
160
140
120
100
80
60
40
20
0
113,67
83,4843
102,4293
143,4768
124,6581
140,8245
29,049
123,3951
79,8216
Information technology
Telecommunications
Wood
Steel industry
Oil and Gas
Finance
Air transport
Chemical production
Industrials
Nuclear energy industry
Securities
Electronic commerce
Electronic commerce
6,73%
Information technology
9,00%
Securities
9,77%
Nuclear energy industry
2,30%
Industrials
11,15%
Chemical production
9,87%
Air transport
6,32%

Web安全基础

01 Web应用安全发展与介绍

02 HTTP协议与会话管理

03 Web应用的组成与网页的渲染

Web应用安全学习路线

安全漏洞

- （SQL注入、命令注入、代码注入、XXE、XSS、CSRF、SSRF、逻辑漏洞、文件上传漏洞、解析漏洞、文件包含漏洞 、反序列化漏洞、SSTI模板注入）
- （JAVA：rmi远程方法调用、jndi、表达式注入；PHP：弱类型、魔术字符串、变量覆盖；JS：原型链污染；ruby：open命令注入；）

编程语言（PHP、 SQL、 JS、Python、JAVA）

操作系统

网络基础（TCP、HTTP）

Web应用安全发展与介绍

Web安全跟随着Web应用的发展也不断发展着：

- Web 1.0时代，更多被关注的时服务器端的脚本的安全问题，如SQL注入等
- Web 2.0时代，2005年Samy蠕虫的爆发震惊了世界，Web安全主战场由服务器端转换到浏览器
- SQL注入和XSS的出现分别是Web安全史上的两个里程碑

Web应用安全发展与介绍

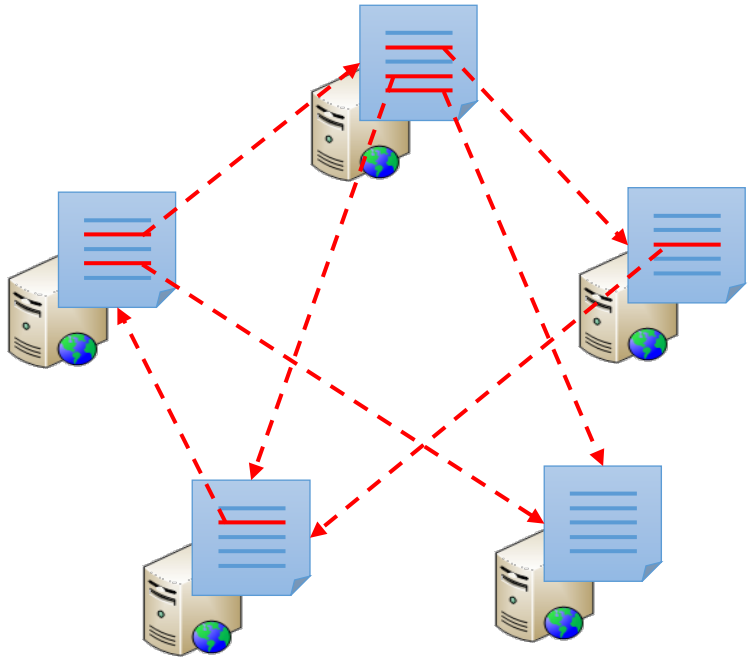
安全的本质是信任问题

- 由于信任，正常处理用户恶意的输入导致问题的产生
- 非预期的输入

HTTP协议简介

- **什么是超文本(HyperText)?**

包含有超链接(Link)和各种多媒体元素标记(Markup)的文本。这些超文本文件彼此链接，形成网状(Web)，因此又被称为页(Page)。这些链接使用URL表示。最常见的超文本格式是超文本标记语言HTML。



HTTP协议简介

- 什么叫URL?

让我们来看一个URL（统一资源定位器）

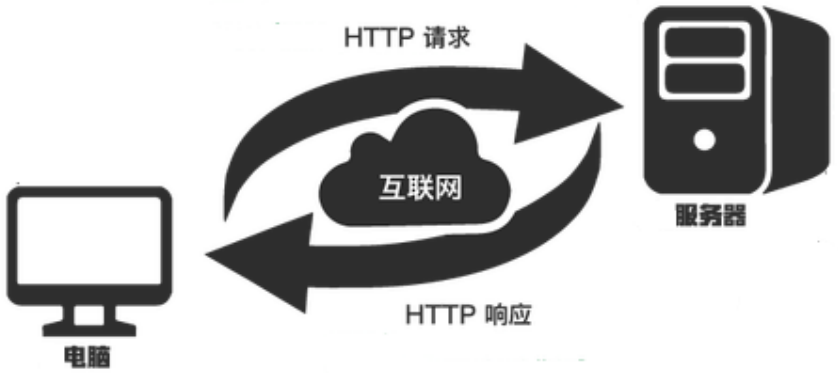
scheme://login:password@address:port/path/to/resource/?query_string#fragment

编号	说明
1	协议名称
2	层级URL的标记符号（固定不变，语法规定）
3	访问资源需要的凭证信息（可选）
4	从哪个服务器获取数据
5	需要连接的端口号（默认80，可选）
6	指向资源的层级文件路径
7	查询字符串
8	片段ID

HTTP协议简介

什么是超文本传输协(HTTP)?

是一种按照URL指示，将超文本文档从一台主机(Web服务器)传输到另一台主机(浏览器)的应用层协议，以实现超链接的功能。

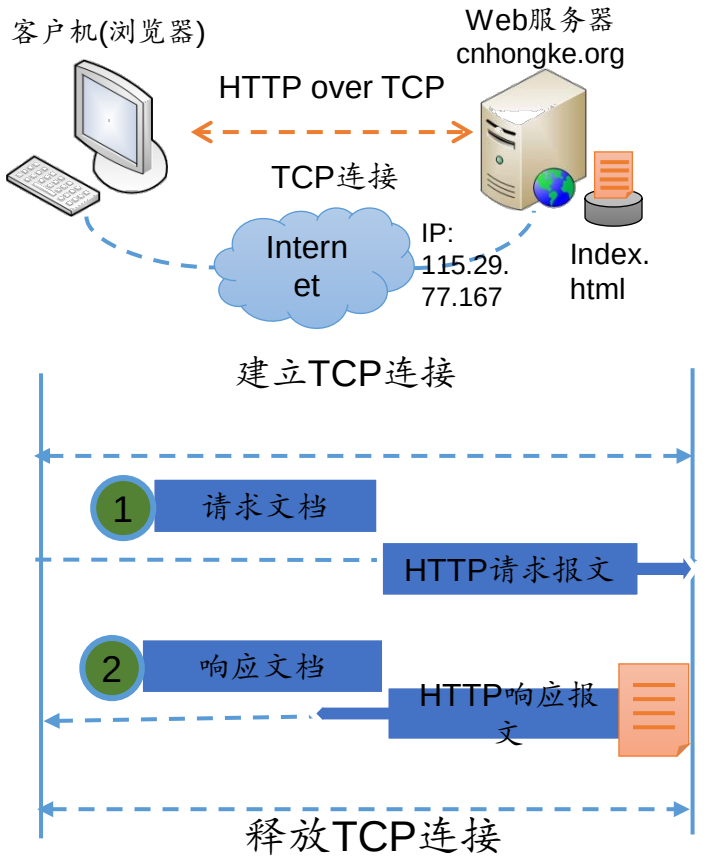


HTTP协议简介

请求/响应交互模型

当用户在浏览器地址栏输入URL为http://cnhongke.org/index.html的链接后，浏览器和Web服务器执行以下动作：

1. 浏览器分析超链接中的URL
2. 浏览器向DNS请求解析cnhongke.org的IP地址
3. DNS将解析出的IP地址115.29.77.67返回给浏览器
4. 浏览器与服务器建立TCP连接(80端口)
5. 浏览器请求文档：GET /index.html
6. 服务器处理请求并发回一个响应，将文档 index.html发送给浏览器
7. 释放TCP连接
8. 浏览器渲染显示index.html中的内容



HTTP协议简介

HTTP的连接方式和无状态性

- **非持久性连接**

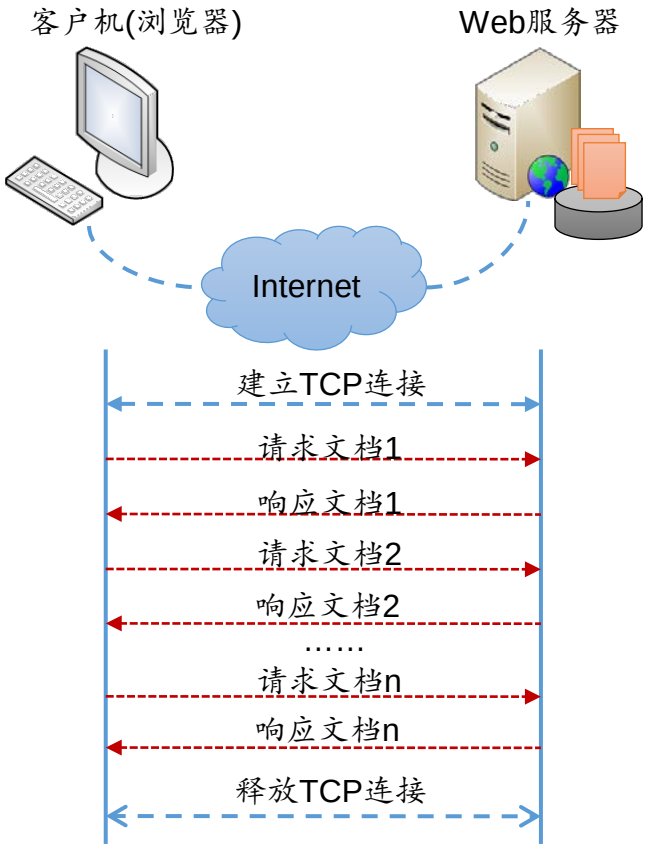
即浏览器每请求一个Web文档，就创建一个新的连接，当文档传输完毕后，连接就立刻被释放。
> HTTP1.0、HTTP0.9采用此连接方式。
> 对于请求的Web页中包含多个其他文档对象（如图像、声音、视频等）的链接的情况，由于请求每个链接对应的文档都要创建新连接，效率低下。

- **持久性连接**

即在一个连接中，可以进行多次文档的请求和响应。服务器在发送完响应后，并不立即释放连接，浏览器可以使用该连接继续请求其他文档。连接保持的时间可以由双方进行协商。HTTP1.1默认使用持久性连接

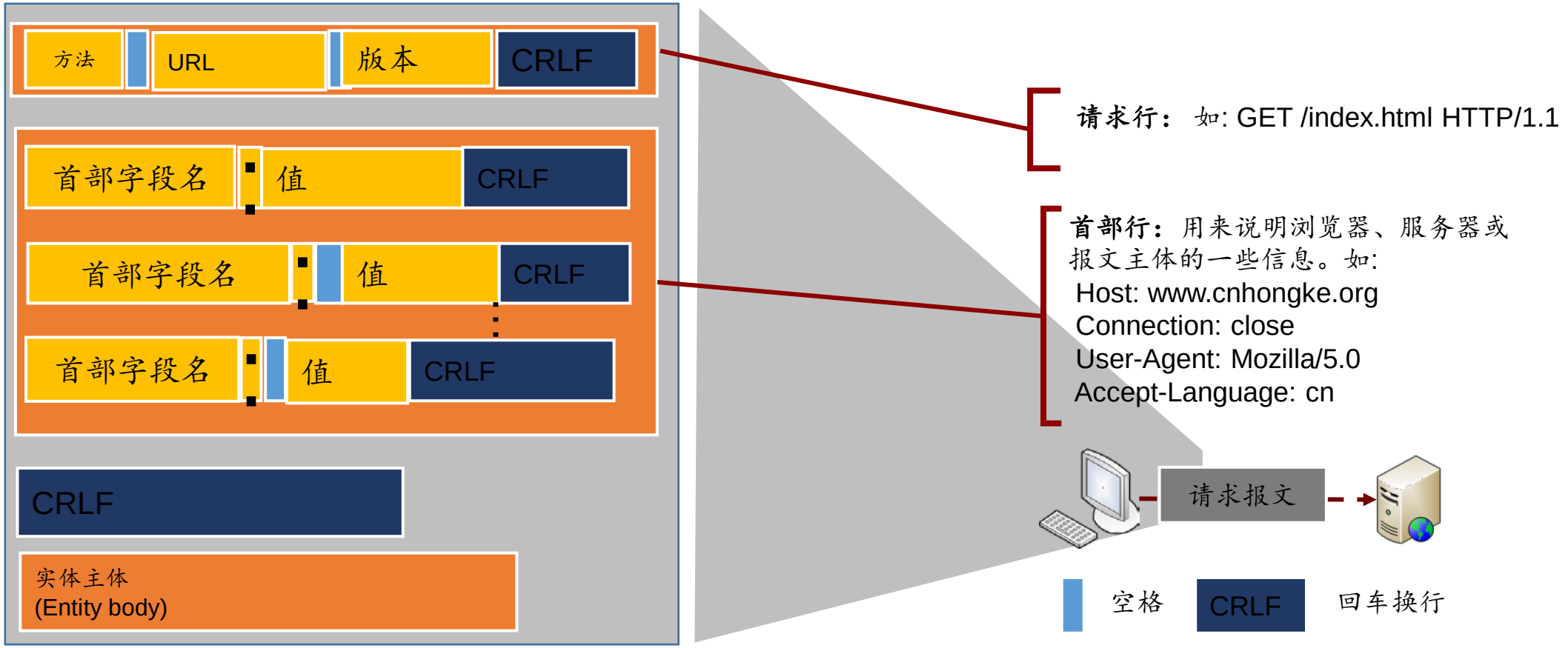
- **无状态性**

是指同一个客户端(浏览器)第二次访问同一个Web服务器上的页面时，服务器无法知道这个客户曾经访问过。HTTP的无状态性简化了服务器的设计，使其更容易支持大量并发的HTTP请求。



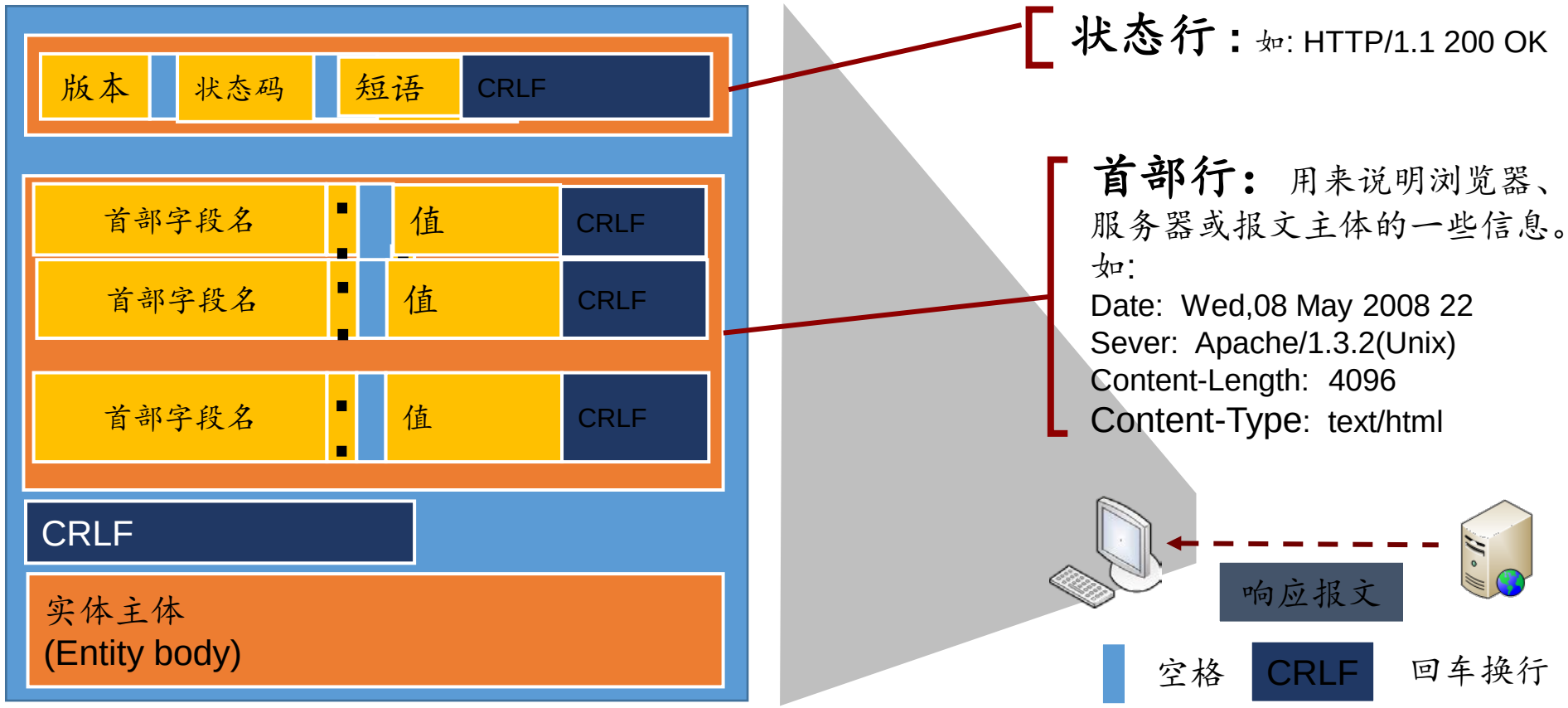
HTTP报文结构

请求报文 即从客户端(浏览器)向Web服务器发送的请求报文。报文的所有字段都是ASCII码。



HTTP报文结构

响应报文　即从Web服务器到客户机(浏览器)的应答。报文的所有字段都是ASCII码。



HTTP报文结构

请求报文中的方法

方法(Method)是对所请求对象所进行的操作,也就是一些命令。请求报文中的操作有:

方法(操作)	含义	方法(操作)	含义
GET	请求读取一个Web页面	HEAD	请求读取一个Web页面的首部
POST	附加一个命名资源(如Web页面)	PUT	请求存储一个Web页面
DELETE	删除Web页面	TRACE	用于测试，要求服务器送回收到的请求
CONNECT	用于代理服务器	OPTION	查询特定选项

HTTP报文结构

响应报文中的状态码

状态码(Status-Code)是响应报文状态行中包含的一个3位数字，指明特定的请求是否被满足，如果没有满足，原因是什么。状态码分为以下五类：

状态码	含义	例子
1xx	通知信息	仅在与HTTP服务器沟通时使用 100(“Continue”)
2xx	成功	成功收到、理解和接受动作 200(“OK”)、201(“Created”)、204(“No Content”)
3xx	重定向	为完成请求，必须进一步采取措施 301(“Moved Permanently”)、303(“See Other”)、304(“Not Modified”)、307(“Temporary Redirect”)
4xx	客户错误	请求包含错误的语法或不能完成 400(“Bad Request”)、401(“Unauthorized”)、403(“Forbidden”)、404(“Not Found”)、405(“Method Not Allowed”)、406(“Not Acceptable”)、409(“Conflict”)、410(“Gone”)
5xx	服务器错误	服务器不能完成明显合理的请求 500(“Internal Server Error”)、503(“Service Unavailable”)

HTTP报文结构

首部字段或消息头

头(header)	类型	说明
User- Agent	请求	关于浏览器和它平台的信息，如Mozilla5.0
Accept	请求	客户能处理的页面的类型，如text/html
Accept-Charset	请求	客户可以接受的字符集，如Unicode-1-1
Accept-Encoding	请求	客户能处理的页面编码方法，如gzip
Accept-Language	请求	客户能处理的自然语言，如en(英语)，zh-cn(简体中文）
Host	请求	服务器的DNS名称。从URL中提取出来，必需。
Authorization	请求	客户的信息凭据列表
Cookie	请求	将以前设置的Cookie送回服务器器，可用来作为会话信息
Date	双向	消息被发送时的日期和时间
Server	响应	关于服务器的信息，如Microsoft-IIS/6.0
Content-Encoding	响应	内容是如何被编码的（如gzip)
Content-Language	响应	页面所使用的自然语言
Content-Length	响应	以字节计算的页面长度
Content-Type	响应	页面的MIME类型
Last-Modified	响应	页面最后被修改的时间和日期，在页面缓存机制中意义重大
Location	响应	指示客户将请求发送给别处，即重定向到另一个URL
Set-Cookie	响应	服务器希望客户保存一个Cookie

会话管理

什么是会话？

可以简单的理解为：用户开一个浏览器，点击多个超链接，访问服务器多个web、资源，然后关闭浏览器，整个过程称之为一个会话。

会话技术要解决的问题？

如何保存会话中的数据并实现多次请求或会话中共享数据的问题：对每个用户来说可以共享多次请求中产生的数据，且不同用户产生的数据要相互隔离会话技术的两种实现方式

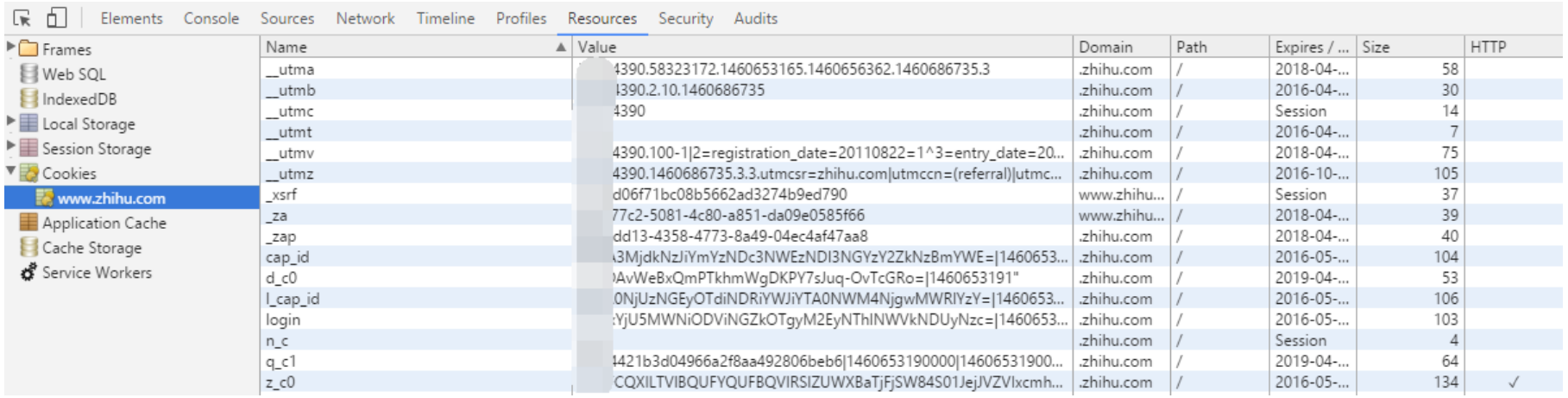
会话技术

Cookie（客户端技术）

程序把每个用户的数据以cookie的形式写给用户各自的浏览器。当用户使用浏览器，再去访问服务器中的web资源时，就会带着各自的数据（cookie）去,这样web资源处理的就是用户各自的数据了

HTTP协议与会话管理

Cookie示例



字段	说明
Name	Cookie名称
Value	Cookie的值
Domain	用于指定Cookie的有效域
Path	用于指定Cookie的有效URL路径
Expires	用于设定Cookie的有效时间
Secure	如果设置该属性，仅在HTTPS请求中提交Cookie
Http	其实应该是HttpOnly，如果设置该属性，客户端JavaScript无法获取Cookie值

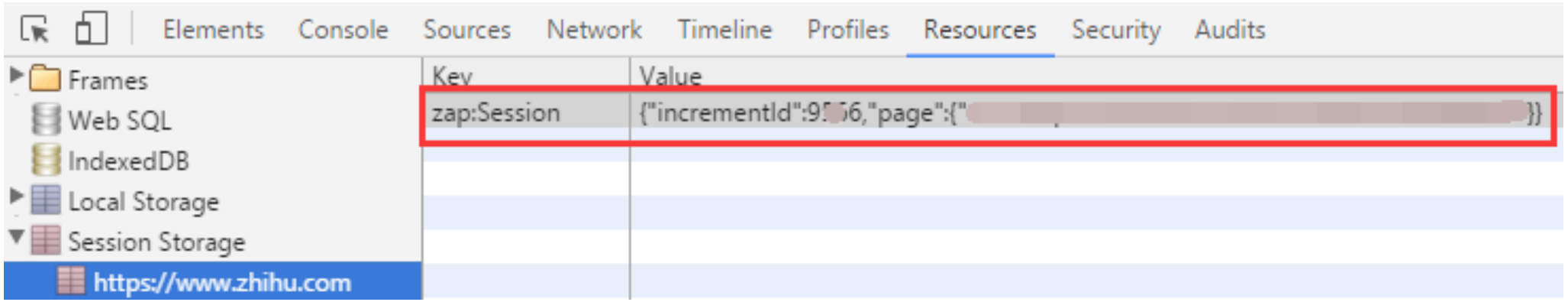
会话技术

HttpSession（服务器端技术）

服务器在运行时可以为每一个用户的浏览器创建一个其独享的HttpSession对象，由于session为用户浏览器独享，所以用户在访问服务器的web资源时，可以把各自的数据放在各自的session中。当用户再去访问服务器中的其它web资源时，其它web资源再从用户各自的session中取出数据为用户服务

会话管理

Session



字段	说明
Key	Session的key
Value	Session对应key的值

HTTP协议与会话管理

Session与Cookie的区别

Cookie的数据保存在客户端浏览器，Session保存在服务器

服务端保存状态机制需要在客户端做标记，所以Session可能借助Cookie机制

Cookie通常用于客户端保存用户的登录状态

Web服务组件

层级划分	WEB服务组件	
静态层	Web前端框架：Bootstrap/jQuery/HTML5框架	跨站脚本攻击
脚本层	Web应用：BLOG/CMS/BBS	远程命令执行
	Web开发框架：ThinkPHP/Struts2/Django	
	Web服务端语言：PHP/JSP/.NET	
服务层	Web容器：Apache/IIS/Nginx/Tomcat/Weblogic	
数据层	存储：数据库存储/内存存储/文件存储	SQL注入
系统层	操作系统：Windows/Linux/UNIX	

"><script>alert(/xss./)</script>

eval($_REQUEST['x']);

%c0.%c0./%c0.%c0./%c0.%c0./%c0.%c0./%20

' union select user, pwd, 3 from users--

; cat /etc/passwd;