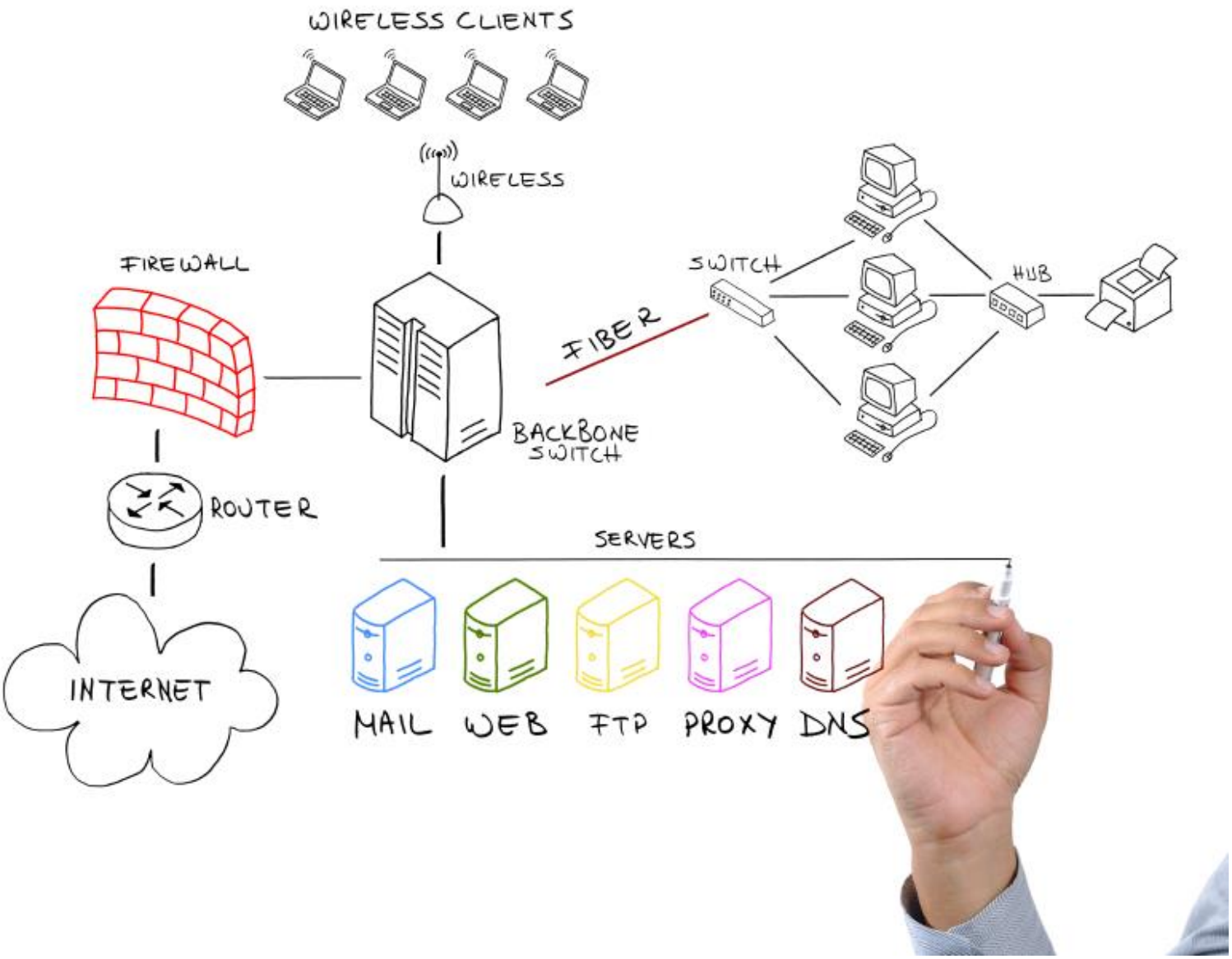
T.C.
FATİH
SULTAN
MEHMET
VAKIF ÜNİVERSİTESİ
2010



BİLGİSAYAR AĞLARI

LABORATUVAR

7

1. Önkoşullar

1.1. Veri yapılarını bilmeli
1.2. Protokol kavramını bilmeli
1.3. Client-Server terimlerinin anlamlarını bilmeli
1.4. Katmanlı mimariyi bilmeli
1.5. Wireshark programını kullanabilmeli

2. HTTP Nedir?

HTTP, Hypertext Transfer Protocol (Hypertext Transfer Protokolü) demektir. Günümüzde web sayfaları HTML yani Hypertext Markup Language (Hypertext İşaretleme Dili) ile yazılır (bunun yanında ek script ve işaretleme dilleri de vardır). Kullandığımız tarayıcılar HTML dosyasını hedef adresten indirir ve işaretlemeleri kullanarak kullanıcıya bu dosyayı görsel bir şekilde gösterir.

Bu laboratuvarda basit GET/POST iletişimini, HTTP mesaj formatını, büyük HTTP dosyaları almayı ve gömülü HTML objelerini göreceğiz.

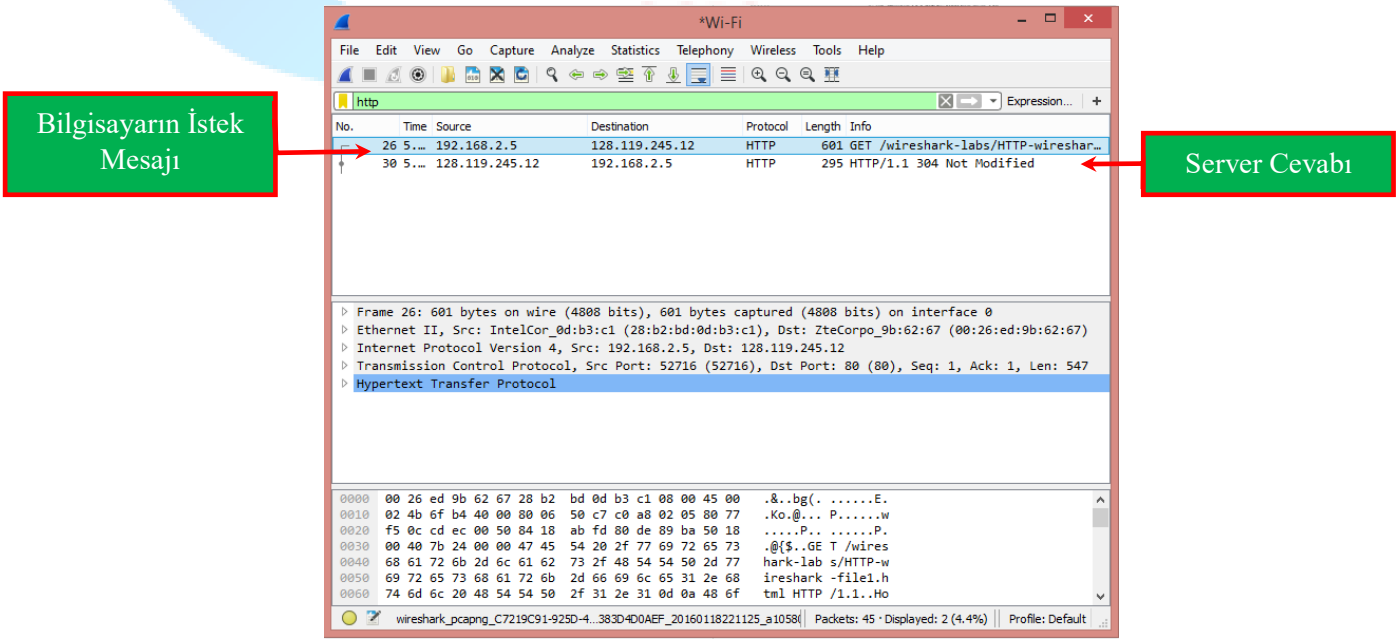
3. Server Client İletişimi?

İçerisinde gömülü objeler bulunan basit bir HTML sitesini tarayıcımızdan açarak başlayacağız.

1- Tarayıcınızı çalıştırınız
2- Wireshark programını çalıştırınız fakat paket dinlemeyi başlatmayınız
3- Filtre alanına "HTTP" yazınız
4- Wireshark paket dinlemeyi başlatıp aşağıdaki linki tarayıcınıza giriniz ve Enter'a basınız

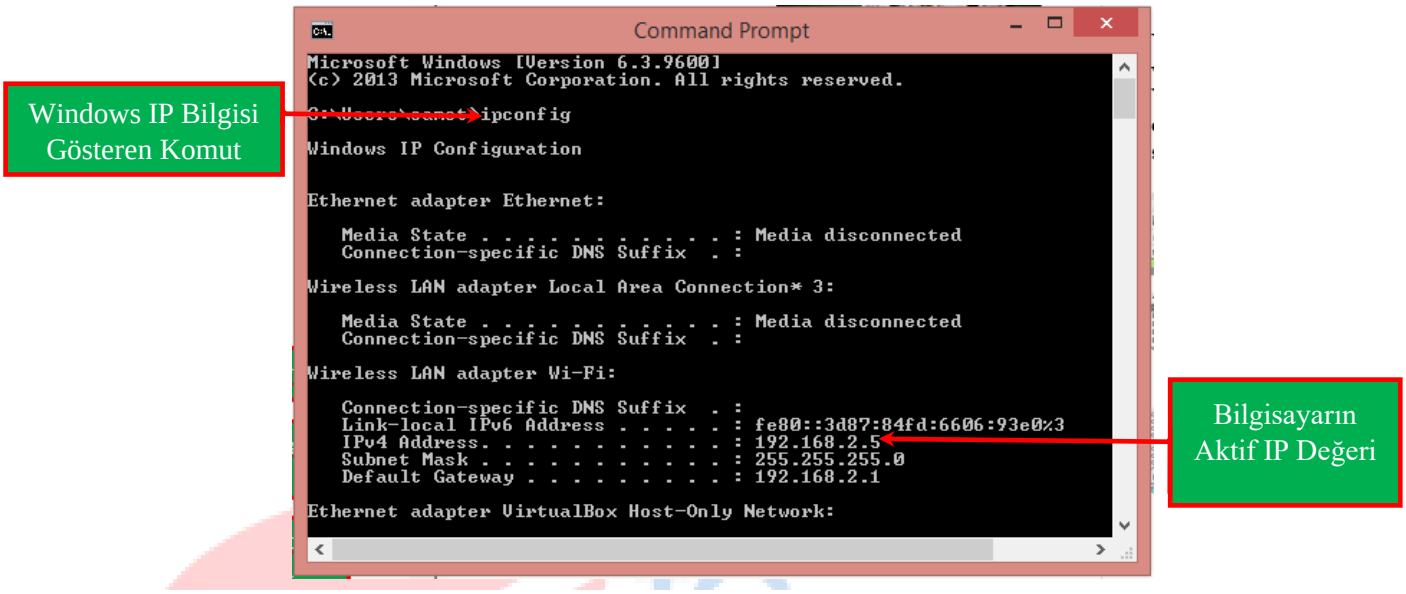
   Link: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

5- Paketin geldiğini görünce Wireshark paket dinlemeyi durdurunuz.



Resim 1: HTTP Get/Response

İşlemleri doğru yaptıysanız Resim 1' de görüldüğü gibi iki adet paket yakalamış olması gerekir. Bağlantı isteği client olan bilgisayarımızdan gitmektedir. Bu "GET" isteği olarak isimlendirilir. Bir başka deyişle bilgisayarımız belirli bir servera bir web sayfası için "GET" isteğinde bulunur. Buna karşılık karşı server HTML dosyasını client bilgisayarımıza indirir ve tarayıcımız sayfayı görselleştirerek son kullanıcıya sunar.



Resim 2: CMD Üzerinden IP Bilgisi Görüntüleme

Resim 2'de bilgisayarımızın IP adresini öğrenebilmek için CMD ekranına "ipconfig" yazıyoruz. IPv4'e karşılık gelen IP adresi bizim bilgisayarımızın IP'sidir. Diğer bilgilerle şimdilik ilgilenmiyoruz.

Resim 3: Server IP Bilgisi Görüntüleme

Bağlanmış olduğumuz server'ın ip değerine erişmek için domain adını, IP gösterebilen herhangi bir siteye giriyoruz. Resim 3'te görüldüğü gibi bağlanmış olduğumuz server IP değerine ulaşabildik.

(IP kontrolü yapılabilecek site: http://ipinfo.info/html/ip_checker.php)

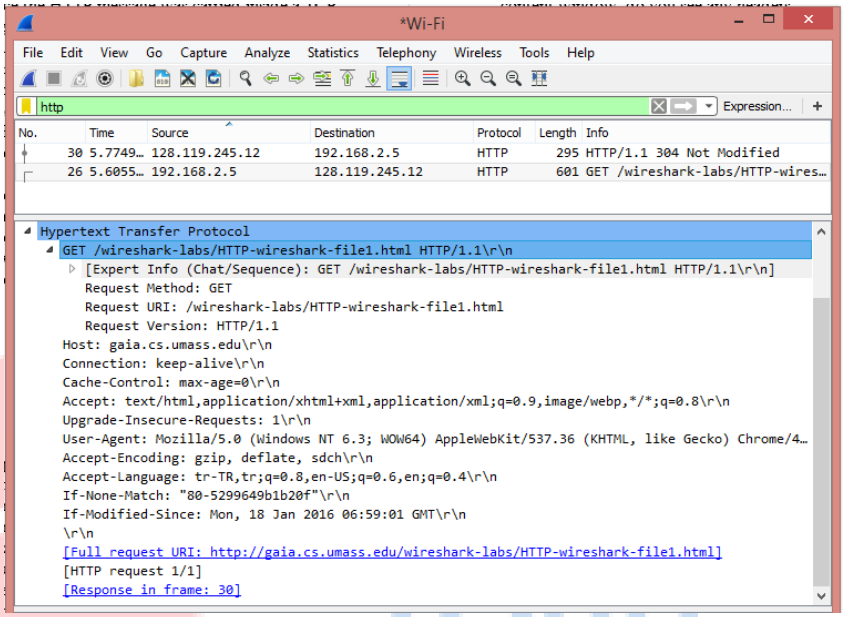
Şimdi tekrardan Wireshark ekranınıza dönünüz ve iki paketin "Source" (kaynak) ve "Destination" (hedef) alanlarını kontrol ediniz. Görüldüğü gibi iki iletişim client-server arasında IP'ler yardımıyla gerçekleşmiştir.

4. HTTP Get/Response

Resim 1'de bir istek mesajı ve bir cevap mesajı gördük. GET mesajı client olan bilgisayarımızdaki tarayıcıdan server'a gitti ve bunun karşılığında server bize bir cevap (response) mesajı döndürdü. Her iki mesajın info kısmında HTTP/1.1 yazısı görülmektedir. Bu HTTP versiyonunu göstermektedir.

Araştırma: HTTP/0.9 HTTP/1.0 HTTP/1.1 arasındaki farkları araştırınız.

Yakalanan iki mesaj arasından ilk önce GET mesaj içeriğini inceleyelim.



Resim 4: HTTP GET Mesajı

HTTP GET mesajında ilk önce request metodunu görüyorsunuz. Bu Web programcılığında server yapılan istek tipiyle alakalıdır. Bunun gibi başka istek tipleri de bulunmaktadır: GET, POST, HEAD, PUT, CONNECT. Fakat genellikle GET ve POST kullanılmaktadır.

URI: "Uniform Resourse Locator". Türkçeleştirirsek "Tekdüzen Kaynak Yerleştirici" demektir. Serverda hangi yoldaki HTML dosyasını istediğimizi URI belirliyor.

Host: Server-host aynı anlama gelebilmektedir. Server daha genel bir isimdir host ise daha çok web sitesi barındırılan serverlara verilen isimdir. Burada hostumuz "gaia.cs.umass.edu" server'ıdır.

Connection: Bağlantı tipini gösterir. "Keep-Alive" (hayatta tut) server ve client arası bağlantının kopmamasını sağlar. Eğer bir site büyükse server site dosyalarını parça parça gönderir fakat biz yine de alındığı kadarını anında görebiliriz. "Keep-Alive" seçeneği kapalı olursa site tamamen bilgisayarımıza inmeden gösterimi olmazdı.

Cache-Control: Her web sitesi bilgisayarımıza indirilir. Her web sitesinin bilgisayarımıza indirilmesi ağımıza belirli bir yük bindirir. Cache-Control bir web sitesinin bilgisayarımızda önbelleğe alınmasını ve site tekrar açılırken servar'dan değil de bilgisayarımızda var olan versiyonundan açılmasını sağlar. Bunun için "max-age=0" kısmına saniye cinsinden ne kadar sürede bir yenilenebileceğini yazıyoruz.

Accept: Accept başlığı client'ın server hangi tip dosyaları kabul edebildiğini söyler.

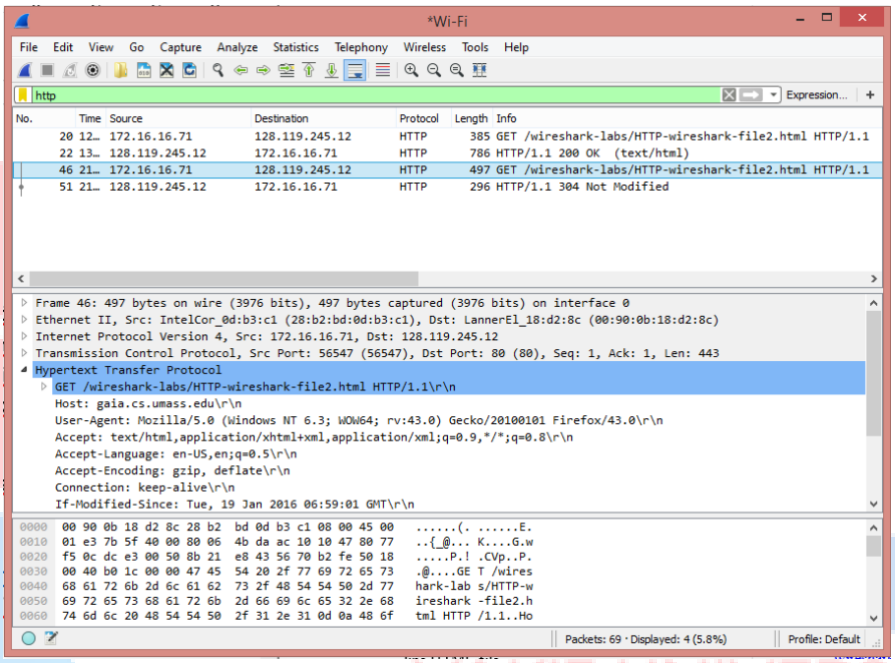
User-Agent: Servera hangi tarayıcıyı kullanarak ulaştığımızı söyler. Server ona göre o tarayıcıya bağlı faklı bir uygulama yapacaksa uygulayabilir ve ona cevap döndürür.

Accept-language: Client'ın kabul edeceği dili server'a gönderir. Server ona göre varsa uygun dilde bir sayfa döndürür.

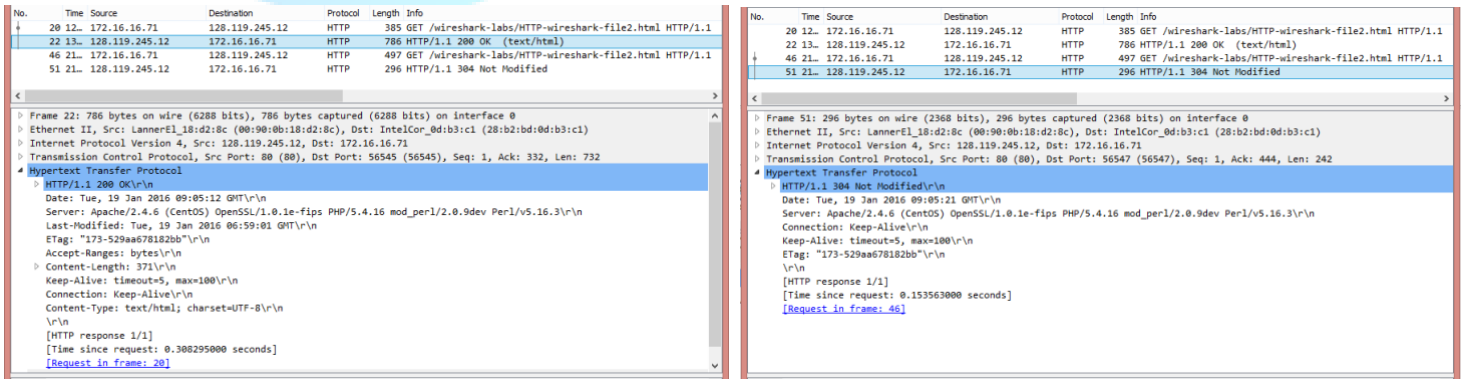
5. HTTP Durumsal Get/Response

Birçok tarayıcı gezdiğimiz web sayfalarını önbelleğe (cache) alır. Tarayıcımızda görüntülediğimiz her sayfa bilgisayarımıza indirilir daha sonra son kullanıcıya gösterilir. Eğer bir web sayfasında hiçbir değişiklik yoksa değişmemiş sayfayı tekrar tekrar indirmek gereksiz bir maliyet olacaktır. Buna engel olmak için değişiklik bulunmayan web sayfasını ya da en son bilgisayara indirilme zamanından çok fazla geçmeyenleri tekrar indirme işlemi yapılmaz. Bu da gereksiz web trafiğini engeller. Şimdi bunun testini yapalım.

1- Kullandığınız tarayıcının geçmişini siliniz.
2- Wireshark programını açınız ve dinlemeyi başlatınız.
3- Aşağıdaki linki tarayıcınıza girin ve Enter'a basın.
   http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
4- İki adet paketi gördüyseniz hemen tekrar tarayıcıdan sayfa yenilemesi yapınız. Sayfa yenilemesini bir iki defa daha deneyiniz.



Resim 5: HTTP Yenileme Mesajı

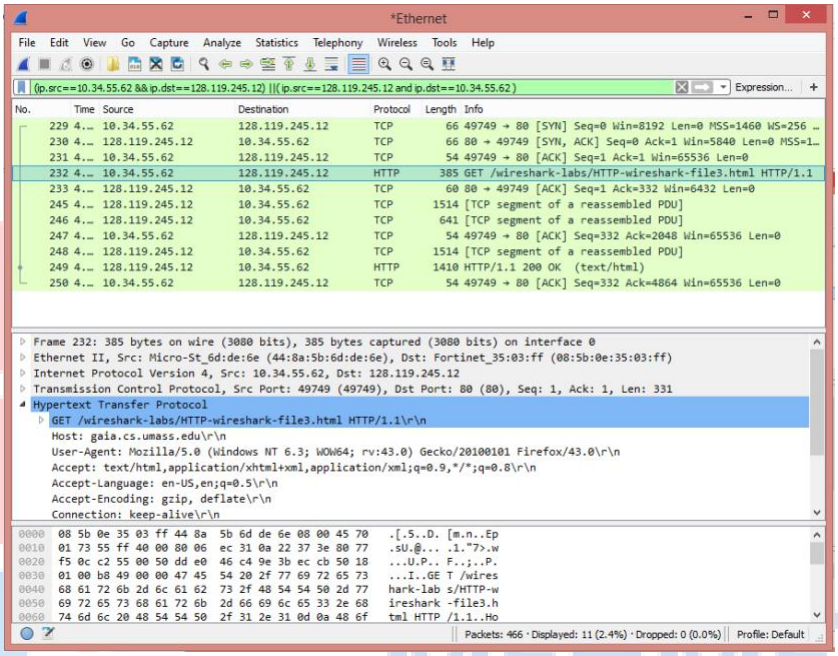


Resim 6: HTTP Yenileme Cevap Mesajları

6. HTTP Büyük Sayfalar

Daha önceki incelemelerimizde basit HTML sayfaları ile çalıştık. Şimdi büyük bir HTML sayfası nasıl bilgisayarımıza iniyor ona bakacağız.

1- Tarayıcınızı çalıştırınız ve önbelleğini (cache) temizleyiniz
2- Wireshark programını çalıştırınız, filtreye "http" yazıp, paket dinlemeyi başlatınız.
3- Aşağıdaki linki tarayıcınıza giriniz ve Enter'a basınız.
   http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
4- Paket dinlemeyi durdurunuz.



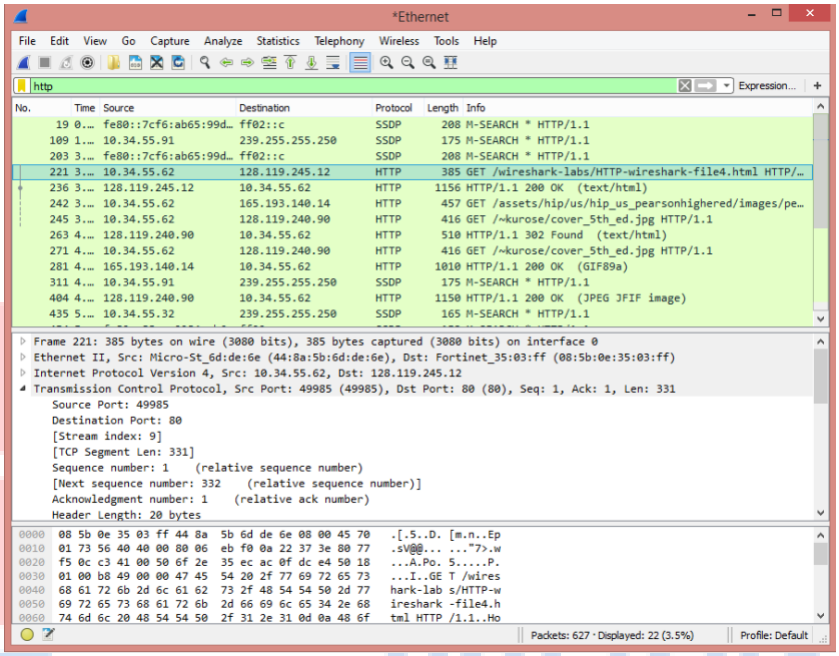
Resim 7: HTTP Büyük Belgeler

Resim 7'de görüldüğü gibi HTTP GET mesajından sonra birçok TCP paketi gelmiştir. Bunun nedeni server'ın cevap mesajı HTTP protokolü içerisine sığmamaktadır ve hatta tek bir TCP paketine de sığmamaktadır. Bundan dolayı gerekli miktarda TCP paketine bölünmüş ve client'a öyle gönderilmiştir.

7. HTTP Gömülü Objeler

Şimdi bir HTML sayfasında yazı haricinde başka objeler bulunduğunda paketimize neler geliyor onu görelim.

1- Tarayıcınızı çalıştırınız ve önbelleğini temizleyiniz
2- Wireshark programını çalıştırınız, filtreye "http" yazıp, paket dinlemeyi başlatınız.
3- Aşağıdaki linki tarayıcınıza giriniz ve Enter'a basınız.
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
4- Paket dinlemeyi durdurunuz.



Resim 8: HTTP Gömülü Objeler

Resim 8' de birden fazla HTTP GET işlemi görülmektedir. Bunun nedeni HTML sayfasında bulunan her bir nesne için bir GET isteğinde bulunulmasıdır.

Soru: HTTP GET isteklerine cevaplar faklı server adreslerinden gelmiştir, neden?

8. Lab Değerlendirme

8.1. HTPP protokolü nerelerde kullanılır?
8.2. HTTP detaylı alan adlarını ve anlamlarını yazınız.

9. Kaynaklar

9.1. http://www-net.cs.umass.edu/wireshark-labs/
9.2. Computer Networking: A Top-Down Approach, 6th edidition. J.F. Kurose and K.W. Ross