# UC Day16

## 预习课

# 预习内容

## UDP协议

# UDP协议

# UDP协议

- UDP不提供客户机与服务器的连接
  - UDP的客户机与服务器不必存在长期关系。一个UDP的客户机在通过一个套接字向一个UDP服务器发送了一个数据报之后，马上可以通过同一个套接字向另一个UDP服务器发送另一个数据报。同样，一个UDP服务器也可以通过同一个套接字接收来自不同客户机的数据报
- UDP不保证数据传输的可靠性和有序性
  - UDP的协议栈底层不提供诸如确认、超时重传、RTT估算以及序列号等机制。因此UDP数据报在网络传输的过程中，可能丢失，也可能重复，甚至重新排序。应用程序必须自己处理这些情况

知识讲解

# UDP协议

- UDP不提供流量控制
  - UDP的协议栈底层只是一味地按照发送方的速率发送数据，全然不顾接收方的缓冲区是否装得下
- UDP是全双工的
  - 在一个UDP套接字上，应用程序在任何时候都既可以发送数据也可以接收数据



知识讲解

# 直播课见

# UC

## C/C++教学体系

# 目录

# UDP函数

# UDP函数

知识讲解

- #include <sys/socket.h>
- ssize_t recvfrom(int sockfd, void* buf, size_t count, int flags,
  struct sockaddr* src_addr, socklen_t* addrlen);
  - 功能：从哪里接收数据
  - 参数：前四个参数和函数recv相同
    src_addr：输出源主机的地址信息
    addrlen：输入输出源主机的地址信息的字节数。
  - 返回值：成功返回实际接收的字节数，失败返回-1

# UDP函数

知识讲解

- #include <sys/socket.h>
- ssize_t sendto(int sockfd, void const* buf, size_t count, int flags,
  struct sockaddr const* dest_addr, socklen_t addrlen);
  - 功能：发送数据到哪里
  - 参数：前四个参数和函数send相同
    dest_addr：目的主机的地址信息。
    addrlen：目的主机的地址信息的字节数。
  - 返回值：成功返回实际发送的字节数，失败返回-1

# 编程模型

# UDP编程模型

- UDP编程模型

知识讲解

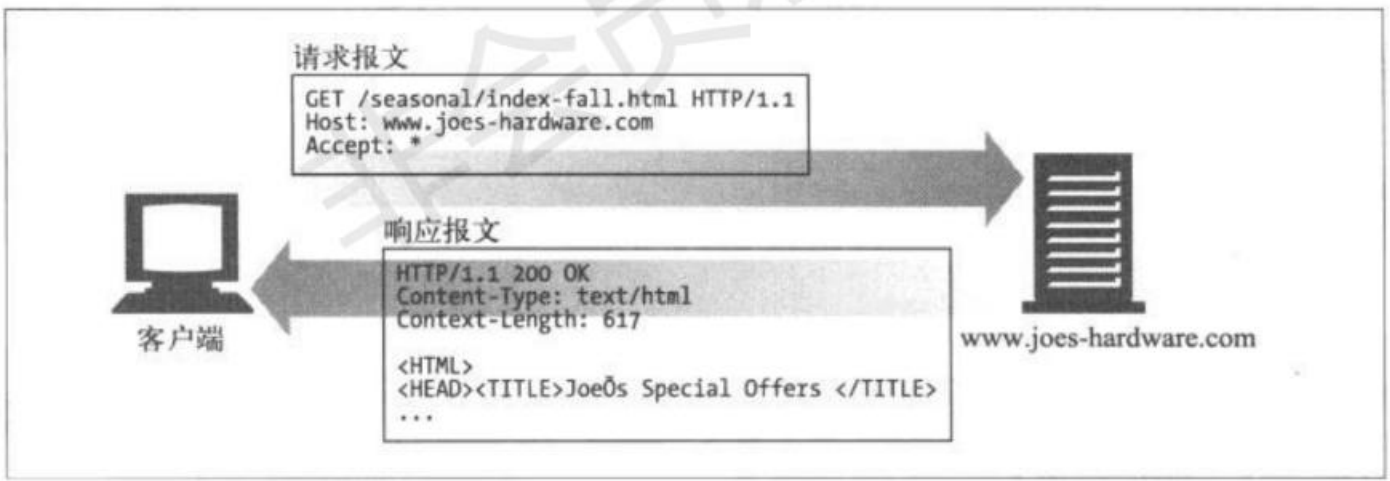
| 步骤 | 服务器 | | 客户机 | | 步骤 |
|---|---|---|---|---|---|
| 1 | 创建套接字 | socket | socket | 创建套接字 | 1 |
| 2 | 准备地址结构 | sockaddr_in | sockaddr_in | 准备地址结构 | 2 |
| 3 | 绑定地址 | bind | —— | —— | —— |
| 4 | 接收请求 | recvfrom | sendto | 发送请求 | 3 |
| 5 | 发送响应 | sendto | recvfrom | 接收响应 | 4 |
| 6 | 关闭套接字 | close | close | 关闭套接字 | 5 |

# HTTP协议

# HTTP协议

知识讲解

- 我们在通过浏览器获取网络资源的过程中，一直在遵循HTTP协议。客户端终端和服务器终端请求和应答的标准，客户端发起一个HTTP请求到服务器上指定端口，默认80，应答服务器上存储着一些资源，比如HTML文件和图像等



# HTTP协议

- 浏览器地址栏中输入URL，按下回车后会经历的流程：
  - 1、浏览器向DNS服务器请求解析该URL中的域名所对应的IP地址。
  - 2、根据解析后的IP地址和默认端口80和服务器建立TCP连接。
  - 3、浏览器发出HTTP请求
  - 4、服务器对浏览器的请求作出响应

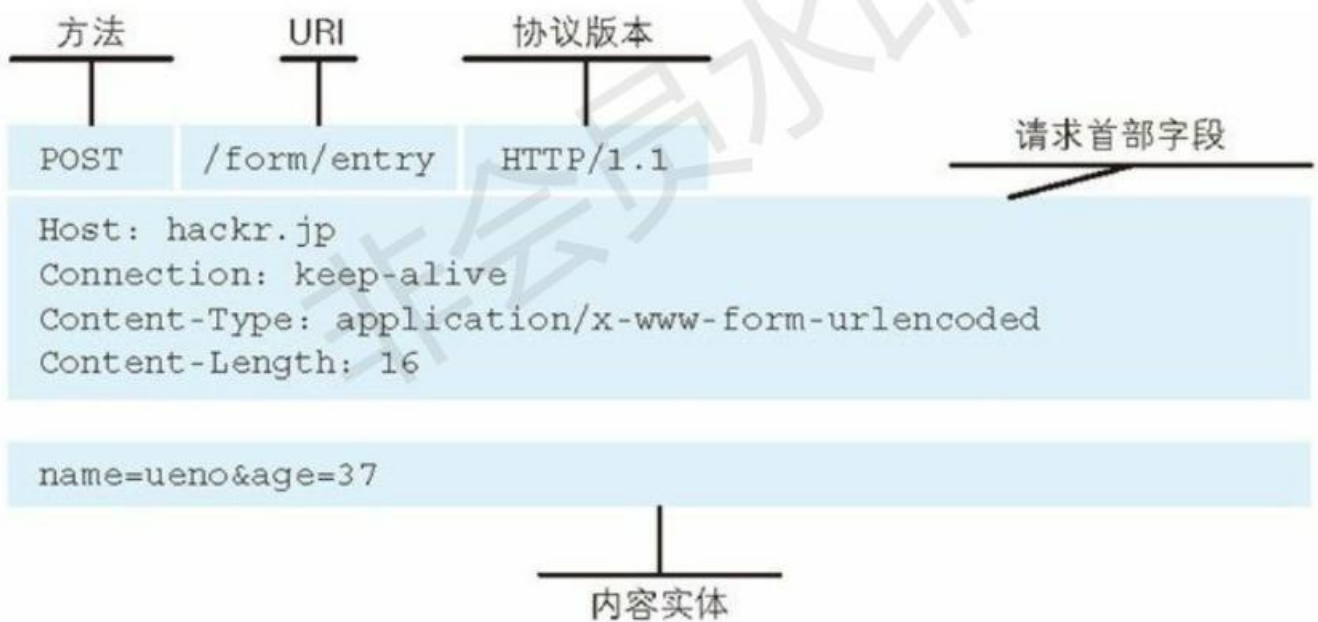
知识讲解



# HTTP协议

- HTTP的请求和响应

知识讲解



# HTTP协议

- HTTP的请求和响应
  - 请求行由请求方法字段、URL字段和HTTP协议版本字段3个字段组成，它们用空格分隔。例如，GET /index.html HTTP/1.1。
  - HTTP1.0定义了三种请求方法： GET, POST 和 HEAD方法。
  - HTTP1.1新增了五种请求方法：OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。
  - 其中GET是最常用的请求方法，用来获取服务器的数据

知识讲解

# HTTP协议

知识讲解

- HTTP的请求和响应
  - 请求头部由关键字/值对组成，每行一对，关键字和值用英文冒号“:”分隔。请求头部通知服务器有关于客户端请求的信息，典型的请求头有：
  - Accept：告诉服务器自己接受什么类型的介质，*/*表示任何类型，type/*表示该类型下的所有子类型，
  - Host：客户端指定自己想访问的web服务器的域名
  - User_Agent：浏览器表名自己的身份，是那种浏览器
  - Referer：浏览器web服务器表名自己是从哪个网页URL获得点击当前请求中的网址
  - Connection：表示是否需要持久连接

# HTTP协议

- HTTP的请求和响应
  - HTTP响应也由三个部分组成，分别是：状态行、响应头、空行、响应正文

知识讲解

# HTTP协议

知识讲解

- HTTP的请求和响应
  - 当浏览者访问一个网页时，浏览者的浏览器会向网页所在服务器发出请求。当浏览器接收并显示网页前，此网页所在的服务器会返回一个包含HTTP状态码的信息头（server header）用以响应浏览器的请求，常见状态码有：
  - 200 OK：客户端请求成功。
  - 400 Bad Request：客户端请求有语法错误，不能被服务器所理解。
  - 403 Forbidden：服务器收到请求，但是拒绝提供服务。
  - 404 Not Found：请求资源不存在，举个例子：输入了错误的URL。
  - 503 Server Unavailable：服务器当前不能处理客户端的请求，一段时间后可能恢复正常

# HTTP协议

知识讲解

- HTTP的请求和响应
  - 当浏览者访问一个网页时，浏览者的浏览器会向网页所在服务器发出请求。当浏览器接收并显示网页前，此网页所在的服务器会返回一个包含HTTP状态码的信息头（server header）用以响应浏览器的请求，常见状态码有：
  - 200 OK：客户端请求成功。
  - 400 Bad Request：客户端请求有语法错误，不能被服务器所理解。
  - 403 Forbidden：服务器收到请求，但是拒绝提供服务。
  - 404 Not Found：请求资源不存在，举个例子：输入了错误的URL。
  - 503 Server Unavailable：服务器当前不能处理客户端的请求，一段时间后可能恢复正常

# HTTP

- HTTP的请求和响应
  - 消息报头一般包括以下内容：
  - Date：响应时间
  - Content-Type：响应类型
  - Content-Length：响应数据大小
  - Connection：连接状态

# 谢谢

# UC Day016

## 复习课

# DDoS攻击

# DDoS攻击

- 拒绝服务(Denial of Service, DoS)攻击是一种简单有效的攻击方式。它通过大量消耗服务器主机的系统资源，阻碍其提供正常网络服务，达到攻击目的
- TCP SYN泛洪攻击是一种典型的拒绝服务攻击，其攻击发起者利用TCP协议漏洞，模拟众多服务请求，使服务器主机疲于奔命，无法响应正常服务请求

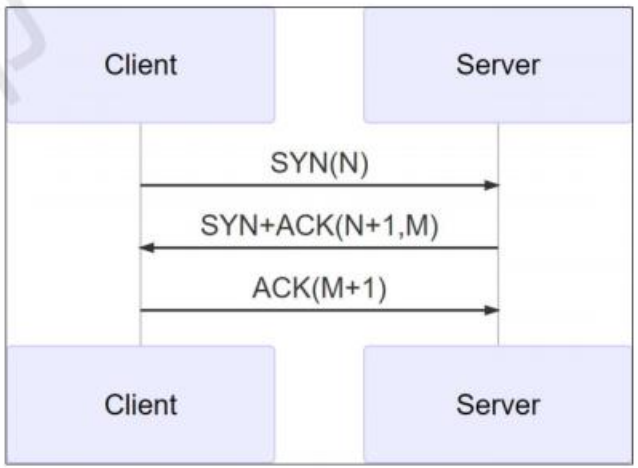
知识讲解



# DDoS攻击

知识讲解

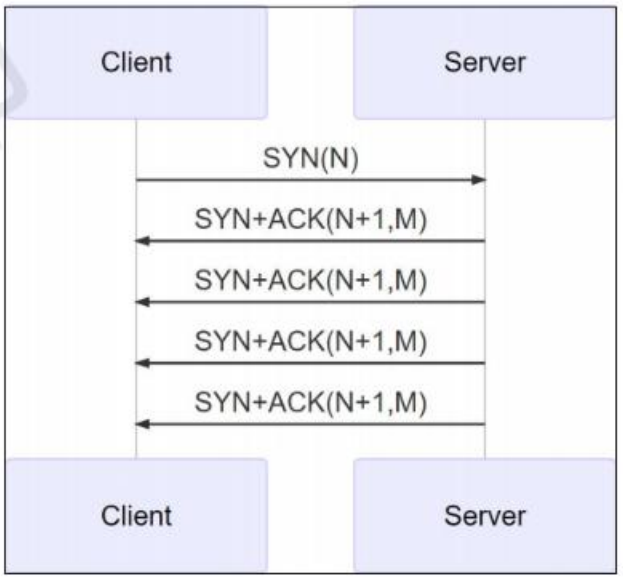
- TCP连接的三路握手过程，如下图所示：
  - 客户端向该端口发送一个TCP数据包，其包头中带有SYN标志位，发送序列号为N
  - 服务器收到该数据包后，向客户端返回一个TCP数据包，其包头中带有SYN和ACK两个标志位，接收序列号为N+1，发送序列号为M
  - 客户端收到该数据包后，再次向服务器发送一个TCP数据包，其包头中带有ACK标志位，接收序列号为M+1



# DDoS攻击

知识讲解

- 如果服务器在向客户端发出SYN+ACK(N+1,M)数据包后，未能在给定时间内收到对方的应答，就会认为所发送数据包已丢失，进而重发该数据包。如果重发多次，始终未能收到客户端的应答，服务器才会最终放弃尝试。如下图所示：
  - 在这个过程中，服务器需要记录客户端数据包信息，维护重发定时器，判断是否超时等等，由此造成的资源消耗远大于正常发送数据的消耗



# DDoS攻击

- 如果攻击者不停地向服务器发送大量的孤立SYN数据包，服务器资源将很快消耗殆尽，无法继续提供正常的网络服务
- 这就是TCP SYN泛洪拒绝服务攻击的基本原理


知识讲解

# DDoS攻击

- 与早期由单台主机发起的单兵作战式的拒绝服务攻击不同，近年来渐渐流行起来的分布式拒绝服务(Distributed Denial of Service, DDoS)攻击借助多台被植入攻击木马的傀儡主机，同时向一个目标主机发起集团作战式的拒绝服务攻击，致使被攻击主机遭受巨大的压力，即使是高带宽、高配置的网络服务器也难以幸免
- 目前大部分分布式拒绝服务攻击都是通过僵尸网络(Botnet)实现的。攻击者先将攻击程序部署在僵尸网络的各个被控主机上，在选定攻击目标后，通过僵尸网络向所有被控主机上的攻击程序发送指令，使其同时向攻击目标发起进攻，剧烈消耗目标主机的网络带宽和运算资源，以致其瞬间瘫痪

# DDoS攻击

- 抵御TCP SYN攻击的方式有很多，例如：正确设置防火墙、充足的带宽保证、服务器硬件的升级等。
- 在众多的方法中，内核加固是一种比较不错的选择
- 通过对内核函数添加必要代码，在数据包进入到上层协议栈之前对其进行拦截，若确系TCP SYN攻击包，则通知协议栈予以丢弃，以此达到系统内核加固的目的。

知识讲解

# 下节课见