在了解分布式拒绝服务攻击的原理之前，先要了解以下两个关键的基础原理

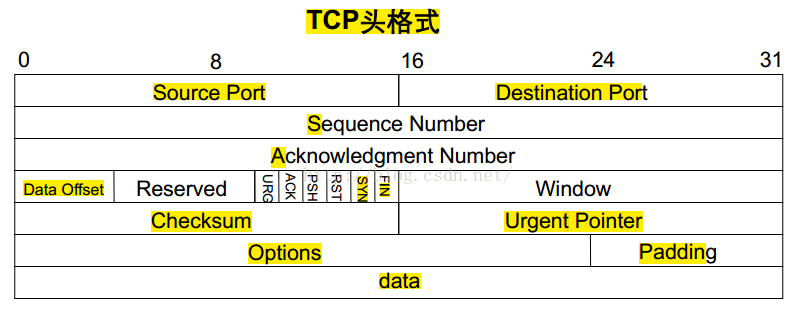
1.TCP饿死：

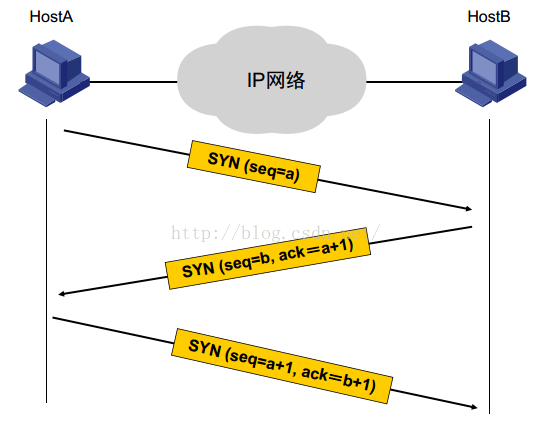
UDP这种传输方式不会控制自己在通信通道里的流量，可理解为不讲道理的人。他们来到了一个热闹地区的KFC中，但是他们不买东西只排队将所有食物的价格都问一遍，占满所有的座位和过道。而常规的TCP服务通过自己的弹窗机制来控制流量，好比讲道理的人，座位被占满了，TCP自然会离开KFC导致正常的服务不能进行。最终的结果就是UDP将整个通道打满堵死。

2.TCP三次握手和四次断开连接：

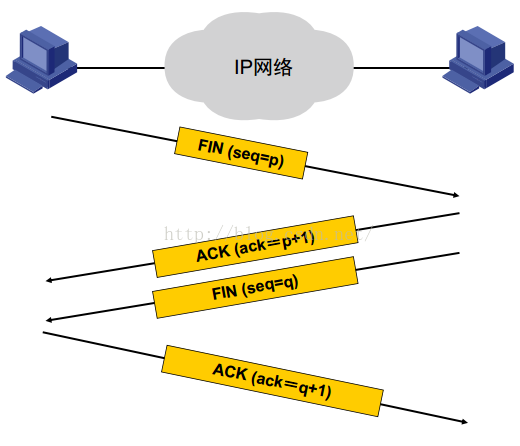
文字说不清楚，见下图

TCP头格式



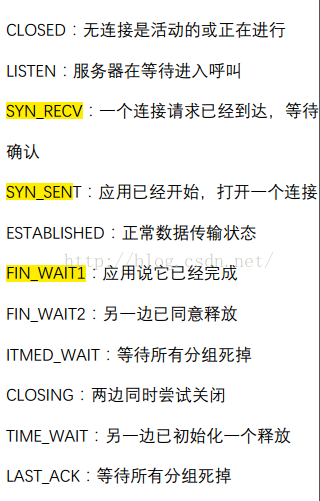


seq:序列号；ack:确认序列号



双方均需要确认才可断开连接

关于双方在整个过程中每个时间段的状态名称·



分析完以上的两个原理便可继续

DOS攻击：一台或多台计算机对受攻击服务器的某一个端口发送大量无关的UDP报文，导致整个通道内的正常服务无法进行。

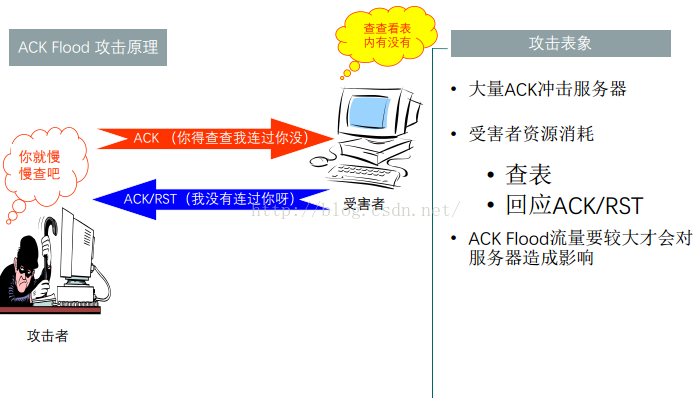
DDOS攻击：大量的肉鸡对服务器的不同端口发送巨型流量的UDP报文，无法通关关闭端口的方式来进行隔离，破坏力极强，严重会造成服务器当机。

根据攻击的时间和方式又可分将DDOS为以下几种

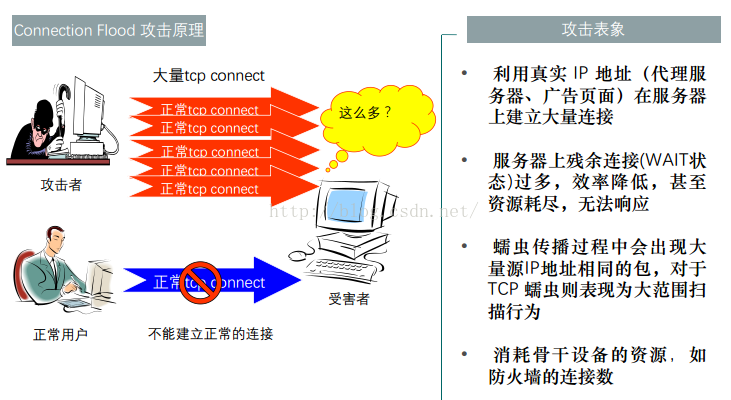
1.SYN Flood



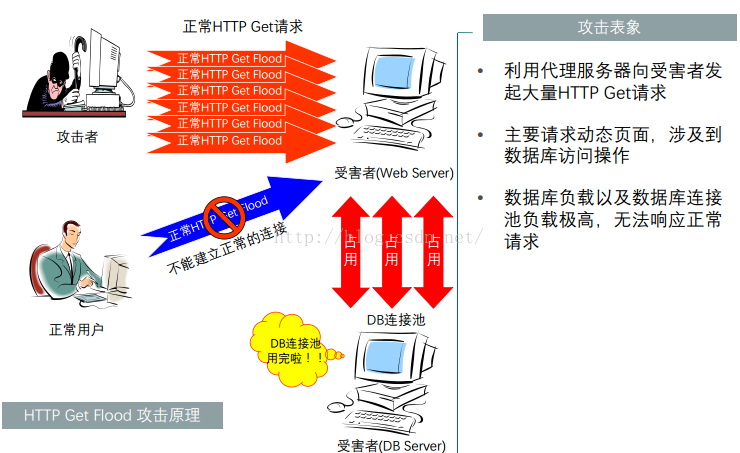
2.ACK Flood



3.Connection Flood



4.HTTP Get Flood



对TCP饿死的应对方案

1.增加带宽，堵死了再买

2.CDN，各地部署子服务器，当子服务遭受到攻击时，其他地区的服务器和主服务器不会受到影响。

3.BGP流量清洗，通过BGP将通道内的无用UDP报文清洗干净再转给服务器

**1、SYN洪水攻击**  
 要理解dos攻击，首先要理解TCP连接的三次握手过程(Three-wayhandshake)。  
 在TCP/IP协议中，TCP协议提供可靠的连接服务，采用三次握手建立一个连接。   
  第一次握手:建立连接时，客户端发送SYN包((SYN=i)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=i+1 )，同时自己也发送一个SYN包((SYN=j)}即SYN+ACK包，此时服务器进入SYN\_RECV状态;[1]   
  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=j+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，客户端与服务器开始传送数据。   
  在上述过程中，还有一些重要的概念:[1]   
  半连接:  **收到SYN包而还未收到ACK包时的连接状态称为半连接，即尚未完全完成三次握手的TCP连接。**  半连接队列:    在三次握手协议中，服务器维护一个半连接队列，该队列为每个客户端的SYN包(SYN=i )开设一个条目，该条目表明服务器已收到SYN包，并向客户发出确认，正在等待客户的确认包。这些条目所标识的连接在服务器处于SYN\_ RECV状态，当服务器收到客户的确认包时，删除该条目，服务器进入ESTABLISHED状态。  
 Backlog参数:    表示半连接队列的最大容纳数目。   
 SYN-ACK重传次数:    服务器发送完SYN-ACK包，如果未收到客户确认包，服务器进行首次重传，等待一段时间仍未收到客户确认包，进行第二次重传，如果重传次数超过系统规定的最大重传次数，系统将该连接信息、从半连接队列中删除。注意，每次重传等待的时间不一定相同。   
**半连接存活时间:    是指半连接队列的条目存活的最长时间，也即服务从收到SYN包到确认这个报文无效的最长时间，该时间值是所有重传请求包的最长等待时间总和。有时也称半连接存活时间为Timeout时间、SYN\_RECV存活时间。   
上面三个参数对系统的TCP连接状况有很大影响。**  
 SYN洪水攻击属于DoS攻击的一种，它利用TCP协议缺陷，**通过发送大量的半连接请求，耗费CPU和内存资源。**SYN攻击除了能影响主机外，还可以危害路由器、防火墙等网络系统，事实上SYN攻击并不管目标是什么系统，只要这些系统打开TCP服务就可以实施。从图4-3可看到，服务器接收到连接请求(SYN=i )将此信息加入未连接队列，并发送请求包给客户端( SYN=j,ACK=i+1 )，此时进入SYN\_RECV状态。当服务器未收到客户端的确认包时，重发请求包，一直到超时，才将此条目从未连接队列删除。配合IP欺骗，SYN攻击能达到很好的效果，通常，客户端在短时间内伪造大量不存在的IP地址，向服务器不断地发送SYN包，服务器回复确认包，并等待客户的确认，由于源地址是不存在的，服务器需要不断的重发直至超时，这些伪造的SYN包将长时间占用未连接队列，正常的SYN 请求被丢弃，目标系统运行缓慢，严重者引起网络堵塞甚至系统瘫痪。过程如下:[1]   
 **攻击主机C(地址伪装后为C&apos;)-----大量SYN包---->被攻击主机**    
  **C&apos;<-------SYN/ACK包----被攻击主机**  
 由于**C’地址不可达，被攻击主机等待SYN包超时。攻击主机通过发大量SYN包填满未连接队列，导致正常SYN包被拒绝服务。另外，SYN洪水攻击还可以通过发大量ACK包进行DoS攻击。**  
 【防御方法】      
 第一种是**缩短SYN Timeout时间**  
 第二种方法是**设置SYN Cookie**，就是给每一个请求连接的IP地址分配一个Cookie，如果短时间内连续受到某个IP的重复SYN报文，就认定是受到了攻击，以后从这个IP地址来的包会被一概丢弃。  
   >netstat -n -p tcp >result.txt  
 【SYN Cookie】  
 当服务器收到一个SYN报文后，**不立即分配缓冲区**，而是利用连接的信息生成一个cookie，并将这个cookie作为将要返回的SYN＋ACK报文的初始序列号。  
 当客户端返回一个ACK报文时，根据包头信息计算cookie，与返回的确认序列号（初始的序列号＋1）的前24位进行对比，如果相同，则是一个正常连接，然后，分配资源，建立连接。                  
 【缺陷】  
 SYN Cookies 的使用不与任何协议定义冲突，照理来说它该和所有的 TCP 实现兼容。然而，当 SYN Cookies 使用的时候，会发生两种值得注意的变化：  
 首先，服务器只能编码八种 MSS 数值，因为只有 3 位二进制空间可用。  
 其次，这个服务器必须拒绝所有的TCP 选用项，例如大型窗口和时间戳，因为服务器会在信息被用其他方式存储时丢弃 SYN 队列条目。

 尽管这些限制将不可避免地导致一个不如最佳的体验，它们的效果很少被客户端注意到——这些改变只在被攻击时值得注意。在这样的情况下，牺牲 TCP 选项来保护连接一般被认为是合乎情理的。   
 Linux内核从 2.6.26 版本开始为 TCP 选用项加入了有限的支持，通过把它们编码在时间戳内实现。   
 较新的TCP Cookie 传输（TCPCT）标准被设计用来克服 SYN Cookies 的这些问题，并且在各种方面改进这套机制。不像 SYN Cookies，TCPCT 是一个 TCP 拓展并且要求两端点都支持 TCPCT。    
      
**2、ACK Flood攻击**

 在TCP连接建立之后，**所有的数据传输TCP报文都是带有ACK标志位的，**主机在接收到一个带有ACK标志位的数据包的时候，需要检查该数据包所表示的连接四元组是否存在，如果存在则检查该数据包所表示的状态是否合法，然后再向应用层传递该数据包。如果在检查中发现该数据包不合法，例如该数据包所指向的目的端口在本机并未开放，则**主机操作系统协议栈会回应RST包告诉对方此端口不存在。**通常状态检测防火墙所做的事情与此类似，只不过防火墙只拦截非法的数据包，而不主动回应。  
  
 对比主机以及防火墙在接收到ACK报文和SYN报文时所做动作的复杂程度，显然ACK报文带来的负载要小得多。所以在实际环境中，只有当攻击程序每秒钟发送ACK报文的速率达到一定的程度，才能使主机和防火墙的负载有大的变化。当发包速率很大的时候，**主机操作系统将耗费大量的精力接收报文、判断状态，同时要主动回应RST报文，正常的数据包就可能无法得到及时的处理**。这时候客户端（以IE为例）的表现就是访问页面反应很慢，丢包率较高。**但是状态检测的防火墙通过判断ACK报文的状态是否合法，借助其强大的硬件能力可以较为有效的过滤攻击报文。**当然如果攻击流量非常大（特别是千兆线路上，我们曾经观察到200~300Mbps左右的ACK Flood），由于需要维护很大的连接状态表同时要检查数量巨大的ACK报文的状态，防火墙也会不堪重负导致全网瘫痪。  
  
3、死亡之ping (pingofdeath)

 ICMP(InternetControlMessageProtocol，Internet控制信息协议)**在Internet上用于错误处理和传递控制信息。**最普通的ping程序就是这个功能。而在TCP/IP的RFC文档中对包的最大尺寸都有严格限制规定，许多操作系统的TCP/IP协议栈**都规定ICMP包大小为64KB，且在对包的标题头进行读取之后，要根据该标题头里包含的信息来为有效载荷生成缓冲区。"PingofDeath"就是故意产生畸形的测试Ping（PacketInternetGroper）包，声称自己的尺寸超过ICMP上限，也就是加载的尺寸超过64KB上限，使未采取保护措施的网络系统出现内存分配错误，导致TCP/IP协议栈崩溃，最终接收方宕机。** 【防御方法】  
**1、用高级设置法预防Ping  
 2、用网络防火墙阻隔Ping**  
   使用防火墙来阻隔Ping是最简单有效的方法，现在基本上所有的防火墙在默认情况下都启用了ICMP过滤的功能。  
 3、**启用IP安全策略防Ping**  
 4、**修改TTL值防Ping**   许多入侵者喜欢用TTL值来判断操作系统，他们首先会Ping一下你的机子，如看到TTL值为128就认为你的系统为Windows NT/2000，如果TTL值为32则认为目标主机操作系统为Windows 95/98，  
   如果为TTL值为255/64就认为是UNIX/Linux操作系统。既然入侵者相信TTL值所反应出来的结果，那么我们不妨修改TTL值来欺骗入侵者，达到保护系统的目的  
 5、防火墙在处理Ping of Death攻击报文时，是通过判定数据包的大小是否大于65535字节，如果数据包大于65535字节，则判定为攻击报文，直接丢弃。  
  
**4、UDP泛洪**

 UDPflood攻击：如今在**Internet上UDP（用户数据包协议）的应用比较广泛，**很多提供WWW和Mail等服务设备通常是使用Unix的服务器，它们**默认打开一些被黑客恶意利用的UDP服务。如echo服务会显示接收到的每一个数据包，而原本作为测试功能的chargen服务会在收到每一个数据包时随机反馈一些字符。**UDPflood假冒攻击就是利用**这两个简单的TCP/IP服务的漏洞**进行恶意攻击，通过伪造与某一主机的Chargen服务之间的一次的UDP连接，回复地址指向开着Echo服务的一台主机，通过将Chargen  
 和Echo服务互指，**来回传送毫无用处且占满带宽的垃圾数据，在两台主机之间生成足够多的无用数据流，**这一拒绝服务攻击飞快地导致网络可用带宽耗尽。   
 【防御方法】  
  UDP协议与TCP 协议不同，是无连接状态的协议，并且UDP应用协议五花八门，差异极大，因此针对UDP Flood的防护非常困难。其防护要根据具体情况对待：

 判断包大小，如果是大包攻击则使用防止UDP碎片方法：根据攻击包大小设定包碎片重组大小，通常不小于1500.在极端情况下，可以考虑丢弃所有UDP碎片。  
 攻击端口为业务端口：根据该业务UDP最大包长设置UDP最大包大小以过滤异常流量。  
 攻击端口为非业务端口：一个是丢弃所有UDP包，可能会误伤正常业务;一个是建立UDP连接规则，要求所有去往该端口的UDP包，必须首先与TCP端口建立TCP连接。不过这种方法需要很专业的防火墙或其他防护设备支持。  
          
5、Land（LandAttack）攻击

**在Land攻击中，黑客利用一个特别打造的SYN包--它的原地址和目标地址都被设置成某一个服务器地址进行攻击**。**此举将导致接受服务器向它自己的地址发送SYN-ACK消息， 结果这个地址又发回ACK消息并创建一个空连接，每一个这样的连接都将保留直到超时，**在Land攻击下，许多UNIX将崩溃，NT变得极其缓慢（大约持续五分钟）。   
 【防御方法】  
 防火墙在处理**Land攻击报文时，通过检查TCP报文的源地址和目的地址是否相同，或者TCP报文的源地址是否为环回地址，如果是则丢弃。**

6、泪滴攻击     
 对于一些大的IP数据包，往往需要对其进行拆分传送，这是为了迎合链路层的MTU（最大传输单元）的要求。比如，一个6 000字节的IP包，在MTU为2 000的链路上传输的时候，就需要分成3个IP 包。在IP报头中有一个偏移字段和一个拆分标志（MF）。如果MF标志设置为1，则表示这个IP包是一个大IP包的片段，其中偏移字段指出了这个片段在整个IP包中的位置。例如，对一个6 000字 节的IP包进行拆分（MTU为2 000），**则3个片段中偏移字段的值依次为0，2 000，4 000。**这样接收端在全部接收完IP数据包后，就可以根据这些信息重新组装这几个分次接收的拆分IP包。在这 里就有一个安全漏洞可以利用了，就是**如果黑客们在截取IP数据包后，把偏移字段设置成不正确的值，这样接收端在收到这些分拆的数据包后，就不能按数据包中的偏移字段值正确组合这些拆分的数据包，但接收端会不断尝试，这样就可能致使目标计算机操作系统因资源耗尽而崩溃。**

 【防御方法】

 检测这类攻击的方法是对接收到的分片数据包进行分析，计算数据包的片偏移量（Offset）是否有误。反攻击的方法是添加系统补丁程序，丢弃收到的病态分片数据包并对这种攻击进行审计。尽可能采用最新的操作系统，或者在防火墙上设置分段重组功能，由防火墙先接收到同一原包中的所有拆分数据包，然后完成重组工作，而不是直接转发。因为防火墙上可以设置当出现重叠字段时所采用的规则。

**7、IP地址扫描攻击  
 IP地址扫描攻击是攻击者运用ICMP报文（如Ping和Tracert命令）探测目标地址**，或者使用TCP/UDP报文对一定地址发起连接，通过判断是否有应答报文，以确定哪些目标系统确实存活着并且连接在目标网络上。

 【防御方法】

  防火墙对收到的TCP、UDP、ICMP报文进行检测，当某源IP地址连续发送报文的目的IP地址与前一个报文的目的IP地址不同时，则记为一次异常，当异常次数超过预定义的阈值时，则认为该源IP的行为为IP地址扫描行为，防火墙会将该源IP地址加入黑名单。  
        

**8、Smurf攻击**

**Smurf攻击通过使用将回复地址设置成受害网络的广播地址的ICMP应答请求(ping)数据包，来淹没受害主机，最终导致该网络的所有主机都对此ICMP应答请求做出答复，导致网络阻塞。更加复杂的Smurf将源地址改为第三方的受害者，最终导致第三方崩溃。**

 攻击的过程是这样的：Woodlly Attacker向一个**具有大量主机和因特网连接的网络的广播地址发送一个欺骗性Ping分组（echo 请求）**，这个目标网络被称为反弹站点，而欺骗性Ping分组的源地址就是Woolly希望攻击的系统。

 这种攻击的前提是，路由器接收到这个发送给IP广播地址（如206.121.73.255）的分组后，会认为这就是广播分组，并且把以太网广播地址FF:FF:FF:FF:FF:FF:映射过来。这样路由器因因特网上接收到该分组，会对本地网段中的所有主机进行广播。

 读者肯定能够想到下面会发生什么情况。网段中的所有主机都会向欺骗性分组的IP地址发送echo响应信息。如果这是一个很大的以太网段，可以会有500个以上的主机对收到的echo请求进行回复。

 由于多数系统都会尽快地处理ICMP传输信息，Woodlly Attacker把分组的源地址设置为目标系统，因此目标系统都很快就会被大量的echo信息吞没，这样轻而易举地就能够阻止该系统处理其它任何网络传输，从而引起拒绝为正常系统服务。

 这种攻击不仅影响目标系统，还影响目标公司的因特网连接。如果反弹站点具有T3连接（45Mbps），而目标系统所在的公司使用的是租用线路（56Kbps），则所有进出该公司的通讯都会停止下来。

这种攻击现在已经很少见,大多数的网络已经对这种攻击免疫了.

【防御方法】

 配置路由器禁止IP广播包进网  
 配置网络上所有计算机的操作系统，禁止对目标地址为广播地址的ICMP包响应。  
 被攻击目标与ISP协商，有ISP暂时阻止这些流量。

 对于从本网络向外部网络发送的数据包，本网络应该将其源地址为其他网络的这部分数据包过

**9、Fraggle攻击**

**类似于Smurf，使用UDP应答消息而非ICMP**。UDP端口7（ECHO）和端口19（Chargen）在收到UDP报文后，都会产生回应。在UDP的7号端口收到报文后，会回应收到的内容，而UDP的19号端口在收到报文后，会产生一串字符流。它们都同ICMP一样，会产生大量无用的应答报文，占满网路带宽。攻击者可以向子网广播地址发送源地址为受害网络或受害主机的UDP包，端口号用7或19.子网络启用了此功能的每个系统都会向受害者的主机做出响应，从而引发大量的包，导致受害网络的阻塞或受害主机的崩溃；子网上没有启动这些功能的系统将产生一个ICMP不可达的消息，因而仍然消耗带宽。也可将源端口改为Chargen。目的端口为ECHO，这样会自动不停地产生回应报文，其危害性更大。   
【防御方法】   
 检查进入防火墙的UDP报文，若目的端口号为7或19，则直接拒绝，并将攻击记录到日志，否则允许通过。

**10、电子邮件炸弹**  
 电子邮件炸弹是最古老的匿名攻击之一，通过设置一台机器不断的大量的向同一地址发送电子邮件，攻击者能够耗尽接受者网络的带宽。   
【防御方法】

对邮件地址进行配置，自动删除来自同一主机的过量或重复的消息。   
  
11、畸形消息攻击   
 各类操作系统上的许多服务都存在此类问题，由于这些服务在处理信息之前没有进行适当正确的错误校验，在收到畸形的信息可能会崩溃。

【防御方法】

 打最新的服务补丁。