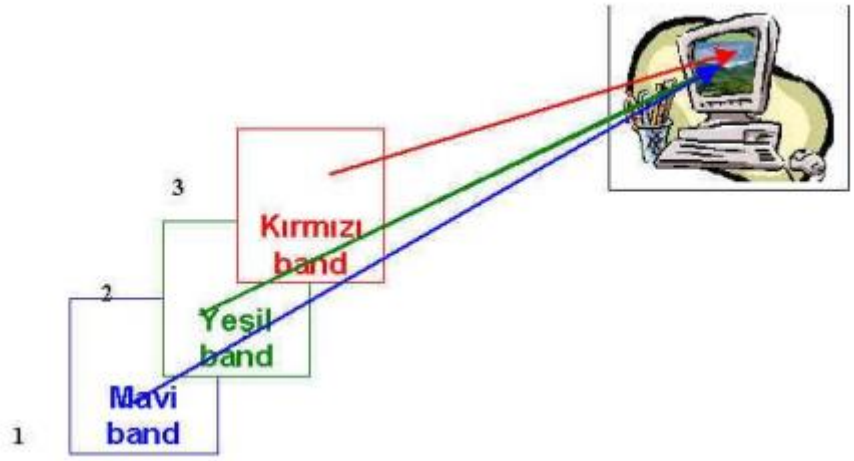
- Askeri (hedef tanıma, izleme)
- Tıp (damar analizi, bilgisayarlı tomografi, ultrason)
- Güvenlik (nesne takibi, hareket algılama, yüz tanıma)
- Trafik (trafik kontrol, plaka tanıma, trafik ışığı (işareti) tanıma)
- Endüstri (nesne sayma, kalite kontrol, robotik uygulamalar)
- Tarımsal uygulamalar (ekin verimliliği tespiti)
- Astronomi (uydu görüntüleri ile hava tahmini)
- Jeodezi ve Fotogrametri (uzaktan algılama)
- Perakende (insan sayma, davranış analizi, mağaza izleme)
- Çevre güvenliği (çevre kirliliği tespiti)

Sayısal resim haline getirilmiş olan gerçek yaşam görüntüsünün, girdi resim olarak işlenerek, resmin özelliklerinin ve görüntüsünün değiştirilmesi sonucunda yeni bir resmin oluşturulması işlemidir. Birçok amaç için kullanılmaktadır.

- Görüntü İyileştirme
- Görüntü Sıkıştırma
- Biyometrik Tanıma
- Otomatik yüz araç obje tanıma ve takip etme

2. İKİLİ GÖRÜNTÜ (BINARY IMAGE)

Bir resmin sayısallaştırılmasının açıklanması amacı ile öncelikle Siyah-Beyaz resim göz önünde bulundurulmuştur. Siyah-Beyaz resim sadece iki gri değerden oluşan bir resimdir. Böylesi bir görüntüde her bir piksel ya siyah ya da beyaz olarak oluşur. Burada sembolik olarak beyaz pikseller 1, siyah pikseller 0 değeri ile gösterilecektir. Renkli görüntüler bilgisayar ekranlarında 24 bit lik veri olarak görüntülenir. Görüntüleme R(Kırmızı), G(Yeşil), B(Mavi) kodlanmış aynı objeye ait üç adet gri düzeyli görüntünün üst üstle ekrana iletilmesi ile oluşur. Elektro-manyetik spektrumda 0,4-0,5 mm dalga boyu mavi renge; 0,5-0,6 mm dalga boyu yeşil renge; 0,6-0,7 mm dalga boyu kırmızı renge karşılık gelir. Bu dalga boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası ile kırmızı-yeşil-mavi kombinasyonunda üst üste düşürülecek olursa renkli görüntü elde edilmiş olur.



Şekil 1. RGB Görüntü

3. NİTELİKLENDİRME

Görüntünün piksel değerlerinin belirli aralıklarda olması, meydana gelen görüntünün niteliğini değiştirir. Örneğin 0 beyazı ve n1 de siyahı temsil ederse ve bu değerler arası gri tonlarını ifade eder. Burada n= 2^b olmak üzere, b değeri görüntünün 1 pikselini ifade etmek için gereken bit sayısıdır. Örneğin b=8 ise 256 adet gri tonu bulunmaktadır. b=1 ise resim sadece 0 ve 1 ‘lerden oluşur ve buna İkili resim(Binary Image) denir.

4. MATLAB İLE GÖRÜNTÜ İŞLEME

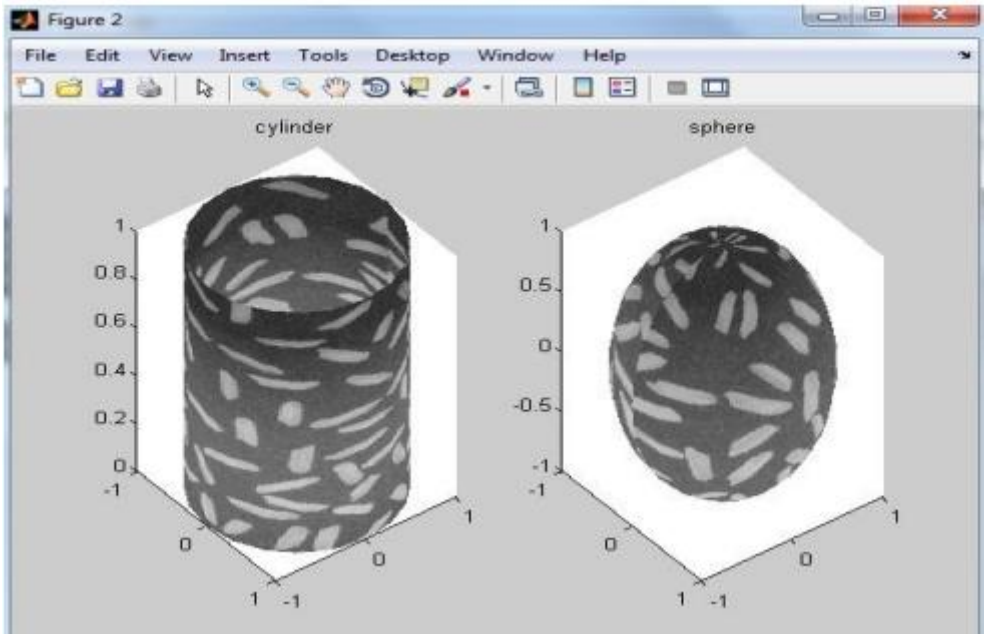
Söz konusu uygulamaları geliştirmek için kullanılan, Matlab’ın görüntü işleme komutlarının ve image processing tools’un bazılarını örneklerle inceleyeceğiz.

imread: imread ile üzerinde çalışılmak istenilen fotoğraf bir değişkene atanarak matlab workspace için tanımlanır ve ardından imshow ile pencerede gösterilebilir. Workspace’te imgenin boyutu, çözünürlüğü gibi bilgileri görebilirsiniz.
imtool: imtool ile bazı işlemlerin yapılabileceği bir pencere açılır. Örnek kodları yazarken workspacedeki verileri temizlemek için, clear all ve clc komutlarını en başa yazmayı unutmamamız gerekir.

```
I = imread('ornek.jpg')
imshow(I)
imtool(I)
```

warp: görüntüyü (içinde yazı olan görüntü daha iyi olur) belli yüzeylerde gösterir.

```
resim = imread('rice.png');
[x1,y1,z1] = cylinder;
[x2,y2,z2] = sphere;
imshow(resim); title('Orijinal resim');
figure;
subplot(1,2,1);
warp(x1,y1,z1,resim);
title('cylinder');
subplot(1,2,2);
warp(x2,y2,z2,resim);
title('sphere');
```



Şekil 2. Elde Edilen Görüntü

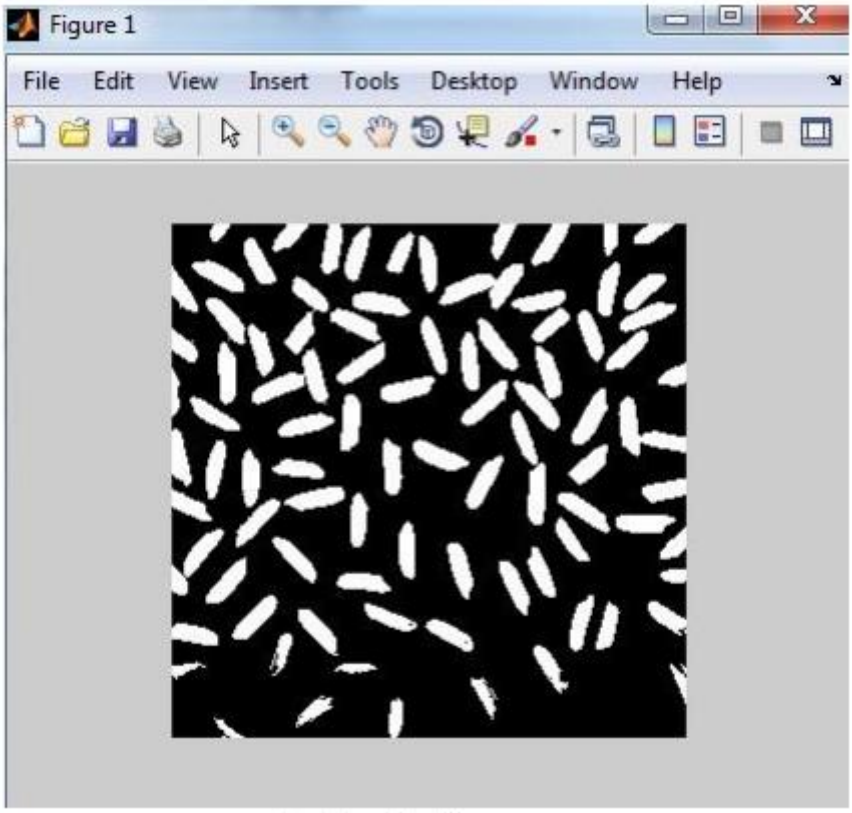
4.1. Tür Dönüşüm Kodları

rgb2gray: renkli görüntüyü gri seviyeye çevirir.

graythresh: graythreshold komutu görüntüdeki parlaklık eşiğini otomatik olarak belirler ve sonuç olarak 0 – 1 arasında bir sayı (level) oluşturur. Görüntüdeki parlaklık sınırları ile yapacağımız işlemlerde graythreshold' tan elde ettiğimiz sayıyı kullanırız. Görüntü üzerinde belirli işlemleri yapabilmemiz için öncelikle gri seviyede çalışmamız gerekir. Bunun için **I= rgb2gray(I)** komutu kullanılır.

im2bw: renkli görüntüyü ikili görüntüye çevirir. Kullanımı aşağıdaki gibidir.

```
I = imread('image.png');
level = graythresh(I);
bw = im2bw(I,level);
bw = bwareaopen(bw, 50);
figure; imshow(bw);
```



Şekil 3. Elde Edilen Görüntü

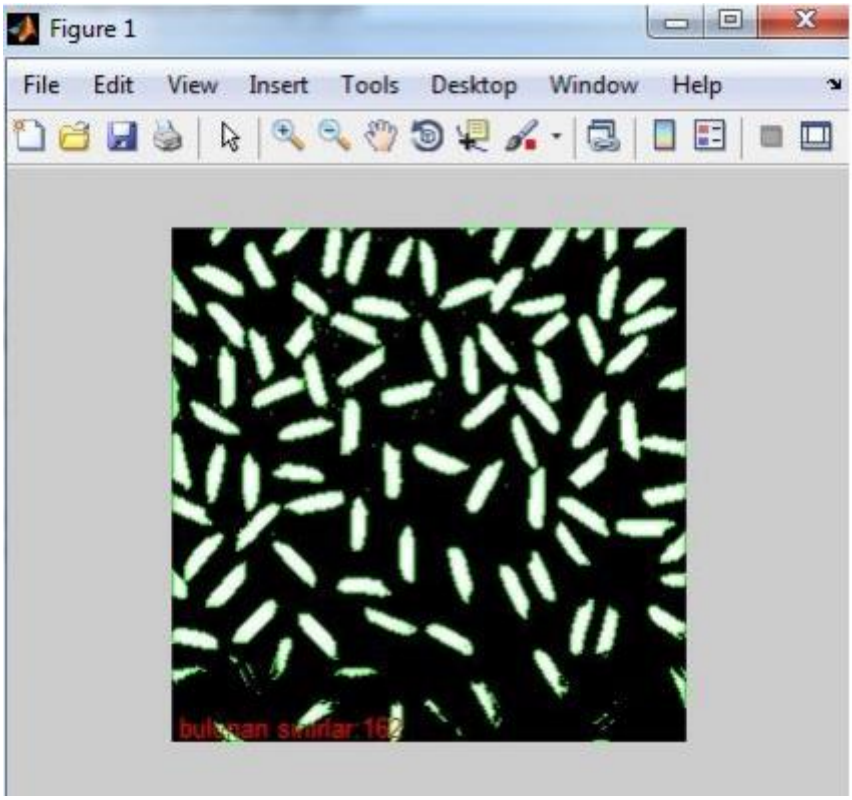
im2uint8: uint8 renk hassasiyetine dönüştürür. Genelde görüntüler uint8 türündedir. Her piksel işaretsiz 8 bit (0 – 255) arası değere sahiptir.

Diğer dönüşüm komutları; demosaic, gray2ind, grayslice,im2int16, label2rgb, im2double, im2uint16, mat2gray…

4.2. Görüntü Analizi Kodları

Bwboundaries: binary modda bölgelerin sınırlarını belirler. Bu örnekte, yeşil çizgiyle sınırlanmış alanların sayısını belirleyebiliriz. Bu örnekle daha sonra bahsedeceğimiz, morfolojik işlemler gerçekleştirilerek basit nesne sayma uygulaması yapılabilir.

```
I = imread('image.png');
BW = im2bw(I, graythresh(I));
B = bwboundaries(BW);
figure; imshow(BW);
text(10,10,strcat('\color{red}bulunan sınırlar:',num2str(length(B))))
hold on;
for k = 1:length(B)
 boundary = B{k};
 plot(boundary(:,2), boundary(:,1), 'g', 'LineWidth', 1)
end
```



Şekil 4. Elde Edilen Görüntü

Edge: özel filtreler yardımıyla gri seviyedeki görüntülerin sınırları belirlenir.

```
I = imread('image.png');
imshow(I);
BW1 = edge(I,'prewitt');
BW2 = edge(I,'canny');
figure; imshow(BW1);
figure; imshow(BW2);
```

Diğer analiz komutları: hough, houghlines, corner…

4.3. Görüntü İyileştirme Komutları

imadjust: görüntü yoğunluğu değerini ve renk haritasını ayarlar.

histeq: histogramı eşitleyerek kontrastı artırır.

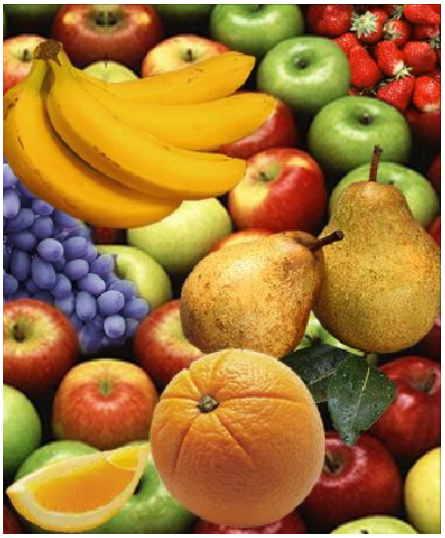
adapthisteq: CLAHE(kontrast sınırlı adaptif histogram eşitleme) algoritması kullanarak kontrastı artırır.

medfilt2: 2 boyutlu medyan filtreleme yapar.

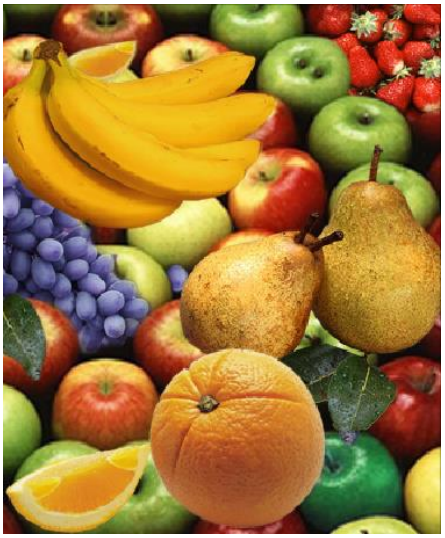
wiener2: 2 boyutlu adaptif gürültü temizleme filtresidir.

5. DENEY

Bu uygulamada verilen benzer iki resim arasındaki farklar bulunup bu farklar hem sayıca hesaplanmakta hem de işaretlenmektedir.

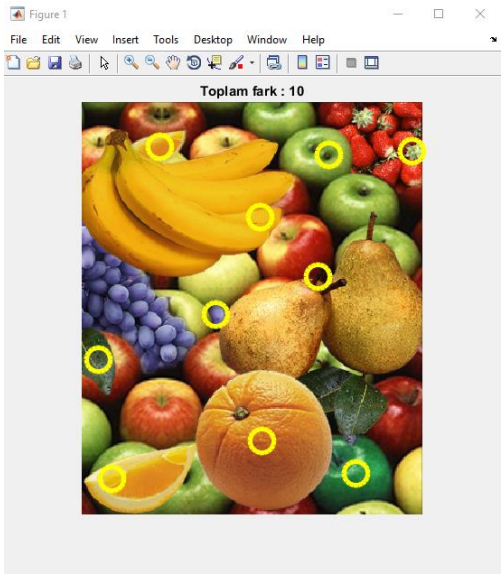


Şekil 5. Benzer Görsel 1



Şekil 6. Benzer Görsel 2

```
im1 = rgb2gray(imread('im1.png'));
im2 = rgb2gray(imread('im2.png'));
fark = imabsdiff(im1,im2);
bw = bwareaopen(fark,55);
bw = imfill(bw,'holes');
SE = strel('square',1);
bw2 = imerode(bw,SE);
fark = regionprops(bw2, 'all');
c = [fark.Centroid];
imshow('im2.png');
title(['Toplam fark : ',num2str(length(fark))]);
hold on;
x = c(1:2:end);
y = c(2:2:end);
plot(x,y,'yo','MarkerSize',20,'LineWidth',4);
```



Şekil 7. Elde Edilen Çıktı

Not: Deney esnasında deney föyündeki ve slayttaki tüm bilgi ve kod içeriğinden sorumlu olduğunuzu unutmayınız…

KAYNAKLAR

1. https://www.mathworks.com/
2. https://www.elektrikport.com
3. https://guraysonugur.aku.edu.tr/wp-content/uploads/sites/11/2019/03/GI-Ders4_1.pdf
4. https://akademik.duzce.edu.tr/Content/Dokumanlar/pakizeerdogmus/Dosya/ab1930a7-d01f-44bf-9715-98261cbe69a4.pdf