云安全技术与应用

- 日照职业技术学院
- 电子信息工程系
- 赵娜

项目六

Http协议

目录

CONTENTS

1 HTTP请求与响应

2 Burpsuite的使用

01.

HTTP请求与响应

HTTP请求方法——URL

- URL（Uniform Resource Locator统一资源定位符）是互联网中标准的资源地址表示方法。
- 协议名://主机名(IP地址)/路径?参数名1=参数值1&参数名2=参数值2
- http://www.example.com.cn/test.php?user=teacher&pass=123
  - 协议名指明了访问网络资源所使用的协议，一般都为HTTP或HTTPS协议，默认为HTTP协议。
  - 在URL中如果指明路径，则是打开一个具体的网页或是某个具体的文件，如果路径省略，则是打开相应网站的首页。
  - 参数用于客户端向服务器传递数据，参数可以有多个，中间用&分隔。

HTTP请求方法——URL

- URL中的主机名，对应的就是Web服务器上的网站主目录。
  - 比如 http://www.test.com/在Web服务器中对应/var/www/html
  - 比如在Web服务器中某个网页文件的路径是/var/www/html/images/b.html，那么这个网页文件所对应的URL地址就是“http://www.test.com/images/b.html”。
- 网站首页是指客户端在访问网站时所默认打开的页面
  - 首页文件通常都是以index命名，如index.html、index.php等。
  - “http://www.test.com/”等同于“http://www.test.com/indext.html”，都表示要去访问Web服务器中的/var/www/html/index.html文件。

HTTP请求方法

- 请求方法
  - 客户端通过请求方法向服务器表达自己的意图。
  - 最常用的是Get方法，表示请求访问服务器中的某个资源。
  - 客户端还可以通过Get方法向服务器传递数据。

HTTP请求方法——Get方法

- Get方法的特点
  - Get方法通过URL向服务器发送数据。
  - 使用Get方法传递的数据会显示在浏览器地址栏中。
  - 由于浏览器对URL的长度会有限制，所以Get方法通常用于发送少量数据。

http://www.example.com.cn/test.php?user=teacher&pass=123

HTTP请求方法——Post方法

- Post方法的特点
  - Post方法将要发送的数据放在HTTP请求报文的正文中。
  - 数据不会显示在浏览器地址栏中，可以增强安全性。
  - 可以用于向服务器发送大量数据，如上传文件、提交留言等。

HTTP请求方法——PHP接收数据

- 在PHP中接收客户端数据主要是通过以下三种预定义变量：
  - $_GET：接收get方法传递的数据
  - $_POST：接收post方法传递的数据
  - $_REQUEST：接收get和post方法传递的数据

```php
<?php
    $username = $_POST['username'];
    $password = $_REQUEST['password'];
    echo "<p>用户名：".$username</p>";
    echo "<p>密　码：".$password</p>";
?>
```

HTTP请求方法——PHP接收数据测试

- HTML代码

```
<form action="UserLogin.php" method="post">
    <p>用户名: <input type="text" name="username"></p>
    <p>密  &nbsp码: <input type="password" name="password"></p>
    <p><input type="submit" name="submit" value="确定">   <input type="reset" value="重置"></p>
</form>
```

HTTP请求方法——CTF练习

- BugKu-Web-GET
- BUUCTF-Web-[极客大挑战2019]Havefun
- BugKu-Web-POST
- 攻防世界-Web新手区-get_post

HTTP请求方法——发送Post数据

- 利用Hackbar给网站发送Post数据
  - Hackbar是火狐浏览器的著名插件


```
$what=$_POST['what'];
echo $what;
if($what=='flag')
echo 'flag{****}';
flagflag{bugku_get_ssseint67se}
```

HTTP请求方法——Post方法

- 利用curl给网站发送Post数据
  - 用-X选项指定请求类型，并用-d选项指定要发送的参数。

```
root@kali:~# curl http://123.206.87.240:8002/post/ -X POST -d "what=flag"
$what=$_POST['what'];<br>
echo $what;<br>
if($what=='flag')<br>
echo 'flag{****}';<br>


flagflag{bugku_get_ssseint67se}root@kali:~#
```

HTTP请求和响应——HTTP协议

- 协议是网络中通信双方所应遵循的规则。
  - Web服务所使用的是HTTP超文本传输协议，它规定了在客户端（浏览器）和Web服务器（Web服务）之间互相通信应遵循的规则。
  - HTTP协议遵循请求（Request）/响应（Responses）模型，HTTP请求只能由客户端发起，Web服务器处理请求并返回相应的应答。

HTTP请求和响应——curl

- 利用curl工具查看HTTP请求和响应

```
root@kali:~# curl www.baidu.com
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-t
tent=IE=Edge><meta content=always name=referrer><link r
z/baidu.min.css><title>百度一下，你就知道</title></head
pper> <div class=s_form> <div class=s_form_wrapper> <di
270 height=129> </div> <form id=form name=f action=//ww
ut type=hidden name=ie value=utf-8> <input type=hidden
en name=rsv_idx value=1> <input type=hidden name=tn val
```

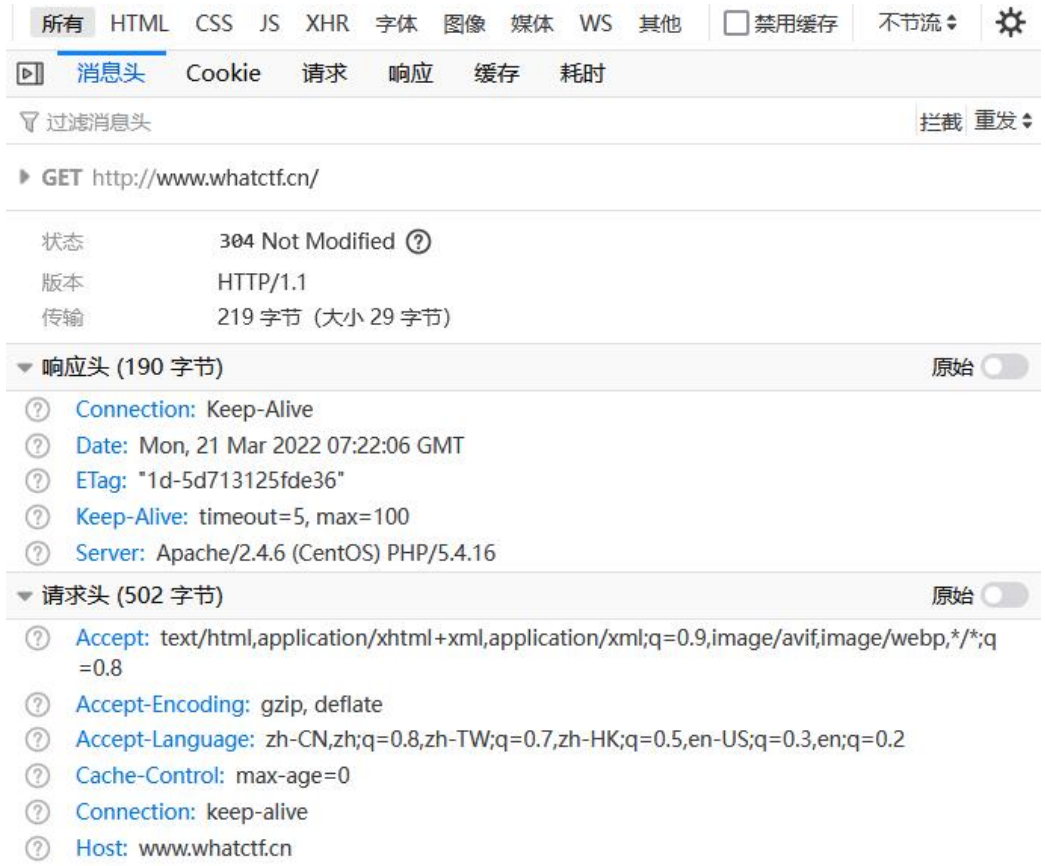
```
root@kali:~# curl -v www.baidu.com
*   Trying 61.135.185.32:80...
* TCP_NODELAY set
* Connected to www.baidu.com (61.135.185.32) port 80 (#0)
> GET / HTTP/1.1
> Host: www.baidu.com
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Cache-Control: private, no-cache, no-store, proxy-revalidate, no-transform
< Connection: keep-alive
< Content-Length: 2381
< Content-Type: text/html
< Date: Fri, 06 Nov 2020 23:44:27 GMT
< Etag: "588604c4-94d"
< Last-Modified: Mon, 23 Jan 2017 13:27:32 GMT
< Pragma: no-cache
< Server: bfe/1.0.8.18
< Set-Cookie: BDORZ=27315; max-age=86400; domain=.baidu.com; path=/
<
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;
tent=IE=Edge><meta content=always name=referrer><link rel=stylesheet type=te
z/baidu.min.css><title>百度一下，你就知道</title></head> <body link=#0000cc>
```

HTTP请求

HTTP响应

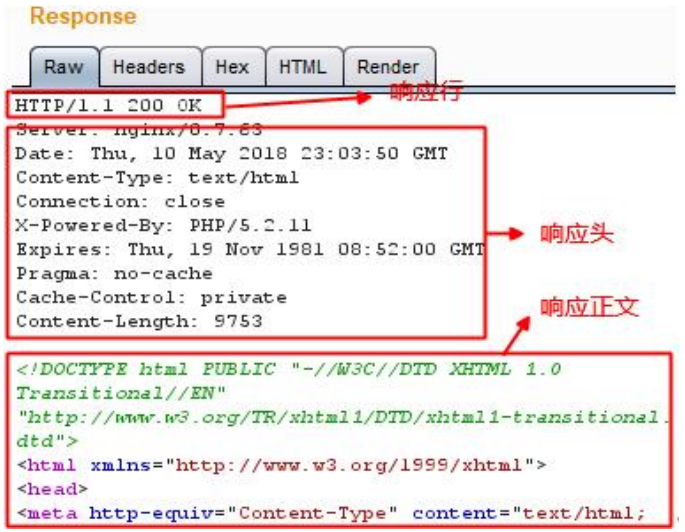
HTTP请求和响应——开发者工具

- 利用浏览器提供的开发者工具查看HTTP请求和响应



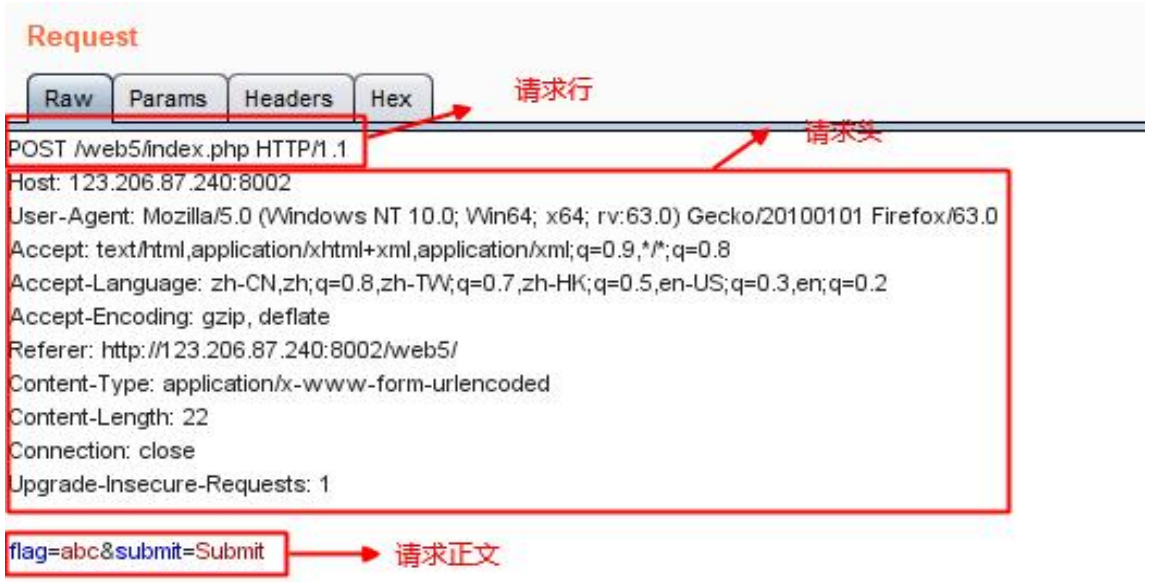
HTTP请求和响应报文——数据包结构

- HTTP协议的请求与响应报文都是由“首部header”和“主体body”两部分组成的。
  - 主体部分是请求和响应的数据。
  - 首部部分规定了请求和响应的内容格式，以及客户端和服务端的一些相关信息。



HTTP请求和响应报文——请求报文

- HTTP请求报文
  - HTTP请求报文由三部分组成：请求行、请求头、请求正文。
  - 在请求头和请求正文之间一般会有两个空行进行间隔。



HTTP请求和响应报文——请求行

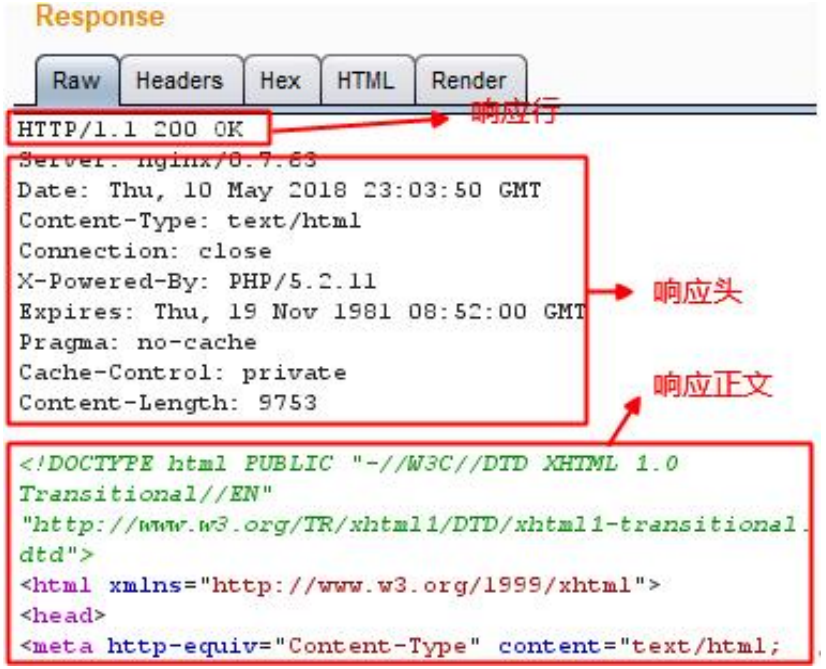
- 请求行由三部分组成：
  - 第一部分“GET”，表明该请求是采用GET方法；
  - 第二部分“/”，表明请求访问的页面，“/”是指网站根目录，也就是要访问网站的首页。它结合请求头的Host字段可以组成一个完整的请求URL：“www.51cto.com/”。
  - 第三部分“HTTP1.1”，表明所使用的HTTP协议版本，目前所使用的都是HTTP1.1版本。

HTTP请求和响应报文——请求头和请求正文

- 服务端根据请求头获取客户端的信息。
- 请求正文是可选的，它最常出现在POST请求方法中。

HTTP请求和响应报文——响应报文

- 响应行
  - 第一部分，HTTP/1.1，HTTP版本；
  - 第二部分，200，状态码；
  - 第三部分，OK，消息。
- 响应头
  - 包含了服务端的相关信息
  - 响应正文
  - 由服务器向客户端发送的HTML数据



HTTP请求和响应报文——响应头

- 响应头中的常规字段
  - Server：服务端所使用的Web服务名称。Server: Apache/2.4.6 (CentOS) PHP/5.4.16
  - Set-Cookie：服务器向客户端设置的Cookie。
  - Last-Modified：服务端资源的最后修改时间。
  - Location：重定向到另一个页面，通常配合302状态码使用。
  - Content-Length：body部分的长度（单位字节）

HTTP请求和响应报文——CTF练习

- BugKu-Web-头等舱
- 攻防世界-Web-baby_web

修改HTTP请求头——请求头中的常规字段

- Host，请求资源的主机和端口号。
- User-Agent，客户端操作系统和浏览器的信息。
  - 网站通过User-Agent字段可以来判断操作系统和浏览器的类型。
  - 网站通过UA可以来判断访问是否合法，是用户访问还是程序访问等。
- Accept，客户端可以接收哪些MIME类型的消息。
  - MIME类型用来设定某种扩展名文件的打开方式。
  - Accept：text/html，表示客户端希望接收HTML文本。
  - Accept：text/plain，表示客户端只能接收纯文本，服务器不能向它发送图片、视频等。

修改HTTP请求头——请求头中的常规字段

- Accept-Language，指定客户端可以接收的语言。
  - 如果请求消息中没有设置这个域，默认是任何语言都可以接收。
  - 该项也可以作为用户地区的判断依据。
- Referer，HTTP来源地址，用来表示从哪儿链接到当前页面。
  - 网站通过Referer可以来判断用户的访问来源。
- Cookie，客户端发给服务器证明用户状态的信息，用来表示请求者的身份。

修改HTTP请求头——伪造客户端IP地址

- X-FORWARDED-FOR，简称XFF。
  - 通过该字段可以指定客户端的IP地址。
  - 通常都是将客户端地址伪造成127.0.0.1，从而实现本地访问。

修改HTTP请求头——CTF练习

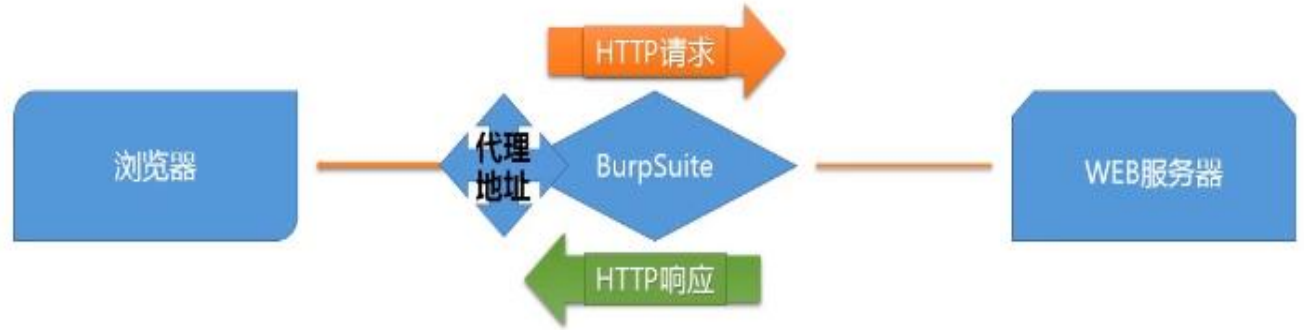
- BugKu-Web-你从哪里来
- BugKu-Web-程序员本地网站
- 攻防世界-Web-xff_referrer
- 攻防世界-Web-cookie
- BUUCTF-Web-[极客大挑战2019]Http

02.

Burpsuite的使用

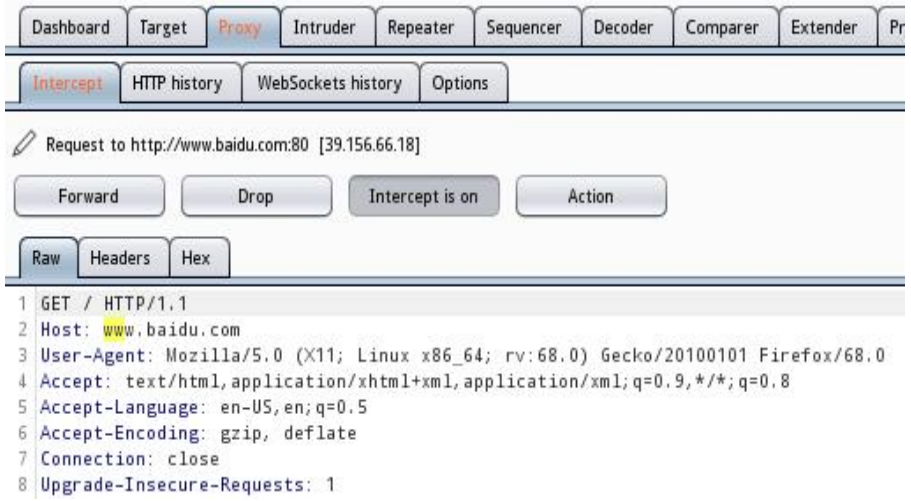
Burpsuite

- 利用Burpsuite拦截HTTP数据
  - Burpsuite可谓Web安全的神器，学习Web安全，必须精通Burpsuite。
  - BurpSuite最主要的功能是拦截客户端与服务器之间传输的数据，然后对数据进行修改并再次发送，从而完成攻击过程。



Burpsuite

- 被Burpsuite拦截的数据会显示在proxy模块中
  - Forward表示将数据包直接转发出去。
  - Drop表示将数据包丢弃，不向外转发。
  - Intercept is on表示开启了拦截功能。
  - Intercept is off表示关闭了拦截功能。



Burpsuite

- Burpsuite的repeater模块
  - 在Proxy模块中拦截的数据通常会发送给其它模块做进一步处理，其中最常发到Repeater模块。
  - 在Repeater模块的左侧窗口中可以对拦截的HTTP请求数据进行修改，然后再发送出去，在右侧窗口中将显示接收的响应数据。