本文面试题汇总：
防范常见的 Web 攻击
重要协议分布层
arp 协议的工作原理
rip 协议是什么？rip 的工作原理
什么是 RARP？工作原理
OSPF 协议？OSPF 的工作原理
TCP 与 UDP 区别总结
什么是三次握手四次挥手？
tcp 为什么要三次握手？
dns 是什么？dns 的工作原理
一次完整的 HTTP 请求过程
Cookies 和 session 区别
GET 和 POST 的区别
HTTPS 和 HTTP 的区别
session 的工作原理？
http 长连接和短连接的区别
OSI 的七层模型都有哪些？
session 的工作原理？什么是 TCP 粘包/拆包？发生原因？解决方案
TCP 如何保证可靠传输？
URI 和 URL 的区别
什么是 SSL ？
https 是如何保证数据传输的安全（SSL 是怎么工作保证安全的）
TCP 对应的应用层协议，UDP 对应的应用层协议
常见的状态码有哪些？

防范常见的 Web 攻击
什么是 SQL 注入攻击
攻击者在 HTTP 请求中注入恶意的 SQL 代码，服务器使用参数构建数据库 SQL 命令时，恶意 SQL 被一起构造，并在数据库中执行。
用户登录，输入用户名 lianggzone，密码 ‘ or ‘1’=’1 ，如果此时使用参数构造的方式，就会出现
select * from user where name = ‘lianggzone’ and password = ‘’ or ‘1’=‘1’
不管用户名和密码是什么内容，使查询出来的用户列表不为空。如何防范 SQL 注入攻击使用预编译的 PrepareStatement 是必须的，但是一般我们会从两个方面同时入手。
Web 端
1）有效性检验。
2）限制字符串输入的长度。
服务端
1）不用拼接 SQL 字符串。
2）使用预编译的 PrepareStatement。
3）有效性检验。(为什么服务端还要做有效性检验？第一准则，外部都是不可信的，防止攻击者绕过 Web 端请求)
4）过滤 SQL 需要的参数中的特殊字符。比如单引号、双引号。

什么是 XSS 攻击

跨站点脚本攻击，指攻击者通过篡改网页，嵌入恶意脚本程序，在用户浏览网页时，控制用户浏览器进行恶意操作的一种攻击方式。如何防范 XSS 攻击

1）前端，服务端，同时需要字符串输入的长度限制。

2）前端，服务端，同时需要对 HTML 转义处理。将其中的”<”,”>”等特殊字符进行转义编码。

防 XSS 的核心是必须对输入的数据做过滤处理。

什么是 CSRF 攻击

跨站点请求伪造，指攻击者通过跨站请求，以合法的用户的身份进行非法操作。可以这么理解 CSRF 攻击：攻击者盗用你的身份，以你的名义向第三方网站发送恶意请求。CRSF 能做的事情包括利用你的身份发邮件，发短信，进行交易转账，甚至盗取账号信息。如何防范 CSRF 攻击

安全框架，例如 Spring Security。

token 机制。在 HTTP 请求中进行 token 验证，如果请求中没有 token 或者 token 内容不正确，则认为 CSRF 攻击而拒绝该请求。

验证码。通常情况下，验证码能够很好的遏制 CSRF 攻击，但是很多情况下，出于用户体验考虑，验证码只能作为一种辅助手段，而不是最主要的解决方案。

referer 识别。在 HTTP Header 中有一个字段 Referer，它记录了 HTTP 请求的来源地址。如果 Referer 是其他网站，就有可能是 CSRF 攻击，则拒绝该请求。但是，服务器并非都能取到 Referer。很多用户出于隐私保护的考虑，限制了 Referer 的发送。在某些情况下，浏览器也不会发送 Referer，例如 HTTPS 跳转到 HTTP。

1）验证请求来源地址；

2）关键操作添加验证码；

3）在请求地址添加 token 并验证。

什么是文件上传漏洞

文件上传漏洞，指的是用户上传一个可执行的脚本文件，并通过此脚本文件获得了执行服务端命令的能力。

许多第三方框架、服务，都曾经被爆出文件上传漏洞，比如很早之前的 Struts2，以及富文本编辑器等等，可被攻击者上传恶意代码，有可能服务端就被人黑了。如何防范文件上传漏洞

文件上传的目录设置为不可执行。

1）判断文件类型。在判断文件类型的时候，可以结合使用 MIME Type，后缀检查等方式。因为对于上传文件，不能简单地通过后缀名称来判断文件的类型，因为攻击者可以将可执行文件的后缀名称改为图片或其他后缀类型，诱导用户执行。

2）对上传的文件类型进行白名单校验，只允许上传可靠类型。

3）上传的文件需要进行重新命名，使攻击者无法猜想上传文件的访问路径，将极大地增加攻击成本，同时向 shell.php.rar.ara 这种文件，因为重命名而无法成功实施攻击。

4）限制上传文件的大小。

5）单独设置文件服务器的域名。

DDos 攻击

客户端向服务端发送请求链接数据包，服务端向客户端发送确认数据包，客户端不向服务端发送确认数据包，服务器一直等待来自客户端的确认
没有彻底根治的办法，除非不使用 TCP
DDos 预防：
1）限制同时打开 SYN 半链接的数目
2）缩短 SYN 半链接的 Time out 时间
3）关闭不必要的服务

重要协议分布图

arp 协议的工作原理
地址解析协议，即 ARP（Address Resolution Protocol），是根据 IP 地址获取物理地址的一个 TCP/IP 协议。
1.发送 ARP 请求的以太网数据帧 广播 到以太网上的每个主机，ARP 请求帧中包含了目的主机的 IP 地址。
2.目的主机收到了该 ARP 请求之后，会发送一个 ARP 应答，里面包含了目的主机的 MAC 地址。
ARP 协议工作原理：
每个主机都会在自己的 ARP 缓冲区中建立一个 ARP 列表，以表示 IP 地址和 MAC 地址之间的对应关系。
主机（网络接口）新加入网络时（也可能只是 mac 地址发生变化，接口重启等）， 会发送免费 ARP 报文把自己 IP 地址与 Mac 地址的映射关系广播给其他主机。
网络上的主机接收到免费 ARP 报文时，会更新自己的 ARP 缓冲区。将新的映射关系更新到自己的 ARP 表中。
某个主机需要发送报文时，首先检查 ARP 列表中是否有对应 IP 地址的目的主机的 MAC 地址，如果有，则直接发送数据，如果没有，就向本网段的所有主机发送 ARP 数据包，该数据包包括的内容有：源主机 IP 地址，源主机 MAC 地址，目的主机的 IP 地址等。
当本网络的所有主机收到该 ARP 数据包时：
（A）首先检查数据包中的 IP 地址是否是自己的 IP 地址，如果不是，则忽略该数据包。
（B）如果是，则首先从数据包中取出源主机的 IP 和 MAC 地址写入到 ARP 列表中，如果已经存在，则覆盖。
（C） 然后将自己的 MAC 地址写入 ARP 响应包中，告诉源主机自己是它想要找的 MAC 地址。
6.源主机收到 ARP 响应包后。将目的主机的 IP 和 MAC 地址写入 ARP 列表，并利用此信息发送数据。如果源主机一直没有收到 ARP 响应数据包，表示 ARP 查询失败。ARP 高速缓存（即 ARP 表）是 ARP 地址解析协议能够高效运行的关键

什么是 RARP？工作原理
概括： 反向地址转换协议，网络层协议，RARP 与 ARP 工作方式相反。 RARP 使只知道自己硬件地址的主机能够知道其 IP 地址。RARP 发出要反向解释的物理地址并希望返回其 IP 地址，应答包括能够提供所需信息的 RARP 服务器发出的 IP 地址。
原理：
(1)网络上的每台设备都会有一个独一无二的硬件地址，通常是由设备厂商分配的 MAC 地址。

主机从网卡上读取 MAC 地址，然后在网络上发送一个 RARP 请求的广播数据包，请求 RARP 服务器回复该主机的 IP 地址。

(2)RARP 服务器收到了 RARP 请求数据包，为其分配 IP 地址，并将 RARP 回应发送给主机。

(3)PC1 收到 RARP 回应后，就使用得到的 IP 地址进行通讯。

dns 是什么？dns 的工作原理
将主机域名转换为 ip 地址，属于应用层协议，使用 UDP 传输。（DNS 应用层协议，以前有个考官问过）

过程：
总结： 浏览器缓存，系统缓存，路由器缓存，IPS 服务器缓存，根域名服务器缓存，顶级域名服务器缓存，主域名服务器缓存。
一、主机向本地域名服务器的查询一般都是采用递归查询。
二、本地域名服务器向根域名服务器的查询的迭代查询。
1)当用户输入域名时，浏览器先检查自己的缓存中是否 这个域名映射的 ip 地址，有解析结束。
2）若没命中，则检查操作系统缓存（如 Windows 的 hosts）中有没有解析过的结果，有解析结束。
3）若无命中，则请求本地域名服务器解析（ LDNS）。
4）若 LDNS 没有命中就直接跳到根域名服务器请求解析。根域名服务器返回给 LDNS 一个 主域名服务器地址。
5） 此时 LDNS 再发送请求给上一步返回的 gTLD（ 通用顶级域）， 接受请求的 gTLD 查找并返回这个域名对应的 Name Server 的地址
6） Name Server 根据映射关系表找到目标 ip，返回给 LDNS
7） LDNS 缓存这个域名和对应的 ip， 把解析的结果返回给用户，用户根据 TTL 值缓存到本地系统缓存中，域名解析过程至此结束

rip 协议是什么？rip 的工作原理
RIP 动态路由选择协议（网络层协议）
RIP 是一种基于距离矢量（Distance-Vector）算法的协议，它使用跳数（Hop Count）作为度量来衡量到达目的网络的路由距离。RIP 通过 UDP 报文进行路由信息的交换，使用的端口号为 520。

工作原理：
RIP 路由协议用“更新（UNPDATES）”和“请求（REQUESTS）”这两种分组来传输信息的。每个具有 RIP 协议功能的路由器每隔 30 秒用 UDP520 端口给与之直接相连的机器广播更新信息。并且在（ 用“路程段数”（即“跳数”）作为网络距离的尺度。每个路由器在）给相邻路由器发出路由信息时，都会给每个路径加上内部距离。
路由器的收敛机制：
任何距离向量路由选择协议（如 RIP）都有一个问题，路由器不知道网络的全局情况，路由器必须依靠相邻路由器来获取网络的可达信息。由于路由选择更新信息在网络上传播慢，距

离向量路由选择算法有一个慢收敛问题，这个问题将导致不一致性产生。
RIP 较少路由收敛机制带来的问题：
1）记数到无穷大机制： RIP 协议允许最大跳数为 15。大于 15 的目的地被认为是不可达。 当路径的跳数超过 15，这条路径才从路由表中删除。
2） 水平分割法： 路由器不向路径到来的方向回传此路径。当打开路由器接口后，路由器记录路径是从哪个接口来的，并且不向此接口回传此路径。
3） 破坏逆转的水平分割法： 忽略在更新过程中从一个路由器获取的路径又传回该路由器
4）保持定时器法： 防止路由器在路径从路由表中删除后一定的时间内（通常为 180 秒）接受新的路由信息。 保证每个路由器都收到了路径不可达信息
5） 触发更新法： 当某个路径的跳数改变了，路由器立即发出更新信息，不管路由器是否到达常规信息更新时间都发出更新信息。

RIP 的缺点
1、由于 15 跳为最大值，RIP 只能应用于小规模网络；2、收敛速度慢；3、根据跳数选择的路由，不一定是最优路由。

OSPF 协议？OSPF 的工作原理
OSPF（Open Shortest Pass First,开放最短路径优先协议），是一个最常用的内部网管协议，是一个链路状态协议。(网络层协议,)
原理：
OSPF 组播的方式在所有开启 OSPF 的接口发送 Hello 包，用来确定是否有 OSPF 邻居，若发现了，则建立 OSPF 邻居关系，形成邻居表，之后互相发送 LSA（链路状态通告）相互通告路由，形成 LSDB（链路状态数据库）。再通过 SPF 算法，计算最佳路径（cost 最小）后放入路由表。

TCP 与 UDP 区别总结？
1.TCP 面向连接（如打电话要先拨号建立连接）提供可靠的服务;UDP 是无连接的，即发送数据之前不需要建立连接，;UDP 尽最大努力交付，即不保证可靠交付。（由于 UDP 无需建立连接，因此 UDP 不会引入建立连接的时延，TCP 需要在端系统中维护连接状态，比如接受和发送缓存，拥塞控制，序号与确认号的参数等，故 TCP 会比 UDP 慢）
2.UDP 具有较好的实时性，工作效率比 TCP 高，适用于对高速传输和实时性有较高的通信或广播通信。
3. 每一条 TCP 连接只能是一对一的;UDP 支持一对一，一对多，多对一和多对多的交互通信
4 UDP 分组首部开销小，TCP 首部开销 20 字节;UDP 的首部开销小，只有 8 个字节。
5. TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流;UDP 是面向报文的（一次交付一个完整的报文，报文不可分割，报文是 UDP 数据报处理的最小单位）。
6.UDP 适合一次性传输较小数据的网络应用，如 DNS，SNMP 等

什么是三次握手四次挥手？tcp 为什么要三次握手？
为了防止已失效的连接请求报文段突然又传送到了服务端，因而产生错误

第一次握手：建立连接时，客户端发送 syn 包(syn=j)到服务器，并进入 SYN_SEND 状态，等待服务器确认；
第二次握手：服务器收到 syn 包，必须确认客户的 SYN（ack=j+1），同时自己也发送一个 SYN

包（syn=k），即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态；
第三次握手：客户端收到服务器的 SYN＋ACK 包，向服务器发送确认包 ACK(ack=k+1)，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。
完成三次握手，客户端与服务器开始传送数据

客户端先发送 FIN，进入 FIN_WAIT1 状态，用来关闭 Client 到 Server 的数据传送
服务端收到 FIN，发送 ACK，进入 CLOSE_WAIT 状态，客户端收到这个 ACK，进入 FIN_WAIT2 状态
服务端发送 FIN，进入 LAST_ACK 状态，用来关闭 Server 到 Client 的数据传送
客户端收到 FIN，发送 ACK，进入 TIME_WAIT 状态，服务端收到 ACK，进入 CLOSE 状态（等待 2MSL 时间，约 4 分钟。主要是防止最后一个 ACK 丢失。）
第一次挥手：主动关闭方发送一个 FIN，用来关闭主动方到被动关闭方的数据传送，也就是主动关闭方告诉被动关闭方：我已经不 会再给你发数据了(当然，在 fin 包之前发送出去的数据，如果没有收到对应的 ack 确认报文，主动关闭方依然会重发这些数据)，但是，此时主动关闭方还可 以接受数据。
第二次挥手：被动关闭方收到 FIN 包后，发送一个 ACK 给对方，确认序号为收到序号+1（与 SYN 相同，一个 FIN 占用一个序号）。
第三次挥手：被动关闭方发送一个 FIN，用来关闭被动关闭方到主动关闭方的数据传送，也就是告诉主动关闭方，我的数据也发送完了，不会再给你发数据了。
第四次挥手：主动关闭方收到 FIN 后，发送一个 ACK 给被动关闭方，确认序号为收到序号+1，至此，完成四次挥手。

GET 和 POST 的区别
get 是获取数据，post 是修改数据
get 把请求的数据放在 url 上， 以?分割 URL 和传输数据，参数之间以&相连，所以 get 不太安全。而 post 把数据放在 HTTP 的包体内（requrest body）
get 提交的数据最大是 2k（ 限制实际上取决于浏览器）， post 理论上没有限制。
GET 产生一个 TCP 数据包，浏览器会把 http header 和 data 一并发送出去，服务器响应 200(返回数据); POST 产生两个 TCP 数据包，浏览器先发送 header，服务器响应 100 continue，浏览器再发送 data，服务器响应 200 ok(返回数据)。
GET 请求会被浏览器主动缓存，而 POST 不会，除非手动设置。
GET 是幂等的，而 POST 不是幂等的
Cookies 和 session 区别
Cookie 和 Session 都是客户端与服务器之间保持状态的解决方案
1，存储的位置不同，cookie：存放在客户端，session：存放在服务端。Session 存储的数据比较安全
2，存储的数据类型不同
两者都是 key-value 的结构，但针对 value 的类型是有差异的
cookie：value 只能是字符串类型，session：value 是 Object 类型
3，存储的数据大小限制不同
cookie：大小受浏览器的限制，很多是是 4K 的大小， session：理论上受当前内存的限制，

4，生命周期的控制
cookie 的生命周期当浏览器关闭的时候，就消亡了
(1)cookie 的生命周期是累计的，从创建时，就开始计时，20 分钟后，cookie 生命周期结束，
(2)session 的生命周期是间隔的，从创建时，开始计时如在 20 分钟，没有访问 session，那么 session 生命周期被销毁

session 的工作原理？
session 的工作原理是客户端登录完成之后，服务器会创建对应的 session，session 创建完之后，会把 session 的 id 发送给客户端，客户端再存储到浏览器中。这样客户端每次访问服务器时，都会带着 sessionid，服务器拿到 sessionid 之后，在内存找到与之对应的 session 这样就可以正常工作了。

一次完整的 HTTP 请求过程
域名解析 --> 发起 TCP 的 3 次握手 --> 建立 TCP 连接后发起 http 请求 --> 服务器响应 http 请求，浏览器得到 html 代码 --> 浏览器解析 html 代码，并请求 html 代码中的资源（如 js、css、图片等） --> 浏览器对页面进行渲染呈现给用户。

HTTPS 和 HTTP 的区别
1.HTTP 协议传输的数据都是未加密的，也就是明文的，因此使用 HTTP 协议传输隐私信息非常不安全， HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全。
2. https 协议需要到 ca 申请证书，一般免费证书较少，因而需要一定费用。
3、http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。
https://www.cnblogs.com/wqhwe/p/5407468.html

OSI 的七层模型都有哪些？
物理层：利用传输介质为数据链路层提供物理连接，实现比特流的透明传输。
数据链路层：接收来自物理层的位流形式的数据，并封装成帧，传送到上一层
网络层：将网络地址翻译成对应的物理地址，并通过路由选择算法为分组通过通信子网选择最适当的路径。
传输层：在源端与目的端之间提供可靠的透明数据传输
会话层：负责在网络中的两节点之间建立、维持和终止通信
表示层：处理用户信息的表示问题，数据的编码，压缩和解压缩，数据的加密和解密
应用层：为用户的应用进程提供网络通信服务

http 长连接和短连接的区别
在 HTTP/1.0 中默认使用短连接。也就是说，客户端和服务器每进行一次 HTTP 操作，就建立一次连接，任务结束就中断连接。而从 HTTP/1.1 起，默认使用长连接，用以保持连接特性。什么是 TCP 粘包/拆包？发生原因？解决方案 一个完整的业务可能会被 TCP 拆分成多个包进行发送，也有可能把多个小的包封装成一个大的数据包发送，这个就是 TCP 的拆包和粘包问题。原因：1. 应用程序写入数据的字节大小大于套接字发送缓冲区的大小.2. 进行 MSS 大小的 TCP 分段。(MSS=TCP 报文段长度-TCP 首部长度)3. 以太网的 payload 大于 MTU 进行 IP 分片。（ MTU 指：一种通信协议的某一层上面所能通过的最大数据包大小。）解决方案：1. 消息定长。2. 在包尾部增加回车或者空格符等特殊字符进行分割 3. 将消息分为消息头和消息

尾。4. 使用其它复杂的协议，如 RTMP 协议等。

TCP 如何保证可靠传输？
1. 三次握手。
2. 将数据截断为合理的长度。应用数据被分割成 TCP 认为最适合发送的数据块（按字节编号，合理分片）
3. 超时重发。当 TCP 发出一个段后，它启动一个定时器，如果不能及时收到一个确认就重发
4. 确认应答：对于收到的请求，给出确认响应
5. 校验和：校验出包有错，丢弃报文段，不给出响应
6. 序列号：对失序数据进行重新排序，然后才交给应用层
7. 丢弃重复数据：对于重复数据 ， 能够丢弃重复数据
8. 流量控制。TCP 连接的每一方都有固定大小的缓冲空间。TCP 的接收端只允许另一端发送接收端缓冲区所能接纳的数据。这将防止较快主机致使较慢主机的缓冲区溢出。
9. 拥塞控制。当网络拥塞时，减少数据的发送。
1
2
3
4
5
6
7
8
9
校验和
序列号
确认应答
超时重传
连接管理
流量控制
拥塞控制

常见的状态码有哪些？
200 OK //客户端请求成功 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，eg：输入了错误的 URL
500 Internal Server Error //服务器发生不可预期的错误 URI 和 URL 的区别 URI，统一资源标识符，用来唯一的标识一个资源。URL 可以用来标识一个资源，而且还指明了如何定位这个资源。

什么是 SSL ？https 是如何保证数据传输的安全（SSL 是怎么工作保证安全的）
SSL 代表安全套接字层。它是一种用于加密和验证应用程序（如浏览器）和 Web 服务器之间发送的数据的协议。 身份验证 ， 加密 Https 的加密机制是一种共享密钥加密和公开密钥加密并用的混合加密机制。SSL/TLS 协议作用：认证用户和服务，加密数据，维护数据的完整性的应用层协议加密和解密需要两个不同的密钥，故被称为非对称加密；加密和解密都使

用同一个密钥的 对称加密。 优点在于加密、解密效率通常比较高 HTTPS 是基于非对称加密的， 公钥是公开的，
（1）客户端向服务器端发起 SSL 连接请求；
（2） 服务器把公钥发送给客户端，并且服务器端保存着唯一的私钥
（3）客户端用公钥对双方通信的对称秘钥进行加密，并发送给服务器端
（4）服务器利用自己唯一的私钥对客户端发来的对称秘钥进行解密，
（5）进行数据传输，服务器和客户端双方用公有的相同的对称秘钥对数据进行加密解密，可以保证在数据收发过程中的安全，即是第三方获得数据包，也无法对其进行加密，解密和篡改。
因为数字签名、摘要是证书防伪非常关键的武器。 “摘要”就是对传输的内容，通过 hash 算法计算出一段固定长度的串。然后，在通过 CA 的私钥对这段摘要进行加密，加密后得到的结果就是“数字签名”

SSL/TLS 协议的基本思路是采用公钥加密法，也就是说，客户端先向服务器端索要公钥，然后用公钥加密信息，服务器收到密文后，用自己的私钥解密。
如何保证公钥不被篡改？
将公钥放在数字证书中。只要证书是可信的，公钥就是可信的。
公钥加密计算量太大，如何减少耗用的时间？
每一次对话（session），客户端和服务器端都生成一个"对话密钥"（session key），用它来加密信息。由于"对话密钥"是对称加密，所以运算速度非常快，而服务器公钥只用于加密"对话密钥"本身，这样就减少了加密运算的消耗时间。
（1） 客户端向服务器端索要并验证公钥。
（2） 双方协商生成"对话密钥"。
（3） 双方采用"对话密钥"进行加密通信。上面过程的前两步，又称为"握手阶段"（handshake）。

《计算机网络》书本：
SSL 工作过程，A：客户端，B：服务器端
1.协商加密算法：A 向 B 发送 SSL 版本号和可选加密算法，B 选择自己支持的算法并告知 A
2.服务器鉴别：B 向 A 发送包含公钥的数字证书，A 使用 CA 公开发布的公钥对证书进行验证
3.会话密钥计算：A 产生一个随机秘密数，用 B 的公钥进行加密后发送给 B，B 根据协商的算法产生共享的对称会话密钥并发送给 A.
4.安全数据传输：双方用会话密钥加密和解密它们之间传送的数据并验证其完整性

TCP 对应的应用层协议
FTP：定义了文件传输协议，使用 21 端口.
Telnet：它是一种用于远程登陆的端口,23 端口
SMTP：定义了简单邮件传送协议，服务器开放的是 25 号端口。
POP3：它是和 SMTP 对应，POP3 用于接收邮件。
HTTP
UDP 对应的应用层协议
DNS：用于域名解析服务，用的是 53 号端口
SNMP：简单网络管理协议，使用 161 号端口

TFTP(Trival File Transfer Protocal)：简单文件传输协议，69