使用动态包过滤的防火墙又叫什么类型的防火墙？以某种端口扫描类型为例说明该种防火墙相对静态包过滤防火墙的优越性。

使用动态包过滤的防火墙又叫状态检测防火墙。（1分）如果Internet上的扫描者对防火墙内部进行扫描，他发送一个SYN+ACK包，对这样的数据包：

1）静态包过滤防火墙不能判断它是对内部用户向外发送请求进行应答的数据包，还只是外部扫描内部的一个单独数据包，为了保证通信的畅通，它会让这个数据包通过，也就是该扫描数据包能通过静态包过滤防火墙。（3分）

2）状态检测防火墙由于能记住在此数据包进入前内部用户是否发送过一个对该外部主机的一个TCP SYN连接请求，如果有这样的请求则允许该外部的SYN+ACK响应包进入，对于现在的这种扫描行为，因为事先内部主机没有相应请求，则不允许该SYN+ACK包进入。

因此状态检测防火墙比静态包过滤防火墙优越。（3分）

**防火墙对经过的数据包有哪四种不同的处理方式？并简单比较一下不同处理方式的安全性。**

**①允许数据流通过；**

**②拒绝数据流通过：防火墙向发送者回复一条“该数据流已被拒绝”消息。**

**③将这些数据流丢弃：防火墙不给发送者任何提示信息——网络扫描所花费时间加长，发送者只能等待回应直至通信超时。**

④对不允许通过的数据包，将其引导到蜜罐中进行访问。

其中，对防火墙自身来说，丢弃数据包比拒绝数据包更安全；将防火墙引导到研究型蜜罐，有助于分析最新的攻击方式。

***什么是BPF、Libpcap、NPF、Winpcap？并简单阐述几者的关系。***

***BPF：UNIX下的分组捕获过滤机制（驱动）；***

***Libpcap：UNIX下的分组捕获函数库；***

***NPF：Windows下的分组捕获过滤机制（驱动）；***

***Winpcap：Windows下的分组捕获函数库；（各1分）***

***BPF/ Libpcap分别从UNIX平台移植到windows平台相应得到NPF/ Winpcap。***

***分组捕获函数库是在分组捕获过滤机制的基础上构建的。（3分）***

嗅探主机在接收不是发送给自己的数据包时，是如何绕过硬件过滤和IP地址过滤而将数据包捕获下来的？

1）系统正常工作时网卡驱动程序会判断所接收到的数据包的目标MAC地址是否是本机，嗅探主机网卡设置成混杂模式后，将绕过硬件过滤，网卡驱动程序将不再检查数据包的目标MAC地址是否匹配，局域网上传输的所有数据被全部接收。（3分）

2）正常主机的网络层驱动程序还要对其目的IP地址进行判断：如果是本地IP，则上传给传输层处理，否则丢弃。嗅探主机安装了分组捕获过滤机制（BPF）后，将绕过IP地址过滤，将网卡驱动程序捕获到的所有数据包发送给操作系统进行处理。（4分）

**B主机对A主机和C主机进行ARP欺骗攻击时，A发给C的数据包能否到达B的网络接口？B能否直接接收这种数据包？为什么？**

**能够到达（2分）可以直接接收（2分）**

**因为ARP欺骗工具同时也带有嗅探功能，具有嗅探协议栈，可以接收目的IP地址不是本机的数据包（3分）**

**最后两问如果从该数据包的目的IP地址不是B，从而B不能直接接收阐述也酌情给分。**

对于不是发送给自己的数据包，嗅探主机的协议栈是如何捕获的？

而普通主机的协议栈是如何接收发送给自己的数据包的呢？

请阐述上面两种情况下数据包的接收过程，以及是否对数据包进行回应。

对于不是发送给自己的数据包，嗅探主机的协议栈捕获过程如下：

1）嗅探主机网卡设置成混杂模式后，将绕过硬件过滤，网卡驱动程序将不再检查数据包的目标MAC地址是否匹配，局域网上传输的所有数据被全部接收。（2分）

2）嗅探主机安装了分组捕获过滤机制（BPF）后，将绕过IP地址过滤，将网卡驱动程序捕获到的所有数据包发送给操作系统进行处理。（2分）

但嗅探主机对捕获到的不是发送给自己的数据包并不进行任何回应。（2分）

普通主机的协议栈接收发送给自己的数据包的过程如下：

1）网卡驱动程序会判断所接收到的数据包的目标地址：不是本地主机，数据包将会被丢弃；是本地主机，数据包将被传到上层OS进一步处理。（2分）

2）网络层驱动程序还要对其目的地址进行判断：如果是本地IP，则上传给传输层处理，否则丢弃。（2分）

普通主机的协议栈是接收发送给自己的数据包后，当然要返回响应包进行回应。（2分）

**交换机上应该接入哪一种入侵检测系统？HIDS还是NIDS？从两者实现原理的不同加以分析。**

**应该接入NIDS，因为交换机是内外网数据通信和局域网内主机通信的流量必经之处，从而可以方便实现NIDS的基本功能——抓包。**

**而HIDS是安装在局域网内重要主机或服务器上的，通过审计日志来检测是否有针对该机的入侵。**

如何充分利用网卡混杂模式嗅探和ARP欺骗两种技术的优缺点，使之既能实现交换网络环境下的嗅探，又能较好地对捕获到的数据包进行分析和解码？请给出图示说明。

设想一个应用环境，比如学校的大型机房，就是一个交换式局域网，此时在其中的任一台主机上安装共享网络嗅探工具Wireshark，尽管它的嗅探功能非常强，但在交换网络中它也无能为力。这时它只能嗅探到从本机进出的数据包。而单独使用ARPspoof工具，虽然能嗅探到主机和主机（或网关）之间的数据流，但界面的不友好和弱的解码功能可能使得用户不方便得到、也可能得不到明文的用户帐号和口令。（6分）

在这种情况下，可以先用ARPspoof对目标主机和网关之间进行ARP欺骗，使它们的通信内容都从本地网卡通过。在进行了欺骗的同时打开Wireshark，这时再用Wireshark进行嗅探，用Wireshark强大的解码功能，就可以分析出大量的希望得到的敏感信息了。当然这种方案不但适用于交换网络环境中，也适用于共享网络环境中以及两者共存的更复杂环境中。（6分）

*什么是网卡的****隐秘模式？隐秘模式与混杂模式的区别是什么？管理员为何要将网卡设置成隐秘模式？NIDS的两块网卡是如何设置的？***

*网卡的****隐秘模式就是网卡设置成混杂模式的同时，外界扫描不到对应的IP地址。从而使得攻击者无法访问NIDS主机。（3分）***

***为方便管理，嗅探主机（如NIDS）设置两块网卡：***

***1）一块设置隐秘模式进行网络监听；***

***2）一块连接到独立网段，由管理员远程配置和管理。（各2分）***

**为了应对攻击者利用分片来穿越防火墙，防火墙在检测时进行数据包的重组（需要花费较长时间），仅当重组后的数据包符合通行规则时，数据包才可放行。攻击者如何利用防火墙这一处理机制对防火墙进行攻击呢？**

**攻击者短时间内向防火墙发送大量分片，耗尽防火墙的处理器资源，导致防火墙被拒绝服务攻击。**

IP欺骗攻击得以实现的两个基本前提是什么？为什么Windows系统中一般不能实现IP欺骗攻击？

IP欺骗就是伪造数据包源IP地址的攻击，基于两个前提：一是目前的TCP/IP网络在路由数据包时，不对源IP地址进行判断，这就给伪造IP包创造了条件；二是两台主机之间，存在着基于IP地址的认证授权访问（即基于rlogin命令的信任），就可以进行“会话劫持”。（5分）

因为Windows系统中不存在基于rlogin命令的信任，所以一般不能实现IP欺骗攻击。（2分）

直接基于局域网的ARP欺骗，黑客能进行哪些攻击（至少两种）？如果黑客已能够对局域网主机（或网关）进行会话注射，试想一下，他下一步就会进行何种攻击？

黑客能进行DNS欺骗和会话劫持等攻击。（4分）

黑客在会话注射的基础上可以进行注入欺骗网页、挂网马等攻击（答案不唯一）。（3分）

网络攻击中黑客可以将数据包真实的源IP地址修改成“假”的地址，请问在实现中源IP地址的“假”具体有哪两种不同的含义？SYN攻击使用的是其中的哪一种？

第一种“假”的含义是“假冒”：即假冒同一网段内另一台真实主机；（3分）

第二种“假”的含义是“虚假”：即假冒的源IP地址在该网段内是可路由但不可到达的。（4分）

**SYN攻击使用的是虚假IP地址。（1分）**

**使用SYN Proxy进行防范，外网主机对服务器进行SYN Flood攻击能成功吗？但它有什么不足呢？这种防范方法中，防火墙是包过滤型防火墙还是应用层网关型的？**

**使用SYN Proxy进行防范，外网主机对服务器进行SYN Flood攻击很难成功。（2分）**

**它的不足是防火墙自身可能被SYN Flood攻击成功。（2分）**

**这种防范方法中，防火墙是应用层网关型的。（3分）**

对一个程序设置SUID位的含义是什么？为什么一般会寻找如下一个程序的漏洞进行缓冲区溢出攻击：具有root执行权限且设置了SUID位？

SUID位赋予普通用户临时的权限提升，以保证用户可以在系统中完成某些特定的任务。（2分）

也就是说，如果某个设置SUID位的程序属主是root用户，那无论谁执行它，都相当于用root身份在运行。（2分）

这样，即使是只具有普通用户权限的黑客运行一个“具有root执行权限且设置了SUID位”的程序，并对其进行缓冲区溢出攻击成功，黑客就能直接获得root权限。（3分）

被加载的应用程序在内存中的影像包括哪三个区域？各有什么作用？其中哪些区域容易受到缓冲区溢出攻击？这些区域在读写上有什么共同特点？

被加载的应用程序在内存中的影像从内存低地址向内存高地址方向分别为：文本区、数据区、堆栈区。（2分）

1）文本区存储程序的执行代码以及只读数据；（1分）

2）数据区包括未初始化数据区（BSS）和已初始化数据区，前者存储静态分配的变量（static类型），后者存储程序的初始化数据；（1分）

3）堆栈区包括堆区和栈区，其中堆区用于存储程序运行过程中动态分配的数据块（调用malloc或calloc函数），栈区用于在函数调用中存储栈帧。（1分）

BSS、堆和栈都容易受到缓冲区溢出攻击。（2分）都是可以写的内存区域。

**C语言与其他程序设计语言不同，它是如何标识字符串结束的？没有使用长度参数对字符串边界进行检测会引起哪两方面问题？其中哪种会造成缓冲区溢出攻击？**

**C语言中字符串是以’\0’（NULL）结束的。（2分）**

**没有使用长度参数对字符串的边界进行检测会引起：**

**1）如果读取数组以外内容，会使程序得出错误结果**

**2）如果是写入，可能破坏该进程内存的其它内容（3分）**

**其中后者会造成缓冲区溢出攻击。（2分）**

