复习提纲

neta24b-tips.txt

neta24b-自荐题目所属知识点统计（仅供复习参考）.xlsx

答疑之crack7475爆破法.txt

tor和i2p的基本原理

Tor和I2P都是提供匿名通信服务的深网技术，它们的基本原理各有特色。

Tor的基本原理

Tor，即"The Onion Router"（洋葱路由器）的简称，是一种用于匿名交流的自由软件。其基本原理如下：

- **多层加密与转发**：Tor网络通过分布式节点（由志愿者运行的中继）转发流量。发送方的数据会经过多层加密，每经过一个节点，就解密一层，直到最后一层在出口节点解密后，数据才会发送到目标服务器。这种设计类似于洋葱的层次结构，因此得名"洋葱路由"。
- **匿名性与隐私保护**：由于数据在传输过程中连续更改路径，并在每个节点处部分解密，因此外界很难追踪数据源头及最终归宿，从而提供了较高的匿名性和隐私保护。

I2P的基本原理

I2P（Invisible Internet Project）是一种基于P2P（Peer-to-Peer）的匿名通信系统，其上运行着多种的安全匿名程序，支持的应用包括匿名的Web浏览、博客、电子邮件、在线聊天、文件分享等。其基本原理如下：

- **大蒜路由与分组交换**：I2P使用大蒜路由（一种洋葱路由的变体），通过不同的隧道将中间节点和目标节点分隔出来。数据被打包并层层加密，然后通过不同的TCP或UDP隧道交叉传输，最终在接收方重组为数据流。这种分组交换的方式有助于避免拥塞和服务中断，提高传输效率。
- **匿名性与隐私保护**：I2P通过多层加密和隧道技术来保护用户的隐私。由于节点间的通信既加密又流式传输，外部攻击者很难访问消息内容。此外，I2P网络中的所有对等体经常发送消息（端到端和网络维护消息），这些消息沿其路径改变大小和数据，进一步增强了匿名性。

Tor与I2P的比较

- **传输方式**：Tor基于电路切换，而I2P基于分组交换。Tor经常应对高拥塞导致高延迟，而I2P中的分组交换有助于避免拥塞和服务中断，更适合大型文件传输。
- **出口节点**：Tor网络中有许多退出节点，主要用于为用户提供匿名访问外部网站的服务。而I2P中只有少数Outproxies（Tor术语中的退出节点）作为标准Internet的网关运行，因为I2P专为I2P网络内的匿名通信而设计。
- **传输协议**：Tor的中继之间主要使用TCP连接，而I2P的路由之间既使用TCP连接又使用UDP进行数据传输。这使得I2P对于某些深度包检测（DPI）设备来说更难跟踪。

综上所述，Tor和I2P都提供了强大的匿名通信服务，但它们在传输方式、出口节点和传输协议等方面存在差异。用户可以根据自己的需求和偏好选择适合自己的匿名通信工具。

“网络”“攻击”与“防范”

“攻击” 5步走：

1.社会工程，获取特定目标（比如已知qq号码）或非特定目标（机会主义者要控制一批机器当僵尸机）

2.扫描，获取目标漏洞信息

3.入侵

黑客入侵后会造成什么危害?

4.拖库或潜伏监控（当跳板去攻击别人、挖矿、作为僵尸网络的一部分、植入勒索软件、利用入侵环境存放违法文件、监视用户的键盘）

5.消除痕迹

“防范”：

1.不要泄露社工信息

2.打补丁堵漏洞

3.观察网络流量，识别正常、异常、攻击，以及拖库等行为(snort,mydlp)

4.对流量载荷中可能的泄露进行检查（ntop一般不关心载荷，用squid+sslbump+icap）

5.日常性的工作，比如勤于观察日志、关心技术论坛和安全新闻等

“网络”系统

1.访问控制，除了os、app登录等，还有iptables、selinux等

2.传输加密，https，vpn，ssh

3.掩盖IP来源，各种proxy和跳转，典型就是vps、tor、i2p

4.对流量进行其他混淆，比如使用ping载荷、dns载荷等

5.其他，比如利用Botnet、APT、供应链攻击等

网络攻击的深层技术（脱离脚本小子的低级趣味）

1.内存安全，以缓冲区溢出为代表，各种exploit和shellcode基本都是这一类

2.二进制分析和逆向，各种平台windows/linux/Android/macos/ios上，可以发现内存安全等等各方面的问题

3.外挂技术，一般也是基于二进制分析的成果进行

4.网络攻击一般不去破译密文，因为密码算法没有容易破解的，但是cmap5.com确实很好用。

安全管理和法规

1.安全管理：“bs7799” vs. 国内网络安全技术标准 vs. 企业安全管理制度

2.国内法规：网络安全法、数据安全法、~个人信息保护法

增补

1.DDOS

DDoS攻击是DoS（拒绝服务攻击）的升级版，它通过利用Internet上现有机器及系统的漏洞，攻占大量联网主机，使其成为攻击者的代理。当被控制的机器达到一定数量后，攻击者通过发送指令操纵这些攻击机同时向目标主机或网络发起DoS攻击，大量消耗其网络带宽和系统资源，导致该网络或系统瘫痪或停止提供正常的网络服务。DDoS攻击具有分布式特征，因此具有比DoS更强大的攻击力和破坏性。

2.ASLR(address space layout randomization)

ASLR是一种内存保护机制，旨在通过随机化进程的地址空间布局来增加攻击者预测系统资源地址的难度，从而提高系统的安全性。在Linux等操作系统中，ASLR的实现原理主要包括对栈、内存映射段（包括共享库）和堆的起始地址进行随机化。这样，即使攻击者能够利用某个漏洞，他们也很难预测到目标系统上的具体内存地址，从而降低了攻击成功的可能性。

3.撞库

撞库是黑客通过收集互联网已泄露的用户和密码信息，生成对应的字典表，然后尝试批量登录其他网站，以获取一系列可以登录的用户。很多用户在不同网站使用的是相同的账号密码，因此黑客可以通过获取用户在A网站的账户从而尝试登录B网站。撞库攻击是网络交易普遍存在的一个主要风险，可以采用大数据安全技术来防护，比如用数据资产梳理发现敏感数据，使用数据库加密保护核心数据等。

4.如何安全地分析计算机病毒

安全地分析计算机病毒需要遵循一定的步骤和原则：

准备阶段：确保分析环境的安全性，包括使用虚拟机、沙箱等隔离技术来避免病毒对真实系统的破坏。同时，准备好必要的工具，如病毒扫描软件、调试器等。
病毒样本获取：从可靠来源获取病毒样本，避免从不可信渠道下载或接收未知文件。
静态分析：在不执行病毒代码的情况下，通过查看病毒文件的二进制结构、代码逻辑等来分析其功能和行为。这可以使用反汇编工具、逆向工程软件等工具来完成。
动态分析：在受控环境中执行病毒代码，观察其行为和产生的影响。这需要使用调试器、网络监控工具等来监控病毒的运行过程。
记录与分析：详细记录分析过程中的发现，包括病毒的行为模式、感染机制、传播方式等。然后对这些信息进行分析和总结，以便制定有效的防御措施。
防护与清理：在分析结束后，及时清理系统中的病毒残留，并确保系统已经得到充分的保护。同时，将分析结果和防御措施分享给相关团队或组织，以便他们也能采取相应的措施来防范类似的病毒攻击。

5.渗透测试

渗透测试是一种模拟黑客攻击的网络安全活动，它通过模拟恶意黑客的攻击行为，对计算机系统、网络或应用程序进行模拟入侵，以评估其安全性。渗透测试旨在发现潜在的安全漏洞，并帮助组织采取相应的措施来修复它们，从而提高信息系统的安全性。渗透测试的原理基于黑客的攻击手法，包括尝试利用漏洞、社交工程、密码猜测等方式来获取未经授权的访问权限。

6.蜜罐（honeypot）

蜜罐是一种安全技术，用于模拟真实系统的漏洞、弱点，并在现实环境下被攻击、入侵，以便收集有关攻击者和攻击技术的信息。蜜罐技术的基本原理是制造一个看似真实的系统，将其置于网络中，用于吸引并欺骗攻击者进入。一旦攻击者进入蜜罐系统，他们的行为将被记录下来并分析，从而获得有关攻击者和攻击技术的信息，以便开展更好的安全防护。蜜罐技术具有欺骗性高、捕获能力强等特点，可以帮助企业了解攻击者的动机和手法，并及时发现潜在的安全威胁。

工具(集)众多，典型举例比如：

1.泄露案和社工库
2.安全扫描器以及一些ctf常用工具
3.metaspolit
4.vulhub

Vulhub提供了一系列预先配置好的、包含已知漏洞的Docker环境。这些环境覆盖了多种操作系统和服务，如Web服务器、数据库系统等，旨在帮助用户在安全可控的环境中学习和实践网络安全技能。通过Vulhub，用户无需具备复杂的网络安全专业知识或技术背景，即可轻松部署这些环境，进而学习如何发现和利用安全漏洞。

5.ntopng
6.snort
7.squid-icap-clamav
8.mydlp

1.一个大型百万级的app如何防止黑客入侵

APP防止黑客入侵是一个综合性的任务，需要从多个方面入手，以下是一些具体的措施：

一、了解常见的攻击类型

首先，了解常见的黑客攻击类型对于制定防御策略至关重要。常见的APP攻击类型包括：

1. **中间人攻击**：攻击者在用户和服务器之间拦截通信，窃取敏感信息。
2. **DDoS攻击**（分布式拒绝服务攻击）：大量的请求涌向APP服务器，使其瘫痪，导致合法用户无法正常使用APP。
3. **数据篡改**：攻击者通过修改APP与服务器之间传输的数据，或者篡改APP本地存储的数据，来达到破坏APP正常功能或获取非法利益的目的。
4. **SQL注入攻击**：通过在输入框中输入恶意的SQL语句，绕过APP的安全验证，获取数据库中的敏感信息。
5. **XSS攻击**：通过在应用界面中注入恶意的脚本，从而获取用户的敏感信息。
6. **信息泄露攻击**：黑客通过各种手段获取到移动应用中存储的敏感信息，例如用户的账户密码、个人资料等。
7. **窃听攻击**：黑客通过窃取应用中传输的数据，从而获取用户的敏感信息。

二、安全防护

针对上述攻击类型，可以采取以下措施来加强APP的网络安全防护：

1. **使用安全的通信协议/网络传输过程中的信息加密**：
   - 使用HTTPS协议，通过SSL/TLS加密技术对APP和服务器之间的数据传输进行加密，防止数据在传输过程中被窃取或篡改。

2. **部署防火墙和入侵检测系统**：
    - 防火墙可以阻止未经授权的网络流量进入APP服务器。
    - 入侵检测系统（IDS）可以实时监测网络中的异常活动，及时发现并告警可能的攻击行为。
3. **加密存储敏感数据**：
    - 使用强加密算法对APP中的敏感数据进行加密存储，如AES算法。
    - 确保加密密钥的安全存储和管理。
4. **实施数据完整性验证机制**：
    - 通过使用哈希函数等技术，对数据进行哈希计算，并在数据传输或存储过程中验证哈希值的一致性。
    - 如果数据被篡改，哈希值就会发生变化，从而可以及时发现数据篡改攻击。
5. **进行严格的代码审查**：
    - 在APP开发过程中，集成代码审查工具，检查代码中是否存在可能导致安全漏洞的问题，如SQL注入漏洞、跨站脚本漏洞等。
    - 结合人工审查，确保代码的安全性。
6. **对用户输入进行严格的验证和过滤**：
    - 对于APP中的输入框等用户输入接口，要限制输入的类型和长度，过滤掉可能包含恶意代码的输入。
    - 只允许用户在输入框中输入合法的字符，对于特殊字符进行转义或拒绝。
7. **精确的授权管理**：
    - 根据用户的角色和权限，为用户分配不同的APP功能访问权限。
    - 限制用户对敏感信息的访问和操作权限。
8. **及时修复安全漏洞及时更新系统**：
    - 开发团队要及时关注安全漏洞信息，并修复发现的安全漏洞。
    - 发布更新版本，确保用户能够及时获得最新的安全防护。
9. **对发布的软件加壳，防止逆向**：

2.如何防止自己的社工信息被泄露

防止自己的社工信息被泄露是一个综合性的任务，需要从多个方面入手。以下是一些具体的建议：

一、加强个人信息保护意识

- **谨慎分享个人信息**：避免在公共场合、社交媒体或不可信的平台上随意分享出生日期、地址、手机号码等隐私信息。这些信息一旦泄露，很容易被不法分子利用。
- **少用真实信息注册网站**：在注册网站或APP时，尽量使用昵称和虚拟手机号码，避免使用真实姓名、身份证号等敏感信息。

二、提升账户安全性

- **使用复杂且不同的密码**：为不同的账户设置复杂且独特的密码，避免使用容易猜测或常见的密码。
- **开启双因素认证**：在支持双因素认证的网站或APP上开启此功能，增加账户的安全性。
- **定期检查账户安全**：定期登录你使用的各个网站和APP，查看是否有未经授权的登录记录或可疑的账户变更。

三、警惕网络钓鱼和恶意软件

- **谨慎点击链接和下载附件**：在阅读邮件和信息时，务必查看发件人和链接地址，避免点击来历不明的链接或下载未经验证的附件。
- **安装可靠的安全软件**：在设备上安装可靠的安全软件，如防病毒软件和防火墙，以防御恶意软件的攻击。

四、定期监控和更新个人信息

- **检查个人信用报告**：定期向征信机构查询个人信用报告，查看是否有可疑的查询记录或不实信息，以发现潜在的身份被盗用情况。
- **更新个人信息**：如果个人信息发生变化（如手机号码更换、住址迁移等），及时在相关网站或APP上进行更新，以确保信息的准确性。

五、采取额外的防护措施

- **合理规划手机号使用**：根据个人需求合理规划手机号的使用，如为家人、工作、金融等用途分别配备不同的手机号，以降低信息泄露的风险。
- **使用隐私保护工具**：在社交媒体上使用隐私设置工具，限制陌生人查看你的个人信息和动态。

六、及时应对信息泄露事件

- **报案**：一旦发现个人信息泄露，应立即向公安机关报案，以便及时采取措施防止损失扩大。
- **收集证据**：在信息泄露后，留心并记录可能收到的邮件、电话等有用信息，这些信息可能对于维权和追踪不法分子至关重要。

3.攻击五步走，写自己的具体使用感受

4.缓冲区，栈内临时变量，call入栈，基址等画出来，或者用代码表示一下

缓冲区溢出（栈溢出）

堆栈：
堆栈是一个特定的存储器或寄存器，其本质就是存储数据的内存。在实际应用中，堆栈会用于存储临时变、函数调用、中断切换时保存和恢复现场数据。

物理地址	堆栈示意图	
012FF178	00000000	
012FF17C	00000000	
012FF180	00000000	
012FF184	00000000	
012FF188	00000000	
012FF18C	00000000	
012FF190	00000000	
012FF194	76ED4EFC	
012FF198	012FF170	
012FF19C	013A4010	
012FF1A0	00000000	
012FF1A4	00000000	
012FF1A8	013A4010	
012FF1AC	012FF158	
012FF1B0	9600BD92	
012FF1B4	012FF1B8	
012FF1B8	000A00E8	
012FF1BC	FFFFFFFE	
012FF1C0	004F0044	
012FF1C4	76EDD0B0	
012FF1C8	012FF1BC	esp
012FF1CC	9600A832	
012FF1D0	FFFFFFFF	
012FF1D4	00000000	
012FF1D8	012FF1E8	ebp

如图所示，堆栈就是内存中的某一片区域，每一个堆栈的内存单元都有唯一一个物理地址，且内存单元的大小都统一是4个字节大小。

> 并不是esp和ebp之间的范围才是堆栈，整个区域都是堆栈，而esp（栈顶）和esp（栈底）的作用只是定位数据的位置。

栈溢出就是发生在这样一个存储区域中。在了解了发生溢出的环境之后，了解溢出就相当于了解了一半了，另一半原因与使用的call指令有关。

Call指令：

call指令通常用于调用函数。用下面的例子理解。

cpu执行的程序

物理地址	机器码（十六进制）	汇编指令
76EA70D2	E8 C9120500	call <ntdll.RtlFillMemoryUlong>
76EA70D7	804B 02 04	or byte ptr ds:[ebx+2],4
76EA70DB	8B55 D8	mov edx,dword ptr ss:[ebp-28]
76EA70DE	8DB7 C0000000	lea esi,dword ptr ds:[edi+C0]
76EA70E4	83BF B4000000 00	cmp dword ptr ds:[edi+B4],0

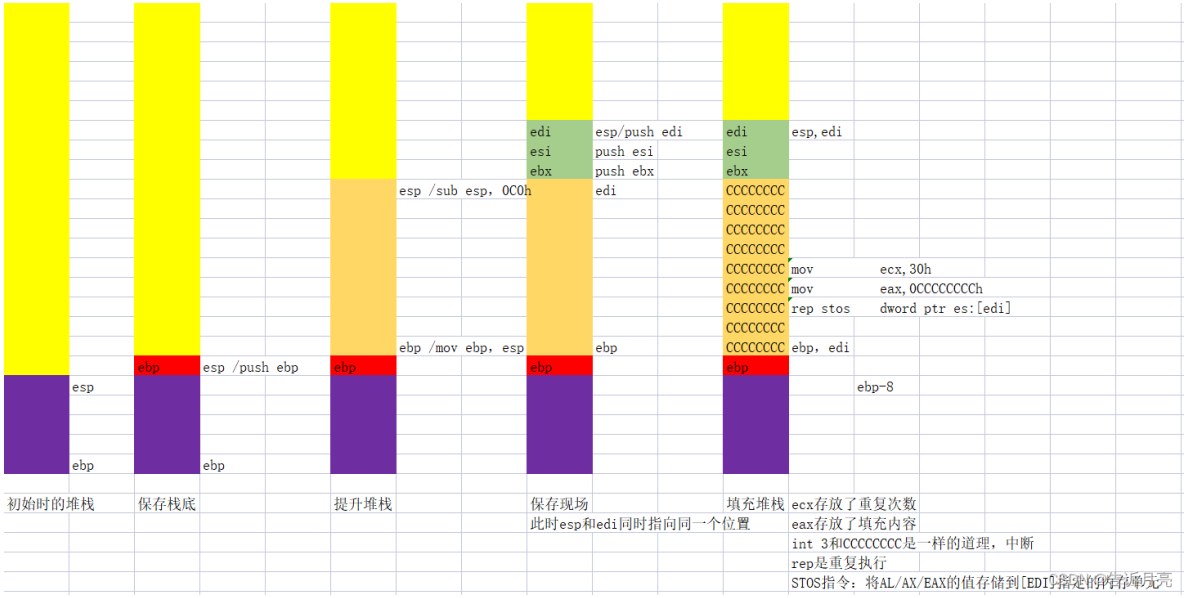
对应堆栈变化

物理地址	堆栈		物理地址	堆栈		
012FE840	FFFFFEBC		012FE840	FFFFFEBC		
012FE844	006F00C0		012FE844	006F00C0		
012FE848	5080016B		012FE848	76EA70D7	esp	call <ntdll.RtlFillMemoryUlong>
012FE84C	018705D0	esp	012FE84C	018705D0		
012FE850	00000A10		012FE850	00000A10		
012FE854	FEEEFEEE		012FE854	FEEEFEEE		
012FE858	E1D9B1E2		012FE858	E1D9B1E2		
012FE85C	00000100		012FE85C	00000100		
012FE860	00000108		012FE860	00000108		
012FE864	01870000	ebp	012FE864	01870000	ebp	
在执行call之前的堆栈			执行完call之后的堆栈			

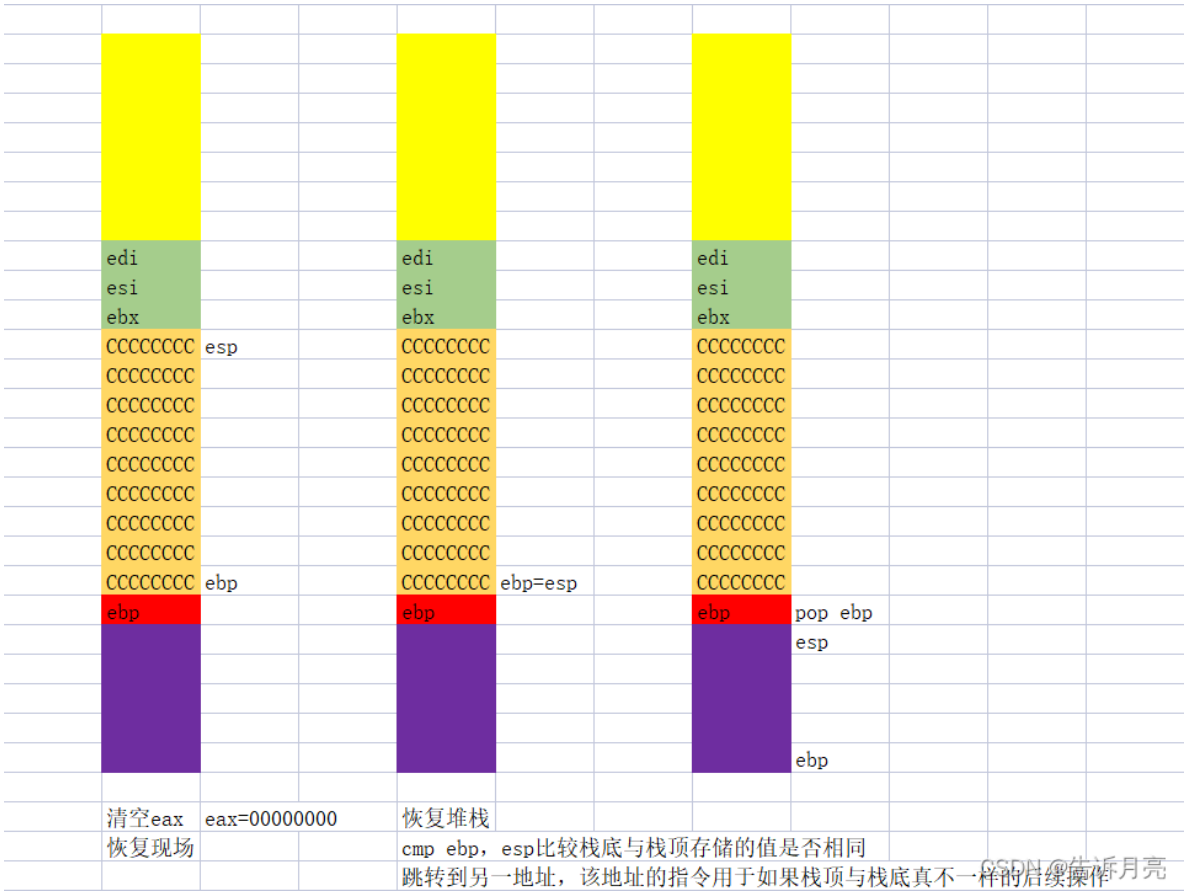
> call指令在调执行时，先看call指令的长度和call所在的物理地址，这样才能计算出call指令的下一个指令地址。并且会把下一个指令地址存储到堆栈中。

之后cpu就会在堆栈中开辟出一段存储空间用于调用函数、存储临时变量。这里不细说。

函数调用时对堆栈的使用（图中用CCCCCCCC填充的内存单元就是缓冲区，存放局部变量就是在这里）



函数调用完后对堆栈的处理



小结：

1、函数调用完之后数据还是存储在堆栈当中，没有被真的删除，变化的只是esp和ebp指向的位置。

2、函数调用完之后，esp和ebp的位置会回到函数开始前的位置。也就是说我们的esp指向的位置是我们在调用函数执行call指令时所存在堆栈中的返回地址。

栈溢出（缓冲区溢出）：
在了解完堆栈结构和call指令后，我们可以清楚的知道如果我们在调用一个函数时传入一个数据，如果这个数据大小超过我们所分配的缓冲区大小，那么就会造成缓冲区溢出。

堆栈图可以帮助我们更好的理解溢出的发生。

buffer name	0012FF78	esp	buffer name	0012FF78	
edi			edi		
esi			esi		
ebx			ebx		
	ebp-4ch			ebp-4ch	
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			cccccccc		
cccccccc			abcd		绿色的堆栈区是我们设置的存储变量的位置
cccccccc			efgh		如果我们按照规定要求输入abcdefg就不会有溢出
	ebp		abcd	ebp	但是如果我们输入abcdefgabcdefg就会超过规定位置造成溢出
call-adr			efgh		
			call strcpy		

图中左边的堆栈可以看到给我们所分配的存储局部变量的位置是[ebp-14h]至[ebp-4]这个范围中，但是当我们输入的数据超过这个范围之后，多余的数据就被安排到了[ebp-adr]这个堆栈单元中了。[ebp-adr]这个单元中存放的是我们的返回地址（call语句的下一条语句的地址）。

缓冲区溢出的危害

缓冲区溢出最基本的影响就是会造成系统报错，更大的危害是利用缓冲区溢出执行非程序本身的操作。

通过修改存在堆栈中的返回地址，可以跳转到我们想要执行的程序的物理地址。我们也可以计算好堆栈的空间，然后输入一串我们编写好的具有特殊功能的Shellcode，通过返回地址调整esp指向位置去执行我们编写的shellcode。

缓冲区的利用方法

普通的数据传入：通过传入大量的数据造成溢出。（就像上面的图）
数组越界：通过往超过数组空间大小的位置传入数据，可以指定修改堆栈中的数据。

```
#include "stdafx.h"//版本：vc 6.0
```

```
void HelloWorld(){
    printf("HelloWorld");
    getchar();
}

void Fun(){
    int arr[5]={1,2,3,4,5};
    arr[6]=(int)HelloWorld;
}

int main(int argc, char* argv[])
{
    Fun();
    return 0;
}
```

这串代码就是用到了数组越界的方法。我们声明的数组大小为五，所以在堆栈中分配的五个堆栈内存单元，但是我们向下标为六的数组空间中存入我们Hello World函数的地址，这就造成了越界。

edi	esp			edi	esp		
esi				esi			
ebx				ebx			
	ebp-54h				ebp-54h		
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
cccccccc				cccccccc			
1	mov dword ptr [ebp-14h],1			1			
2	mov dword ptr [ebp-10h],2			2			
3	mov dword ptr [ebp-0ch],3			3			
4	mov dword ptr [ebp-8],4			4			
5	mov dword ptr [ebp-4],5			5			
	ebp				ebp		
call-adr				0040100a	ebp+4		
int arr[5]={1,2,3,4,5}				arr[6]=(int)Helloworld			

当程序开始开始通过返回地址返回时，就会跳转到Hello World函数处，而Hello World函数并没有返回到主函数的返回地址，所以就会造成报错。

防止缓冲区溢出的方法—GS

防止缓冲区溢出和数组越界的方法还有很多，以上也仅是缓冲区溢出的基本知识。

5.校园网网络卡到选不了课，带宽用VOD，每个ip限速，因为P2P协议，网络传输用whiresharp，ntop（ng）主要看宏观统计，怎么用的

针对您提出的校园网网络拥堵问题，以及提出的采用VOD（视频点播）系统、IP限速、P2P协议拥堵原因、Wireshark和ntopng等工具的应用方案，以下是一个综合的解决方案和工具使用说明：

一、解决带宽问题：采用VOD系统

VOD系统可以通过分布式架构和有效的节目缓存策略，降低对主干网络的带宽要求。在校园网环境中，可以将VOD系统部署在网络末梢，如各个教学楼或宿舍楼，使大量用户的选课请求在本地完成，从而减少网络拥堵。同时，通过预测课程点播概率，优化节目（此处可理解为课程视频）的存储和轮换，进一步提升系统性能。

二、解决速度问题：采用IP限速

IP限速是一种有效的网络流量管理策略，可以通过限制每个IP地址的带宽使用，防止个别用户占用过多带宽，从而影响其他用户的网络体验。在校园网中，可以部署IP限速策略，为每个用户或设备设置合理的带宽上限，确保选课等关键业务的流畅进行。

三、P2P协议拥堵原因及解决方案

拥堵原因：

P2P（Peer-to-Peer）技术允许用户直接通过互联网进行文件分享和数据传输，无需中央服务器。这种去中心化的网络结构在提高效率和资源利用率的同时，也带来了网络拥堵问题。当大量节点同时下载或上传数据时，网络带宽可能会达到饱和，导致传输速度显著下降。在校园网环境中，如果大量用户同时使用P2P软件（如过去的迅雷等）进行下载或上传，就会引发网络拥堵，影响选课等关键业务的正常进行。

解决方案：

1. 限制或禁止P2P流量：在校园网中，可以通过部署网络流量管理设备或软件，限制或禁止P2P流量的传输，从而减轻网络负担。
2. 引入P2P限速工具：对于必须允许P2P流量的场景，可以引入P2P限速工具，通过智能限速算法，动态调整P2P流量的速率，确保网络资源的合理分配和有效利用。

四、使用Wireshark观察微观流量

Wireshark是一款强大的网络协议分析工具，可以用于捕获和分析网络数据包。在校园网环境中，可以使用Wireshark来观察微观流量，即每个数据包的内容、来源和去向等信息。通过Wireshark，可以定位网络拥堵的源头，分析网络流量的构成和分布情况，为制定有效的网络流量管理策略提供数据支持。

使用方法：

1. 打开Wireshark软件。
2. 选择要捕获数据包的网络接口。
3. 开始捕获数据包。
4. 使用过滤器功能，筛选出与选课相关的网络流量。
5. 分析数据包的内容、来源和去向等信息，定位网络拥堵的源头。

五、使用ntopng查看宏观统计

ntopng是一款开源的网络流量监控和分析工具，可以提供网络流量的宏观统计数据，如每个IP地址的流量使用情况、网络协议的分布情况等。在校园网环境中，可以使用ntopng来监控和分析网络流量的宏观情况，为制定网络流量管理策略提供决策支持。

使用方法：

1. 安装并配置ntopng软件。

2. 启动ntopng服务。
3. 通过Web浏览器访问ntopng的Web界面。
4. 在Web界面中查看网络流量的宏观统计数据，如流量排名、协议分布等。
5. 根据统计数据，分析网络流量的特点和趋势，制定有效的网络流量管理策略。

综上所述，通过采用VOD系统、IP限速、限制或禁止P2P流量（或引入P2P限速工具）、使用Wireshark观察微观流量和使用ntopng查看宏观统计等策略，可以有效地解决校园网网络拥堵问题，确保选课等关键业务的流畅进行。

6.https里s是ssl只能看ip头看不到tcp头

关于HTTPS和TCP头
HTTPS：HTTPS是一种安全的网络通信协议，它在HTTP的基础上加入了SSL/TLS加密层，用于保护数据的传输安全。在HTTPS通信中，用户只能看到IP头信息，而无法直接看到TCP头信息，因为TCP头信息被加密在SSL/TLS层中。
TCP头：TCP头包含了TCP连接的各种信息，如源端口号、目标端口号、序列号、确认号等。这些信息对于了解TCP连接的状态和性能是非常重要的。然而，在HTTPS通信中，由于TCP头被加密，因此无法直接查看。

7.看一下这些法律的时间，以及遇到具体事件触犯什么法律，有什么法律依据

安全问题1备份2免责

几个安全的法律的名字：

《中国人民共和国网络安全法》

《数据安全法》

关于网络安全的法律，以下是一些重要的法律法规的名称：

1. **《中华人民共和国网络安全法》**：自2017年6月1日起施行，共7章79条。该法特别确立了以“告知—同意”为核心的个人信息处理规则，落实国家机关保护责任，并加大了对违法行为的惩处力度。
2. **《中华人民共和国数据安全法》**：该法旨在保障国家数据安全，促进数据开发利用，保护个人、组织的合法权益，维护国家主权、安全和发展利益。
3. **《中华人民共和国个人信息保护法》**：此法保护个人信息权益，规范个人信息处理活动，促进个人信息合理利用，并根据宪法，制定相关法律法规。

4. **《中华人民共和国保守国家秘密法》**：此法旨在保守国家秘密，维护国家安全和利益。
5. **《中华人民共和国国家安全法》**：此法规定了维护国家安全的相关内容，包括网络安全在内的多个方面。
6. **《中华人民共和国电子签名法》**：此法规范电子签名的行为，确保电子签名的真实性、完整性和不可否认性，从而保护网络安全和交易安全。
7. **《关键信息基础设施安全保护条例》**：该条例旨在保护关键信息基础设施免受攻击、侵入、干扰和破坏，保障其安全稳定运行。

此外，还有以下与网络安全相关的法规和管理规定：

1. **《计算机信息系统国际联网保密管理规定》**
2. **《涉及国家秘密的计算机信息系统分级保护管理办法》**
3. **《互联网信息服务管理办法》**
4. **《计算机信息网络国际联网安全保护管理办法》**
5. **《中华人民共和国计算机信息系统安全保护条例》**
6. **《互联网上网服务营业场所管理条例》**
7. **《网络安全审查办法》**：为了进一步保障网络安全和数据安全，维护国家安全而制定的部门规章。

这些法律法规共同构成了我国网络安全法律体系的基石，为保护公民、组织和国家在网络空间中的合法权益提供了有力的法律保障。