PRE：

各位同学好，我是来自第七小组的刘轩铭，今天由我来给大家带来我们小组的Advanced Topic展示。我们小组的选题是缓存区溢出，BufferOverflow。以下是我们的小组成员名单。

今天我将从这几个方面给大家介绍缓存区溢出这个主题。

其实，经过老师在malicious code一节的介绍，以及对于第二次作业的完成，大家应该对buffer overflow这个概念已经有一定的了解了。具体而言，缓存区溢出是指：当程序向缓存区写入数据的时候，数据超出了缓存区的范围，覆盖了相邻区域的数据。这个概念应该是很好理解的。那么，我们说缓存区溢出是一种很危险的行为，黑客又是怎么利用这一漏洞的呢？我们知道，在存储空间中，指令和存储的数据都是以二进制的形式存在的，也就是说，原本写入缓存区打算用来存储的数据也有可能被当作指令进行执行。那么，黑客们就可以用事先准备好的一些恶意代码，覆盖原本安全的数据，从而完成一些有预谋的恶意攻击。另外，其实这样的攻击并不需要非常高深的计算机知识，也非常容易完成。实际上，这样的攻击在网络上是非常常见的。据数据统计，在2017年的上半年，有百分之十四的安全漏洞与缓存溢出相关，而这一数据也排在了所有类型攻击的第三位。所以对于我们程序员来说，了解这方面的知识是十分有意义的。

在当前网络与分布式系统安全中，被广泛利用的都是缓冲区溢出，其中最著名的例子是1988年利用fingered漏洞的蠕虫，这个例子老师上课已经给出了一些介绍。而缓冲区溢出中，最为危险的是堆栈溢出，因为入侵者可以利用堆栈溢出，在函数返回时改变返回程序的地址，让其跳转到任意地址，带来的危害一种是程序崩溃导致拒绝服务，另外一种就是跳转并且执行一段恶意代码，比如得到shell，然后为所欲为。

首先我们介绍栈溢出，在图中的c程序中，我们在子函数foo中调用了strcpy函数，大家应该对这个库函数不陌生。但是我们很少注意到，这个函数其实是不安全的，因为他并没有进行字符串长度的验证。另一方面呢，在计算机系统课程中我们学过，主程序调用子程序的时候，会把相关的参数，原本的寄存器的值，以及预备的缓存区这些数据压入栈中，栈会向着内存地址低的方向生长。那么，假设我们运行这个程序，将一大串数据放入栈中，如果数据长度较小那没有什么问题，但如果数据过长，或者故意放入过长的数据，那么绿色和红色的区域就有可能被覆盖。请注意红色部分是原本的return ip，也就是预期的子程序调用完成后应该回到的位置。那么假设对这个地址的覆盖是有预期的，则黑客可以实现带有特定目的的一个跳转。

那么有可能用户会获取一个超级用户的权限，也有可能程序跳到了一个很危险的地方，比如把某个应用程序给删了，或者也有可能，被覆盖的区域涉及某些很重要的参数，从而引起了更严重的bug。

下面是一个实际的例子，codeRedworm。他是利用微软的web server的漏洞进行的缓存溢出攻击。黑客在浏览器提交的请求中包含的文件名存储在局部变量缓冲区中，若它很长，超过600个字符时，比如这里N之后的数据，那就会导致局部变量缓冲区溢出，覆盖返回地址空间，使IIS崩溃。EIP会被 0x7801CBD3 重写，它导致程序流重定向回堆栈。堆栈上的代码将会跳到蠕虫代码。更进一步，如果在缓冲区中植入一段精心设计的代码，可以使之以系统超级权限运行。

那么其他的一些例子也是存在的。

接下来我们再介绍一下堆溢出的概念。我们知道，操作系统中的堆主要是用来分配动态内存的。比如我们熟悉的malloc函数便是在堆中分配了一块区域给一个特定的指针。和栈不同的是，堆中数据的生长方向，是朝着内存地址增大的方向。同样，堆也有一个堆指针进行牵引。假设我们的动态内存中的数据超出了范围，就有可能覆盖这个指针。从而有可能我们将被引入一个未知的区域，这就有可能引起系统的死机。

为了加深大家对于缓存区溢出概念的理解，接下来，请大家观看我们组的张旭学长录制的一个演示视频。

最后，我们再来看一看一些预防缓存溢出的方法。首先，我们可以选择一些更为安全的编程语言和方式。比如C和C++，以及汇编这样的编程语言是可以直接访问内存的，而且没有进行这方面的检测，而python等语言则不存在这样的问题。另外，我们在调用函数时，应该进行适当的包装，检测是否缓存溢出。第三，Address space layout randomization，如刚刚视频里提到的ASLR技术，这样的系统防范措施，也可以帮助我们一定程度上防范攻击。此外，一些检测工具，如gcc中的stack guard，也可以帮助到我们。最后要提到的是对栈进行不可执行化，这种方法主要是让操作系统将数据和指令分隔开，禁止系统将堆栈中的代码当作指令进行执行，这样就可以从源头上切断缓存溢出了。然后各种方式都有着自己的优点和缺点，我们应该在工作中合理的进行取舍，有兴趣的同学可以自行在网上进一步了解。

那么总而言之，缓存溢出是一种当今很常见的攻击方式。那么，作为一名合格的程序员，我们当然有必要进行这些方面的了解和学习。在学习和工作中，我们应当做好这一方面的预防工作。

以上就是我们小组的展示环节，谢谢大家。