CLASS 9

Return to libc

吴瑞欣-E41614059

1、名词解释：ARP cache poisoning，ICMP Redirect Attack，SYN Flooding Attack，TCP Session Hijacking。如果想监听局域网内另外一台机器，一般先要进行什么步骤?

ARP cache poisoning

地址解析协议中毒（ARP 中毒）是一种攻击形式，攻击者通过伪造的 ARP 请求和回复数据包更改目标计算机的 ARP 缓存来更改媒体访问控制（MAC）地址并攻击以太网 LAN。这会将层 - 以太网 MAC 地址修改为黑客已知的 MAC 地址以对其进行监控。由于 ARP 回复是伪造的，因此目标计算机会无意中将帧发送到黑客的计算机，而不是将其发送到原始目标。结果，用户的数据和隐私都受到损害。用户无法检测到有效的 ARP 中毒尝试。

ICMP Redirect Attack

路由器使用 ICMP 重定向消息为主机提供最新的路由信息，主机最初具有最少的路由信息。 当主机接收到 ICMP 重定向消息时，它将根据消息修改其路由表。 由于缺乏验证，如果攻击者希望受害者以特定方式设置路由信息，则可以向受害者发送伪造的 ICMP 重定向消息，并欺骗受害者修改其路由表。

SYN Flooding Attack

这要从操作系统的 TCP/IP 协议栈的实现说起。当开放了一个 TCP 端口后，该端口就处于 Listening 状态，不停地监视发到该端口的 Syn 报文，一旦接收到 Client 发来的 Syn 报文，就需要为该请求分配一个 TCB（Transmission Control Block），通常一个 TCB 至少需要 280 个字节，在某些操作系统中 TCB 甚至需要 1300 个字节，并返回一个 SYN ACK 命令，立即转为 SYN-RECEIVED 即半开连接状态

从以上过程可以看到，如果恶意的向某个服务器端口发送大量的 SYN 包，则可以使服务器打开大量的半开连接，分配 TCB，从而消耗大量的服务器资源，同时也使得正常的连接请求无法被相应。而攻击发起方的资源消耗相比较可忽略不计。

TCP Session Hijacking

会话劫持是指通过非常规手段，来得到合法用户在客户端和服务器段进行交互的特征值（一般为 sessionid），然后伪造请求，去访问授权用户的数据。

2、阅读下面这篇文章并且了解 Netwox/Netwag 的基本操作：

Netwox/Netwag Troubleshooting guide

http://www.cis.syr.edu/~wedu/seed/Documentation/Misc/netwox.pdf

已经阅读并了解 Netwox/Netwag 的基本操作

3、解释 linux 用 root 执行下面两条命令

sysctl -q net.ipv4.tcp_max_syn_backlog //表示 SYN 队列的长度

sysctl -w net.ipv4.tcp_syncookies=0 // 表示开启 SYN Cookies，当出现 SYN 等待队列溢出时，启用 cookies 来处理，可防范少量 SYN 攻击，默认为 0，表示关闭

4、阅读 Smashing C++ VPTRs 这篇文章，阐述其原理。

http://www.phrack.org/issues.html?issue=56&id=8

目前为止，缓冲区溢出都是针对 c 语言程序的，但是在越来越多的 c++程序中也会出现缓冲区溢出，是通用的。而且 c++的面向对象特性也导致了新的缓冲区溢出问题

C++的虚拟指针（Virtual Pointer,VPTR）

5、考虑 Linux 环境下如何实现精确计算程序运行的时间，搜索下相关资料。

在 Linux/Unix 环境下，计算 C 程序运行时间可以通过以下三个函数来实现：clock()、time()、gettimeofday()

clock()函数是 ANSI C 的标准库函数，是 C/C++ 十分常用的计时函数，其声明定义在 time.h 头文件中：

clock_t clock(void);

对于 clock()函数来说，需要注意以下几点：

1、它返回的是 CPU 耗费在本程序上的时间。也就是说，途中 sleep 的话，由于 CPU 资源被释放，那段时间将不被计算在内。因此，若函数中存在 sleep()函数，则 sleep()函数消耗的时间将不包含在内；

2、得到的返回值其实就是耗费在本程序上的 CPU 时间片的数量，也就是 Clock Tick 的值。该值必须除以 CLOCKS_PER_SEC 这个宏值，才能最后得到以秒为单位的运行时间。在 POSIX 兼容系统中，CLOCKS_PER_SEC 的值为 1,000,000 的，也就是 1MHz。

3、这个函数的精度不适很高，大约为 10ms，也可能是 1ms。低于精度的程序全部输出 0ms。像计算 printf()之类的调用时间是不可能实现的，因为 printf()的速度太快了，基本上和 clock()的速度一样。所以，此函数适合用于计算一些大型程序，或循环程序。

time()函数来获得当前日历时间（Calendar Time）。所谓的日历时间就是用“从一个标准时间点（一般是 1970 年 1 月 1 日 0 时 0 分 0 秒）到此时的时间经过的秒数”来表示的时间。其原型为：

time_t time(time_t * timer);

gettimeofday() 函数

gettimeofday()的函数原型为：

#include<sys/time.h>

int gettimeofday(struct timeval*tv, struct timezone *tz)