料整理自网络，仅作免费交流分享，侵权删！**

更多网安面试资料请关注微信公众号：马哥教育

1、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。

3、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。

3、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。

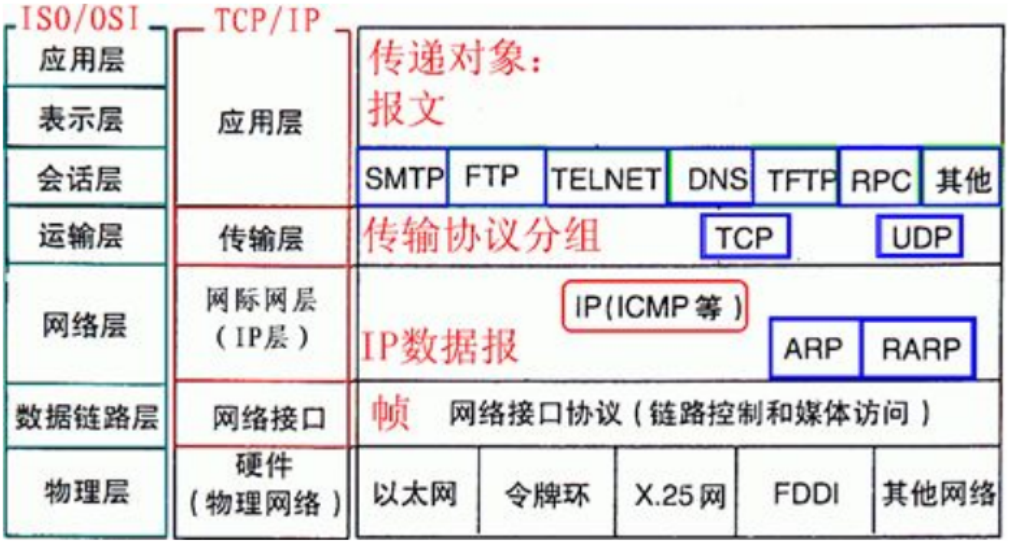
4、什么是文件上传漏洞

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。

5、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

6、重要协议分布图



7、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

8、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。

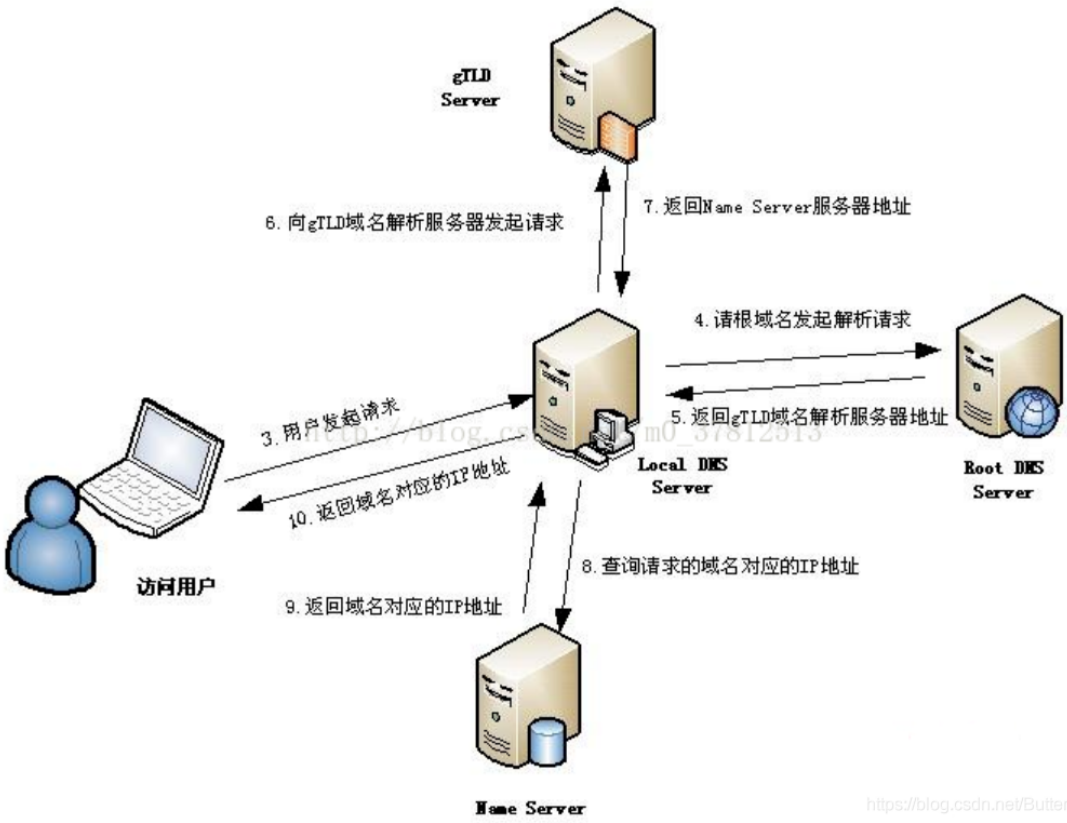
(2)RARP服务器收到了RARP请求数据包，为其分配IP地址，并将RARP回应发送给主机。

(3)PC1收到RARP回应后，就使用得到的IP地址进行通讯。

9、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



10、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。

11、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。

12、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。

13、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。

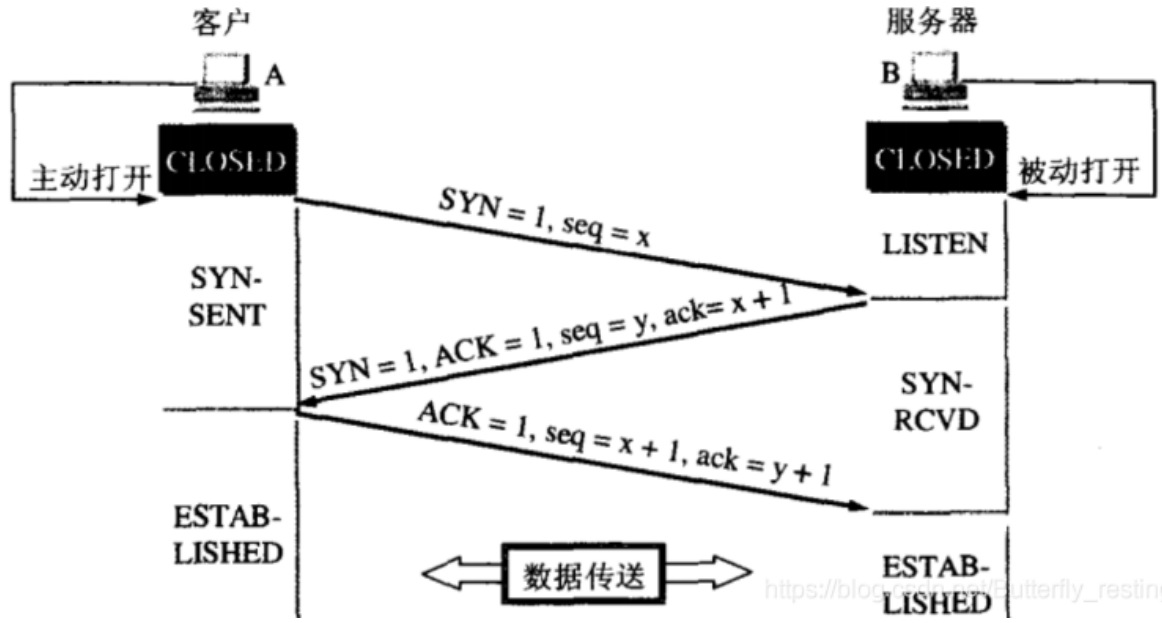
3. 每一条TCP连接只能是一对一的;UDP支持一对一，一对多，多对一和多对多的交互通信
   4 UDP分组首部开销小，TCP首部开销20字节;UDP的首部开销小，只有8个字节。
4. TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流;UDP是面向报文的（一次交付一个完整的报文，报文不可分割，报文是UDP数据报处理的最小单位）。

6.UDP适合一次性传输较小数据的网络应用，如DNS，SNMP等

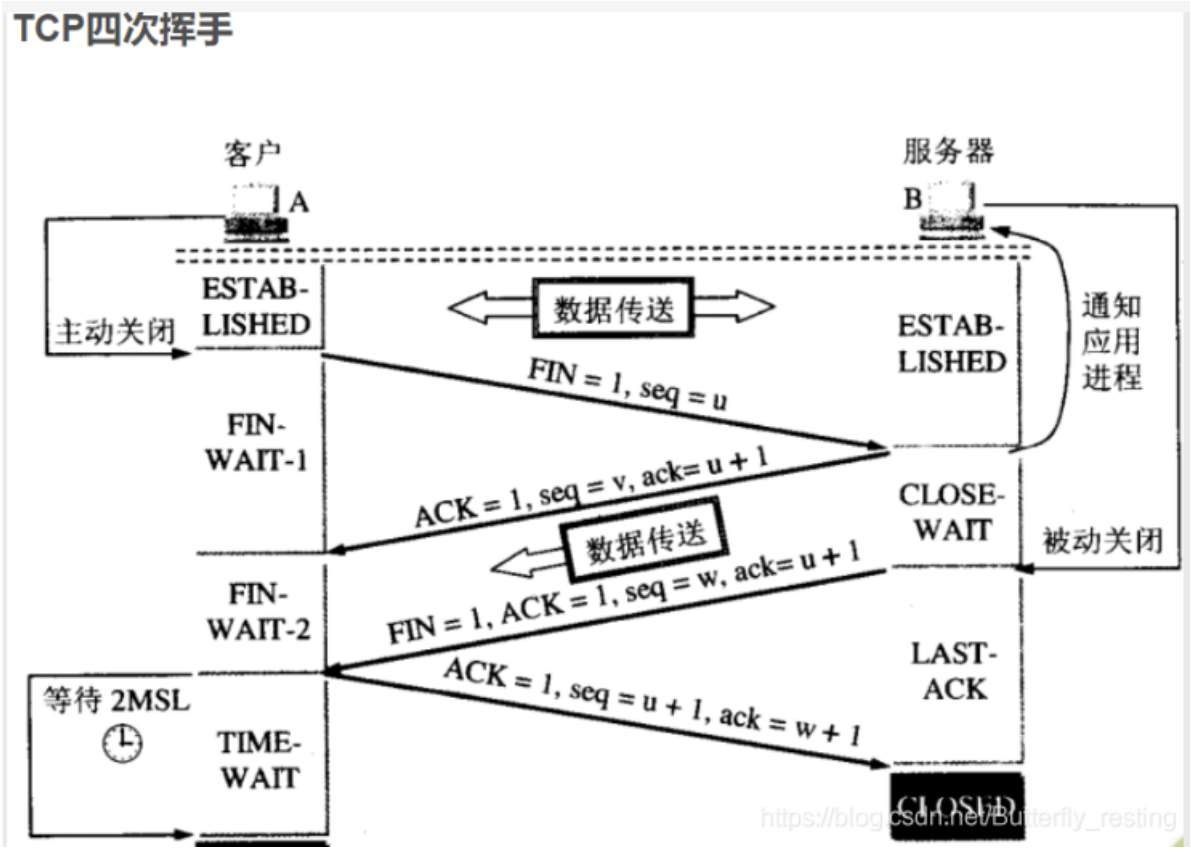
14、什么是三次握手四次挥手？tcp为什么要三次握手?

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。



15、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

16、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

17、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。

18、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。

19、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

20、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

21、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。

22、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。

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

23、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。

24、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。

25、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）。
————————————————————

1-25题来自 https://blog.csdn.net/Butterfly_resting/article/details/89638424

26、php爆绝对路径方法?

单引号引起数据库报错
访问错误参数或错误路径
探针类文件如phpinfo
扫描开发未删除的测试文件
google hacking
phpmyadmin报路径：/phpmyadmin/libraries/lect_lang.lib.php
利用漏洞读取配置文件找路径
恶意使用网站功能，如本地图片读取功能读取不存在图片，上传点上传不能正常导入的文件

27、你常用的渗透工具有哪些，最常用的是哪个?

burp、nmap、sqlmap、awvs、蚁剑、冰蝎、dirsearch、御剑等等

28、xss盲打到内网服务器的利用

钓鱼管理员
信息收集

29、鱼叉式攻击和水坑攻击?

鱼叉攻击：指利用木马程序作为电子邮件的附件，发送到目标电脑上，诱导受害者去打开附件来感染木马
水坑攻击：分析攻击目标的上网活动规律，寻找攻击目标经常访问的网站的弱点，将网站攻破并植入恶意程序，等待目标访问

30、什么是虚拟机逃逸?

利用虚拟机软件或者虚拟机中运行的软件的漏洞进行攻击，以达到攻击或控制虚拟机宿主操作系统的目的

31、中间人攻击?

原理：

在同一个局域网中，通过拦截正常的网络通信数据，并进行数据篡改和嗅探

防御：

在主机绑定网关MAC与IP地址为静态
在网关绑定主机MAC与IP地址
使用ARP防火墙

32、TCP三次握手过程?

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

33、七层模型?

应用层、表示层、会话层、传输层、网络层、数据链路层、物理层

34、对于云安全的理解

融合了并行处理、网格计算、未知病毒行为判断等新兴技术和概念，通过网状的大量客户端对网络中软件行为的异常监测，获取互联网中木马、恶意程序的最新信息，传送到Server端进行自动分析和处理，再把病毒和木马的解决方案分发到每一个客户端

35、了解过websocket吗?

WebSocket是一种在单个TCP连接上进行全双工通信的协议，最大特点是服务器可以主动向客户端推送信息，客户端也可以主动向服务器发送信息，是真正的双向平等对话。

36、DDOS是什么？有哪些？CC攻击是什么？区别是什么?

DDOS：

分布式拒绝服务攻击，利用合理的服务请求来占用过多的服务资源，从而使合法用户无法得到服务的响应
主要方式：
SYN Flood
UDP Flood
ICMP Flood
Connection Flood
HTTP Get
UDP DNS Query Flood

CC攻击：

模拟多个正常用户不停地访问如论坛这些需要大量数据操作的页面，造成服务器资源的浪费，CPU长时间处于100%，网络拥塞

两者区别：

CC攻击网页，DDOS攻击服务器，更难防御
CC门槛较低，DDOS需要大量服务器
CC持续时间长，DDOS产生的影响大

37、land攻击是什么?

局域网拒绝服务攻击，DDOS攻击的一种，通过发送精心构造的、具有相同源地址和目标地址的欺骗数据包，致使缺乏相应防护机制的目标设备瘫痪

38、你会如何进行信息收集?

服务器信息：ip、中间件、操作系统
域名whois、ipwhois、网段归属
子域名探测
网站目录扫描、接口信息扫描
端口扫描
各大引擎搜索相关信息

39、什么是CRLF注入攻击?

通过“回车”和“换行”字符注入HTTP流，实现网站篡改、跨站脚本、劫持等。

40、防止XSS，前端后端两个角度?

前端：

用户输入特殊字符过滤转义为html实体
用户输出编码

后端：

实体化编码
函数过滤
限制字符长度

41、如何防护一个端口的安全?

利用WAF、IDS、IPS等设备
危险服务端口禁止对外访问或限制IP访问
服务定期更新版本

42、webshell检测思路?

静态检测：匹配特征码，特征值，危险函数
动态检测：WAF、IDS等设备
日志检测：通过IP访问规律，页面访问规律筛选
文件完整性监控

43、GPC是什么？开启了怎么绕过

GPC：

php.ini配置文件中的magic_quotes_gpc，实现为get、post、cookie传入的单引号、双引号、反斜线、NULL字符添加反斜线\

绕过：

PHP5的GPC对$_SERVER的忽略，可在http请求头注入
二次注入
宽字节注入

44、web常用的加密算法有什么

单向散列加密 MD5、SHA、MAC
对称加密 AES、DES
非对称加密 RSA、RSA2

45、XSS除了获取cookies还能做什么?

获取管理员ip
xss蠕虫
钓鱼攻击
前端JS挖矿
键盘记录
屏幕截图

46、运营商（或其他）网络劫持

运营商劫持：广告投放
DNS劫持：通过各种手段篡改DNS，劫持网络

47、DNS欺骗是什么

攻击者冒充域名服务器的一种欺骗行为

48、缓冲区溢出原理和防御

原理：

当写入缓冲区的数据量超过该缓冲区所能承受的最大限度时，发生缓冲区溢出，溢出的数据被黑客加以利用，形成远程代码执行漏洞。

防御：

基于操作系统防御
缓冲区边界检查
安全编程

49、网络安全事件应急响应

断网：条件允许时优先断网，防止黑客进一步操作或删除痕迹
取证：通过分析登录日志、网站日志、服务日志寻找黑客ip，查看黑客进行的操作
备份：备份服务器文件，对比入侵前后产生变化的文件
查漏：通过上述步骤寻找业务薄弱点，修补漏洞
杀毒：清除黑客留下的后门、webshell、管理账号
溯源：通过黑客ip地址，入侵手段等
记录：归档、预防

50、企业内部安全

实名制联网
重要网段隔离
禁止接入任何USB设备
禁用WIFI网络
IP与MAC地址绑定
部署网络监控、IDS、IPS设备
定期培训，提高员工安全意识

51、业务上线前，怎么测试，从哪些角度测试

安全测试：寻找产品漏洞，页面漏洞，服务漏洞，敏感信息泄露，逻辑漏洞，弱口令
性能测试：压力测试
功能完整性测试

52、应用有漏洞，但是无法修复和停用，你怎么办

限制IP白名单访问
使用WAF、IDS、防火墙设备

53、CSRF怎么防护?

验证HTTP Referer字段
添加Token字段并验证
添加自定义字段并验证

54、文件上传绕过方法?

WAF绕过:

修改上传表单字段
表单字段大小写替换
表单字段增加或减少空格
表单字段字符串拼接
构造双文件上传表单，同时上传双文件
编码绕过
垃圾数据填充绕过
文件名大小写绕过

服务器检测绕过:

MIME类型绕过
前端JS检测抓包改包绕过
黑名单绕过：php3、asa、ashx、windows特性（test.asp_、流特性）、apache解析漏洞
图片内容检测使用图片马绕过
.htassess绕过

白名单检测绕过:

截断上传绕过
IIS6/7/7.5解析漏洞，nginx低版本解析漏洞
文件包含绕过

55、验证码相关利用点

验证码复用
验证码可识别
验证码失效
验证码DDOS

56、cookie你会测试什么内容

sql注入
xss
权限绕过
敏感信息泄露

57、说出几个业务逻辑漏洞类型?

任意用户密码重置
短信轰炸
订单金额修改
忘记密码绕过

恶意刷票
验证码复用

58、简述文件包含漏洞

调用文件包含函数时，未严格限制文件名和路径，如include()、require()等函数

59、业务逻辑漏洞，用户任意密码重置有什么例子，因为什么因素导致的?

普通用户重置管理用户密码
普通用户重置普通用户密码

未设置用户唯一Token，导致越权

60、渗透测试过程中发现一个只能上传zip文件的功能，有什么可能的思路?

shell压缩上传，程序自解压getshell
尝试解析漏洞getshell
寻找文件包含漏洞
木马钓鱼管理员

61、为什么aspx木马权限比asp大?

aspx使用的是.net技术,IIS中默认不支持，ASPX需要依赖于.net framework，ASP只是脚本语言
入侵的时候asp的木马一般是guest权限APSX的木马一般是users权限

62、只有一个登录页面有哪些思路?

SQL注入、万能密码
暴力破解
权限绕过
目录扫描
敏感信息泄露

63、请求头中哪些是有危害的?

COOKIE注入
user-agent注入
X-Forwarded-For注入
Referer注入

64、谈谈水平/垂直/未授权越权访问的区别?

水平越权：普通用户越权访问普通用户
垂直越权：普通用户越权访问管理用户
未授权访问：权限控制不严，导致无需登录访问已登录用户页面

65、xss有什么？执行存储型的xss的危害和原理

存储型、反射型、DOM型
存储型XSS是指应用程序通过Web请求获取不可信赖的数据，在未检验数据是否存在XSS代码的情况下，便将其存入数据库

存储型XSS危害：

窃取用户Cookie
XSS钓鱼攻击
XSS蠕虫攻击
获取键盘记录
获取用户信息
获取屏幕截图

66、主机疑似遭到入侵，要看哪里的日志

系统登录日志
服务访问日志
网站日志
数据库日志

67、python常用的标准库

正则表达式 re
时间模块 time
随机数 random
操作系统接口 os
科学计算 math
网络请求 urlib
http库 requests
爬虫库 Scrapy
多线程库 threading

68、reverse_tcp 和 bind_tcp 的区别?

reverse_tcp：攻击机设置一个端口和IP，Payload在测试机执行连接攻击机IP的端口，这时如果在攻击机监听该端口会发现测试机已经连接
白话就是让受控机主动连接我们
bind_tcp：攻击机设置一个端口（LPORT），Payload在测试机执行打开该端口，以便攻击机可以接入
白话就是我们主动连接受控机
使用reverse_tcp较为安全，一般不会被防火墙发现

69、oauth认证过程中可能会出现什么问题，导致什么样的漏洞?

CSRF
redirect_uri校验不严格
错误的参数传递

70、做了cdn的网站如何获取真实IP

全球ping
查询历史解析记录
探针文件如phpinfo等
利用命令执行连接我们的服务器或DNSlog
寻找网站配置
通过二级域名
全网扫描，title匹配

71、如何实现跨域?

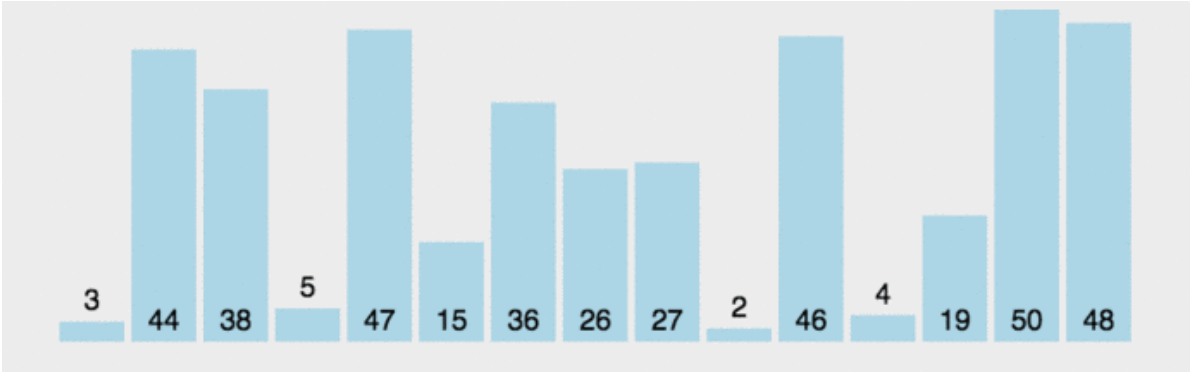
jsonp
CORS跨域资源共享
代理跨域请求
Html5 postMessage 方法
修改 document.domain 跨子域
基于 Html5 websocket 协议
document.xxx + iframe

72、jsonp跨域与CORS跨域的区别?

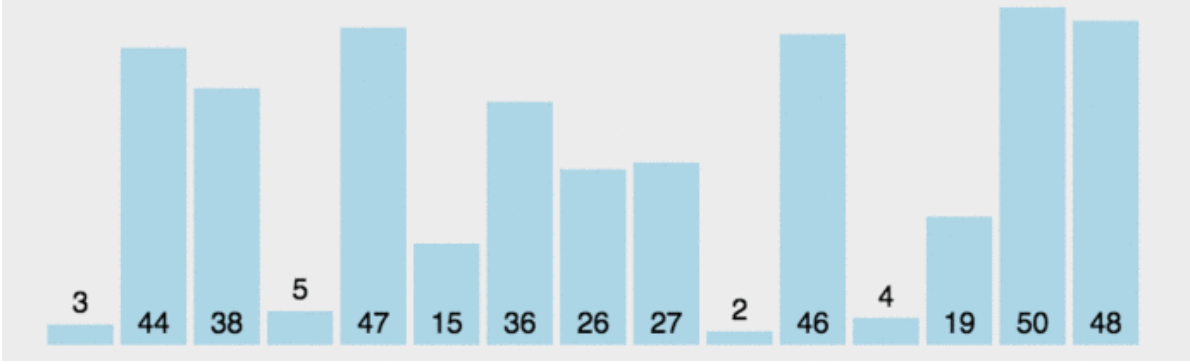
jsonp浏览器支持较好，CORS不支持IE9及以下浏览器
jsonp只支持GET，CORS支持所有类型的HTTP请求
jsonp只发一次请求，复杂请求CORS发送两次

73、算法？了解过什么排序?

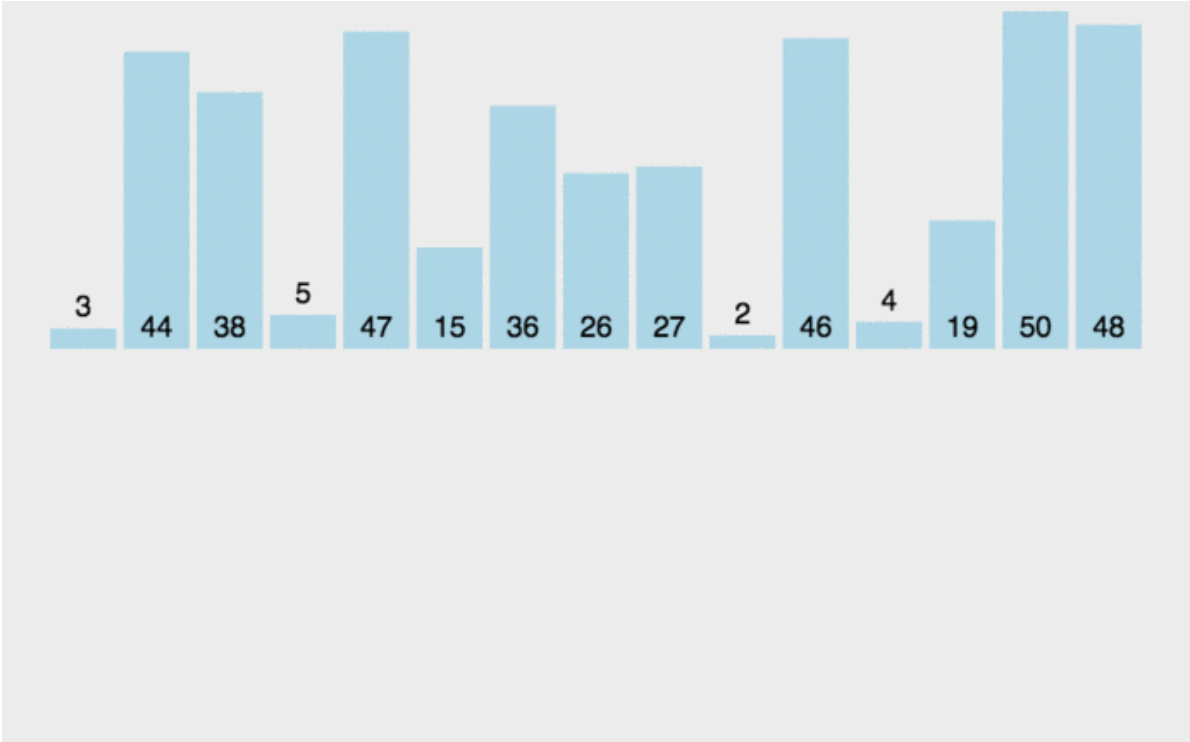
冒泡排序



选择排序



插入排序



74、SSRF漏洞利用?

本地文件读取
服务探测、端口扫描
攻击内网redis、mysql、fastcgi等服务

利用到的协议有：http/s、file、gopher、tftp、dict、ssh、telnet

75、常见后门方式?

Windows:

注册表自启动
shift后门
远控软件
webshell
添加管理用户
影子用户
定时任务
dll劫持
注册表劫持
MBR后门
WMI后门
管理员密码记录
Linux:
SSH后门
SUID后门
Crontab计划任务
PAM后门
添加管理员账号
Rootkit

76、open_basedir访问目录限制绕过方法?

使用命令执行函数绕过
使用symlink()函数绕过
glob伪协议绕过

77、PHP代码审计中容易出问题的点?

参数拼接方式皆有可能产生SQL注入（老生常谈）
全局变量注册导致的变量覆盖
fwrite参数未过滤导致的代码执行
权限校验疏漏导致的后台功能访问
接口任意文件上传
unserialize反序列化漏洞

78、红蓝对抗中蓝队反杀红队场景和姿势?

钓鱼、蜜罐、蚁剑RCE

79、linux计划任务，黑客隐藏自己的计划任务会怎么做?

临时任务：at、batch命令

80、Redis未授权常见getshell的几种方式?

web绝对路径写shell
写入ssh公钥获取服务器权限
主从复制getshell

81、JWT的攻击手法？（头部、负载、签名）

加密算法置为空绕过身份验证
爆破弱密钥
kid参数：任意文件读取、SQL注入、命令注入
未校验签名，内容重新编码

82、JAVA中间件的漏洞，举几个例子?

JBoss反序列化
WebLogic反序列化
tomcat任意文件写入、弱口令+后台getshell

83、DNS外带可以用在哪些漏洞?

SQL盲注
无回显的命令执行
XXE盲打
SSRF盲打

HTTP-Only禁止的是JS读取cookie信息，如何绕过这个获取cookie
劫持登录页面钓鱼绕过

84、中间件漏洞总结?

这里只写常利用的漏洞

IIS:

IIS6.0 PUT漏洞
IIS6.0 远程代码执行漏洞
IIS6.0 解析漏洞
IIS启用.net 短文件名漏洞
IIS7.0/7.5 解析漏洞

Apache:

未知扩展名解析漏洞
配合错误导致的解析漏洞、目录遍历
Nginx：
配置错误导致的解析漏洞、目录遍历

Tamcat:

配置错误导致的任意代码执行、任意文件写入漏洞
弱口令+管理后台war包部署getshell
manager/html 管理后台弱口令爆破

JBoss:

5.x/6.x反序列化漏洞（CVE-2017-12149）
JMXInvokerServlet反序列化
EJBInvokerServlet反序列化
JMX Console未授权访问
弱口令+管理后台war包部署getshell

WebLogic:

XMLDecoder 反序列化漏洞（CVE-2017-10271 & CVE-2017-3506）
wls9_async_response,wls-wsat 反序列化远程代码执行漏洞（CVE-2019-2725）
WLS Core Components 反序列化命令执行漏洞（CVE-2018-2628）
弱口令+管理后台war包部署getshell

85、谈一谈Windows系统与Linux系统提权的思路?

Windows:

数据库提权：mysql、sqlserver
第三方软件提权：serv-u
DLL劫持
系统内核溢出漏洞提权：cve系列

Linux：

sudo提权
suid提权
redis
内核提权

86、ython有哪些框架，其中出现过哪些漏洞

Django、Flask、Scrapy
Django任意代码执行
Flask模板注入

87、小程序的渗透和普通渗透的差异

渗透过程不变，依旧是抓包修改参数渗透
不同点是小程序会将包下载到本地，可以使用逆向还原工具反编译

88、app本身的漏洞测试 四大组件

Activity组件:

activity绑定browserable与自定义协议
ActivityManager漏洞

Service组件:

权限提升，拒绝服务攻击

Broadcast Receiver组件:

权限管理不当
BroadcastReceiver导出漏洞
动态注册广播组件暴露漏洞

Content Provider组件:

读写权限漏洞
Content Provider中的SQL注入漏洞
Provider文件目录遍历漏洞

89、IDS/IPS防护原理及绕过思路

原理：

IDS工作在网络层，旁路部署，通过抓取和分析网络流量来发现攻击
IPS一般也是在网络层旁路，可以理解为具备阻断能力的IDS，是IDS的升级版（也有IDS检测到攻击通知阻断设备执行阻断动作的设备联动模式），可以覆盖网络层和应用层

绕过：

TCP分片：拆分出两个TCP包
IP分片：原理同TCP分片，但是丢包严重
程序bug/性能问题：发送大量无效包，消耗IPS性能
伪造TCP状态：绕过基于状态追踪的IPS
IPV6绕过：使用IPV6地址绕过

90、json的csrf的利用

使用XMLHttpRequest、fetch构造出JSON请求，利用Flash的跨域与307跳转来绕过http自定义头限制

91、json格式的数据包可以测哪些漏洞

csrf
json劫持
xss

92、内网服务器，如何进行信息收集?

使用脚本收集：端口信息、服务信息
系统命令收集：域内用户可使用域命令收集域信息，net group "domain users" /domain等
端口扫描工具全段扫描
本机信息收集：管理密码、登录日志看管理员ip、服务密码收集、网段信息查看、历史记录查看
内网DNS域传送漏洞

93、如果拿下了内网边界层的某一个机器，如何对内网其他进行探测?

首先使用代理进入内网reg、ew等
第二在本机进行信息收集，包括管理员ip、端口服务、账号密码、路由信息、网段信息等
第三扩展到收集到的网段进行渗透，利用常用服务:SMB、MYSQL、SQLserver、ftp、telnet等
借助轻量化脚本或扫描器扫描，但一般不这么做，动静太大容易被管理员发现

26-93题整理自网络，侵权删!