1. 阐述黑客进行网络攻击的一般流程

答：

1. 信息的收集  
   　　 信息的收集并不对目标产生危害，只是为进一步的入侵提供有用信息。黑客可能会利用下列的公开协议或工具，收集驻留在网络系统中的各个主机系统的相关信息。  
   2、系统安全弱点的探测  
   　　 在收集到一些准备要攻击目标的信息后，黑客们会探测目标网络上的每台主机，来寻求系统内部的安全漏洞。  
   3、建立模拟环境，进行模拟攻击  
   　　 根据前面两小点所得的信息，建立一个类似攻击对象的模拟环境，然后对此模拟目标进行一系列的攻击。在此期间，通过检查被攻击方的日志，观察检测工具对攻击的反应，可以进一步了解在攻击过程中留下的“痕迹”及被攻击方的状态，以此来制定一个较为周密的攻击策略。  
   4、具体实施网络攻击  
   　　 入侵者根据前几步所获得的信息，同时结合自身的水平及经验总结出相应的攻击方法，在进行模拟攻击的实践后，将等待时机，以备实施真正的网络攻击。

（2）什么是端口扫描？有哪些扫描技术？

答：

端口扫描：端口扫描是指某些别有用心的人发送一组端口扫描消息，试图以此侵入某台计算机，并了解其提供的计算机网络服务类型（这些网络服务均与[端口号](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3%E5%8F%B7/10883658" \t "https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3%E6%89%AB%E6%8F%8F/_blank)相关）。

常用的扫描技术：

****1、**TCP connect()扫描：**

1. **TCP SYN扫描：**
2. **TCP FIN 扫描：**
3. **IP段扫描：**
4. **TCP 反向 ide**nt扫**描：**
5. **FTP 返回攻击：**
6. **UDP ICMP端口不能到达扫描：**
7. **UDP recvfrom()和write() 扫描：**

（3）网络踩点与网络扫描的差别？

答：差别：

网络踩点Footprinting 主动或被动的获取信息的情报工作  
网络扫描技术Scanning 检测目标系统是否同互联网连接，以及开放的网络服务类型。

（4）Windows系统有哪些查点方法？

答：常见的查点方法

****1、****“查点NetBIOS”线路

****2、****空会话（null session）

****3、****SNMP 代理和活动目录（active directory）

（5）简述Unicode攻击方法

答：Unicode攻击方法：Unicode标准被很多软件开发者所采用，无论何种平台、程序或开发语言，Unicode均为每个[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "https://baike.baidu.com/item/Unicode%E6%BC%8F%E6%B4%9E/_blank)提供独一无二的序号，如向IIS服务器发出包括非法Unicode

UTF-8序列的URL，攻击者可使服务器逐字“进入或退出”目录并执行任意程序，该攻击即称为目录转换攻击。Unicode用“%2f”和“%5c”分别代表“/”和“\”字符，但也可用“超长”序列来代替这些字符。“超长”序列是非法的Unicode表示符，如用“%c0%af”代表“/”字符。由于IIS不对超长序列进行检查，因此在URL中添加超长的Unicode序列后，可绕过微软的安全检查，如在一个标记为可执行的文件夹发出该请求，攻击者即可在服务器上运行可执行文件。

（6）木马系统各部分的作用是什么？

答：

1. 硬件部分：建立木马连接所必须的硬件实体。 控制端：对服务端进行远程控制的一方。 服务端：被控制端远程控制的一方。 INTERNET：控制端对服务端进行远程控制，数据传输的网络载体。  
   (2)软件部分：实现远程控制所必须的软件程序。 控制端程序：控制端用以远程控制服务端的程序。 木马程序：潜入服务端内部，获取其操作权限的程序。 木马配置程序：设置木马程序的端口号，触发条件，木马名称等，使其在服务端藏得更隐蔽的程序。  
   (3)具体连接部分：通过INTERNET在服务端和控制端之间建立一条木马通道所必须的元素。控制端IP，服务端IP：即控制端，服务端的网络地址，也是木马进行数据传输的目的地。控制端端口，木马端口：即控制端，服务端的数据入口，通过这个入口，数据可直达控制端程序或木马。

（7）计算机病毒结构有哪几个部分组成？分别的作用是什么？

答：计算机病毒主要由潜伏机制模块、传染机制模块和表现机制模块构成。

潜伏机制的功能包括：初始化、隐藏和捕捉；潜伏机制模块随着感染的宿主程序进入内存，初始化其运行环境，使病毒相对独立于其宿主程序，为传染机制做准备。. 利用各种隐藏方式躲避检测。. 不停地捕捉感染目标交给传染机制；不停地捕捉触发条件交给表现机制。.

传染机制的功能包括：判断和感染；传染机制首先通过感染标记判断侯选目标是否已被感染，一旦发现侯选目标没有感染标记，就对其进行感染。

表现机制的功能包括：判断和表现；表现机制首先对触发条件进行判断，然后根据不同的触发条件决定什么时候表现，如何表现。

（8）简述计算机病毒的传播、执行和激活机制

答：

传播机制:

这个部分是病毒运行的一个核心部分，计算机病毒就像人体病毒一样会传染，无法感染其他细胞的病毒，总会死在白血球抗体的手上，同样无法感染其他文件的计算机病毒也是无法生存的。在攻击传染这部分所使用的手段也是各有各的不同，但其主要还是实现象没有染毒的文件传染病毒这一目标。有些病毒通过拦截中断向量，对申请中断的文件一一检测，如没有感染病毒则对其进行感染再去执行原中断，如是带毒文件的话则直接执行原中断。采用这种手段的有Yankee病毒，Dir病毒还有臭名昭著的CIH。还有的病毒则是对磁盘上所有可感染的文件进行扫描，对其中没有感染的文件进行感染。最近十分流行的Nimda病毒在单机上就是采用这种感染方式的。随着网络的迅速发展越来越多的病毒开始通过网络传播，常见的蠕虫病毒是通过扫描Microsoft Outlook系列或是其他邮件软件的通讯簿选取其中的几位或是所有人的邮箱进行发送以染毒文件为附件的邮件。或是针对各种服务器的系统Bug进行感染传播。

执行机制:

早期的计算机病毒多带有恶作剧性质，例如播放一段音乐，或是在屏幕上显示一副图片等。而随这病毒编制者水平的提高，计算机病毒攻击的对象从简单的恶作剧向文件，系统发展，自从台湾同胞陈盈豪的CIH开创攻击计算机硬件的先河后。现在的计算机病毒的表现往往是极具破坏性的，不是系统崩溃就是硬件损坏。其表现时间也从早期的感染就发作到现在的没到具体时间才发作的。历史上有名的黑色星期五病毒每当星期五有逢每月的13号才发作。

激活机制:

主动传染：在系统运行时，计算机病毒通过计算机病毒载体，即系统的外存储器进入系统的内存储器，常驻内存，并在系统内存中监视系统的运行；病毒程序的引导模块，修改中断向量的入口地址，使之指向病毒程序的传染模块. 一旦系统执行磁盘读写操作或系统功能调用，病毒传染模块就被激活，当传染条件满足时，利用INT 13H将病毒自身传染给被读写的磁盘或被加载的程序。

（9）简述蠕虫病毒及其传播机制

答：

蠕虫病毒:蠕虫是一种可以自我复制的代码，并且通过网络传播，通常无需人为干预就能传播。蠕虫病毒入侵并完全控制一台计算机之后，就会把这台机器作为宿主，进而扫描并感染其他计算机。当这些新的被蠕虫入侵的计算机被控制之后，[蠕虫](https://baike.baidu.com/item/%E8%A0%95%E8%99%AB/4454380" \t "https://baike.baidu.com/item/%E8%A0%95%E8%99%AB%E7%97%85%E6%AF%92/_blank)会以这些计算机为宿主继续扫描并感染其他计算机，这种行为会一直延续下去。蠕虫使用这种递归的方法进行传播，按照指数增长的规律分布自己，进而及时控制越来越多的计算机。

传播机制:根据蠕虫病毒的程序其工作流程可以分为漏洞扫描、攻击、传染、现场处理四个阶段，首先蠕虫程序随机(或在某种倾向性策略下)选取某一段IP地址，接着对这一地址段的主机扫描，当扫描到有漏洞的计算机系统后，将蠕虫主体迁移到目标主机。然后，蠕虫程序进入被感染的系统，对目标主机进行现场处理。同时，蠕虫程序生成多个副本，重复上述流程。各个步骤的繁简程度也不同，有的十分复杂，有的则非常简单。

（10）蠕虫病毒、木马和传统病毒之间有什么区别？

答：

首先病毒，木马，蠕虫统称为电脑病毒。病毒（包含蠕虫）的共同特征是自我复制、传播、破坏电脑文件，对电脑造成数据上不可逆转的损坏。而木马独有特征是伪装成正常应用骗取用户信任而入侵，潜伏在电脑中盗取用户资料与信息。

病毒：

病毒是编制者在计算机程序中插入的破坏计算机功能或者数据的代码，能影响计算机使用，能自我复制的一组计算机指令或者程序代码。

木马：

木马也称木马病毒，是指通过特定的程序来控制另一台计算机。与一般的病毒不同，它不会自我繁殖，也专并不“刻意”地去感染其他文件，它通过将自身伪装吸引用户下载执行，向施种木马者提供打开被种主机的门户，使施属种者可以任意毁坏、窃取被种者的文件，甚至远程操控被种主机。

蠕虫病毒：

蠕虫病毒一种能够利用系统漏洞通过网络进行自我传播的恶意程序。它不需要附着在其他程序上，而是独立存在的。当形成规模、传播速度过快时会极大地消耗网络资源导致大面积网络拥塞甚至瘫痪。