HTTP PROTOKOLÜ

HTTP NEDİR?

Açılımı Hypertext Transfer Protocol yani Hiper Metin Transfer Protokolü ' dür. HTTP protokolü network üzerinden web sayfalarının dijital ortamda görüntülenmesini sağlayan protokoldür. HTTP protokolü istemci yani internete bağlandığımız cihaz ile sunucu yani server arasındaki data alışveriş kurallarını belirler. Port olarak ise 80 portunu kullanır.

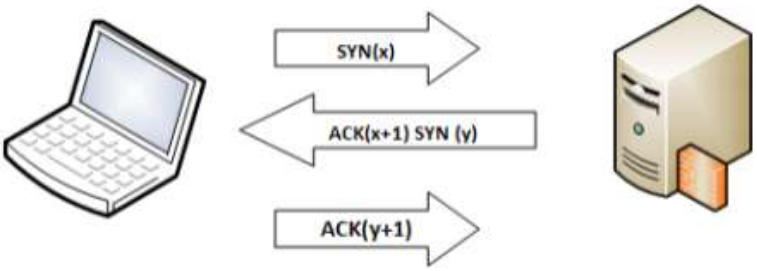
HTTP üzerinden iletilen iletiler birkaç nedenden dolayı başarıyla teslim edilemez:

1. Kullanıcı kaynaklı hatalar
2. Web tarayıcısının veya web sunucusu kaynaklı arızalar.
3. Web sayfalarının oluşturulmasından kaynaklanan sorunlar.
4. Geçici ağ hataları.

Bu hatalar meydana geldiğinde, protokol arızanın nedenini yakalar ve bir hata kodunu HTTP durum satırı / kodu olarak adlandırılan tarayıcıya geri gönderir. Hatalar, ne tür bir hata olduğunu belirtmek için belirli bir sayı ile gösterilir.

HTTP PROTOKOLÜ NASIL ÇALIŞIR?

HTTP oturumunun başlaması için, 3lü el sıkışmanın istemci ve server arasında tamamlanması gerekmektedir.



1- Kullanıcı, ulaşmak istediği web sitesinin sunucusuna bir adet SYN paketi gönderir.
2- Bu paketi alan sunucu, cevap verebilecek durumdaysa SYN+ACK paketini kullanıcıya gönderir.
3- Kullanıcı bu paketi alır ve ACK paketi "tamam" der.

3'lü el sıkışma tamamlandıktan sonra HTTP protokolü üzerinden trafik başlayacaktır.

Genel olarak anlatırsak öncelikle istemci sunucuya bir istek gönderir. Bu istek Internet Explorer, Google Chrome veya Mozilla Firefox gibi web browser'lar aracılığıyla iletilir. Sunucu bu isteği alır ve Apache veya IIS gibi web sunucu programları aracılığıyla cevap verir.

HTTP METOTLARI

GET: Sunucudan veri almak, sunucudaki kaynaklara erişmek için kullanılırlar. GET metodu ile sorgu metinleri URL içinde gönderilebilir. Bunun en önemli faydası kullanıcıların bookmark edebilmeleri ve aynı sorguyu içeren istekleri daha sonra gönderebilmelerini sağlaması ve tarayıcıda önceki sorguların "geri" tuşu ile veya tarayıcı geçmişinden çağrılarak aynı sayfalara ulaşabilmeleridir. Güvenlik açısından URL'lerin ekranda görüntüleniyor olması ve URL'in hedefine ulaşıncaya kadar ve hedef sunucu üzerinde iz kayıtlarında görülebilmesi gönderilen parametrelerin gizlilik ihtiyacı varsa sıkıntı yaratabilir. Bu nedenlerle hassas isteklerin GET ile gönderilmemelidir.

POST: Post metodu belirtilen kaynağa bir varlık (entity) göndermek için kullanılır. Bu metotla istek parametreleri hem URL içinde hem de mesaj gövdesinde gönderilebilir. Tarayıcılar geri butonuna basıldığında POST isteğinin mesaj gövdesinde yer alan parametreleri tekrar göndermek isteyip istemediğimizi sorarlar. Bunun temel nedeni bir işlemi yanlışlıkla birden fazla yapmayı engellemektir. Bu özellik ve de güvenlik gerekçeleriyle bir işlem gerçekleştirileceğinde POST metodunun kullanılması önerilir.

HEAD: GET metoduyla benzer işleve sahiptir ancak geri dönen yanıtta mesaj gövdesi bulunmaz (yani başlıklar ve içerikleri GET metoduyla aynıdır). Bu nedenle GET mesajı gönderilmeden önce bir kaynağın var olup olmadığını kontrol etmek için kullanılabilir.

TRACE: Bu metot ile istek-request değerinin değiştirilip değiştirilmediği kontrol edilir.

OPTIONS: Sunucuda kullanılabilecek http metotların listesi istenir.

PUT: Sunucuya veri eklemek için kullanılan bir yöntemdir. Kontrolü iyi yapılandırılmalıdır. Yoksa zararlı veri enjekte edilebilir.

DELETE: Sunucudaki herhangi bir veriyi silmek için kullanılır.

PATCH: Bir kaynağa kısmi değişiklikler uygulamak için kullanılır.

CONNECT: Bir proxy sunucu üzerinden başka bir sunucuya bağlanmak ve proxy sunucuyu bir tünel gibi kullanmak için kullanılır.

HTTP DURUM KODLARI

HTTP durum kodları, sunucu tarafından istemciye gönderilen cevaplarda isteğin nasıl sonuçlandığını belirten alanlardır. HTTP durum kodları genel kategorilere ayrılır ve bu kategorilere göre ilk sayıları değişir. Bu kategoriler aşağıdaki gibidir:

1xx: Bilgilendirme mesajlarını gösterir.

2xx: İsteğin başarılı olduğunu gösterir.

3xx: İsteğin başka sayfaya yönlendirileceğini gösterir.

4xx: Tarayıcı hatalarını gösterir.

5xx: Sunucu hatalarını gösterir.

COOKİE

Çerezler, istemci ile sunucu arasında etkileşimi daha etkili hale getirmek için kullanılır.

expires: Cookie ' nin browserda tutulacağı süreyi belirler. Opsiyonel bir değer olduğu için, belirtilmediği takdirde, oturum süresince browser belleğinde tutulur. Browserın kapanması ile birlikte de silinir.

domain: Cookie ‘ nin hangi domainler üzerinde çalıştığını ve ait olduğunu belirtir. Eğer belirtilmezse cookie ‘ nin alt domainlerinde aktif olmaz.

path: Cookie ‘ nin domain üzerinde bulunan hangi adres yolu üzerinde olacağını belirtir.

secure: Cookie'yi secure olarak bildirdiğimizde, eğer bağlantı tipi HTTPS yani güvenli bir bağlantı ise gönderilebileceğini belirtiyoruz. Bu opsiyonel değer set edilmediğinde güvenli ya da değil, domain ve path şartına uyan tüm isteklerle birlikte gönderilir. Secure tanımlı bir Cookie'yi ancak HTTPS olarak yapılmış bir isteğin yanıtında set edebiliriz.

HttpOnly: Javascript cookieleri değiştirebilen bir dildir. Bu büyük bir güvenlik açığı oluşturur. XSS zafiyeti sayesinde cookieler ele geçirilebilir. Cookie oluştururken HttpOnly kullanılırsa cookieler Javascript ile erişilemez hale gelir.

→Cookie oluşturma:

```
Set-Cookie: value[; expires=tarih][; domain=alan adı][; path=path][; secure]
```

→ Cookie belirteci içeren örnek bir HTTP yanıtı:

```
HTTP/1.1 200 OK
Content-Type: text/html
Set-Cookie: CookieName=CookieValue; path=/;
```

HTTP REQUEST

```
GET /hello.php?name=%3cscRipt%3enetsparker(0x000334)%3c%2fscRipt%3e HTTP/1.1
Host: php.testsparker.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: PHPSESSID=082e1539c6237346c459b0feb50459e8
Referer: http://php.testsparker.com/process.php?file=Generics/index.nsp
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
X-Scanner: Netsparker Cloud
```

GET →Kullanılan http metodu.

/hello.php?name=%3cscRipt%3enetsparker(0x000334)%3c%2fscRipt%3e →Sunucudan istenen sayfa

HTTP/1.1 →Kullanılan http sürümü.

Host →Erişilmekte olan URL'in tamamında bulunan hostname bilgisini içerir.

Accept →Kabul edilecek format(örneğin html ya da json gibi)

Accept-Encoding → Sunucunun kabul edebileceği format bilgisi

Accept-Language →Sitenin dil bilgisi

Cookie →Sunucunun istemciye verdiği ek parametreleri göndermek için kullanılır.

Referer →İsteğin hangi URL'den yapıldığını belirtmek için kullanılır.

User-Agent →Tarayıcı bilgisini gösterir.

HTTP RESPONSE

```
HTTP/1.1 200 OK
Server: Apache/2.2.8 (Win32) PHP/5.2.6
X-Powered-By: PHP/5.2.6
Content-Length: 2908
Content-Type: text/html
Date: Thu, 21 Jun 2018 15:32:45 GMT


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Netsparker Test Web Site -  PHP</title>
<link href="style.css" rel="stylesheet" type="text/css" media="screen" />
</head><body>
<div id="wrapper">
```

HTTP/1.1 200 OK → HTTP protokolünün kullanıldığı ve versiyon bilgisi. Ayrıca isteğin başarılı olup olmadığı dair sunucunun verdiği cevabın rakamsal değeri ile string değerini gösterir.

Server →Sunucu ile ilgili bilgi

X-Powered-By → Sitenin hangi dille yazıldığını gösterir.

Content-Lenght → Dönen cevabın body kısmının byte cinsinden değeri

Content-Type → Gösterilen html sayfasının uzantı bilgisi ve dil formatını gösterir.

Date → gelen istek sonucunda sunucunun istenilen nesneleri kendi dosya sisteminden çekip istemciye sunduğu anın tarihi ve zamanıdır.

Set-Cookie → İstemci-sunucu arasındaki çerez bilgisini tutar. Bir sonraki etkileşim için kullanılır.

SSL NEDİR?

SSL'in açılımı Secure Sockets Layer (güvenli giriş katmanı) dır. SSL, server ile alıcı iletişimi esnasında verilerin şifrelenerek yapılması işlemidir. En bilinen kullanımı ise, web sitesindeki veri alışverişi esnasında, server ile internet tarayıcısı arasındaki iletişimi şifrelenmesidir.

SSL, standart bir algoritmadan oluşmaktadır. Güvenli veri iletişimi için birçok web sitesi SSL teknolojisini kullanmaktadır. SSL işleminin çalışması için server tarafında bir anahtar ve alıcı tarafında çalışacak bir sertifika olması gerekmektedir.

Özellikleri

- Bağlantı gizlidir.
- Haberleşen uçların kimlikleri doğrulanabilir.
- Doküman arşivi oluşturulmasını kolaylaştırır.
- Bağlantı güvenilirdir. Mesaj akışı, mesajın bütünlüğünün kontrolünü de içerir.
- Veriyi gönderenin ve veriyi alanın doğru yerler olduğunu garanti eder.
- İletilen dokümanların tarih ve zamanını doğrular.

TLS NEDİR?

TLS (Transport Layer Security) veya Taşıma Katmanı Güvenliği, iki iletişim uygulaması arasındaki verileri şifreleyerek güvenli şekilde iletilmesini sağlayan güvenlik katmanıdır. İlk defa 1999 yılında içeriği belli olan TLS, kimlik doğrulamak için asimetrik şifreleme algoritmasını kullanır. TLS, yine Netscape tarafından geliştirilen farklı bir güvenlik katmanı olan SSL (Secure Sockets Layer) protokolünün daha gelişmiş ve güvenli hali olarak kabul görmektedir.

Özellikleri

- Tarayıcılar ile sunucular(server) arasında güvenli haberleşmeyi sağlar.
- Asimetrik kriptografi (asymmetric cryptography) algoritmasını kullanarak key(anahtar) üzerinden verileri şifreler.
- Veri şifrelemesinde HTTPS, SMTP, POP3, FTP gibi IP protokollerinin birçoğu tarafından desteklemektedir.

TLS Katmanlar

- TLS Record Protocol (TLS Kayıt Protokolü)
- TLS Handshake Protocol (TLS El Sıkışma Protokolü)

Handshake Protokolü ile sunucu ve kullanıcıların kimlik doğrulamaları yapılır. Handshake, veri iletişimi yapılmadan önce şifreleme algoritmaları ile şifreleme anahtarlarına izin verirken; Record Protocol bağlantının güvenli olmasını sağlar.

HTTPS

HTTPS, HTTP ile SSL/TLS iletişim kurallarının, şifrelenmiş iletişim ve güvenli tanımlama amacıyla birleşimidir. Varsayılan olarak 443'üncü porttan bağlantı kurar. HTTPS'in asıl amacı güvenli olmayan bir iletişim ağı üzerinden güvenli bir kanal oluşturmaktır. Bu yöntem hattı dinlemek isteyenlere karşı yeterli korumayı da sağlar. Verinin bütünlüğü ve gizliliği korunur. Araya giren saldırganlar trafiği görebilir ancak anlamlandıramaz. HTTPS ile http ' nin çalışma mantıkları güvenli protokol desteği hariç aynıdır.