**SCANNER报告**

------161520322仲奕杰

1. **全局依赖**

#include<stdlib.h>

#include<stdio.h>

#include<sys/socket.h> //引入recvfrom、sendto等函数

#include<sys/time.h> //获取系统时间

#include<sys/types.h> //数据类型定义

#include<time.h>

#include<string.h>

#include<errno.h> //抛出错误信息

#include<unistd.h>

#include<arpa/inet.h> //提供IP地址转换函数

#include<linux/tcp.h> //提供通用的文件、目录、程序及进程操作的函数

#include<fcntl.h> //提供对文件控制的函数

#include<netdb.h> //提供设置及获取域名的函数

#include<signal.h> //进行信号处理

#include<netinet/in.h> //获取protocol参数的常值

#include<netinet/ip.h> //ip帧头定义

#include<netinet/ip\_icmp.h> //icmp帧头定义

#include<pthread.h> //提供多线程操作的函数

声明各种依赖的头文件

#define maxbuf 20480

#define sport 1 //默认扫描起始端口号

#define eport 1024 //默认扫描结束端口号

扫描局域网内存活主机，默认只扫描1-1024的常用端口

//global

struct scansock

**{**

char src\_ip**[**maxbuf**];** //源ip地址

char dst\_ip**[**maxbuf**];** //目标ip地址

int start\_port**;** //扫描起始端口

int end\_port**;** //扫描结束端口

**};**

struct scanbody

**{**

char src\_ip**[**maxbuf**];**

char dst\_ip**[**maxbuf**];**

int src\_port**;**

int dst\_port**;**

**};**

扫描所需要的自定义结构体，包括所需要的目标ip源ip起始结束端口等，传入扫描模块

//tcp伪头部

struct pseudohdr

**{**

struct in\_addr src\_ip**;**

struct in\_addr dst\_ip**;**

unsigned char zero**;**

unsigned char protocol**;** //协议类型

unsigned short length**;** //伪头部长度

//struct tcphdr tcpheader;

**};**

定义了tcp伪头部结构体

int pingflag**,** ipcount**=**0**;**

char dst\_ip**[**maxbuf**][**20**];**

char dst\_name**[**maxbuf**][**20**];**

//tcp\_connect\_scan\_port

struct list

**{**

int data**;**

struct list **\***next**;**

**};**//链表

用于存储扫描结果，每检测到一个开放端口，就新建立一个节点接在链表尾部。

1. **整体流程**

首先获取本机ip，随后通过arp指令获取已经存在的关于局域网内存活主机的信息，检测这些主机当前是否存活，对于存活的主机发起tcp的connect请求，如果成功说明端口开放，并在链表中记录。

1. **具体实现**
2. 获取本机ip

使用popen函数执行外部linux命令，并可以得到其返回值。system不能获取到系统返回信息。使用指令

/sbin/ifconfig | grep 'inet ' | grep -v '127' | awk -F ':' '{print $2}' | cut -d ' ' -f1，

得到本机ip并返回给主程序

1. 获取局域网内主机信息

通过arp指令能够得到当前已经获得到的局域网主机信息，主机名与ip的对应信息在/etc/hosts中存储。

使用指令

/usr/sbin/arp -a | awk -F ' ' '{print $1 $2}'

可以得到当前存活主机的字节流，通过void get\_dst(char dst\_ip[maxbuf][20],char dst\_name[maxbuf][20],int \*count)函数，可以得到当前存活的主机。

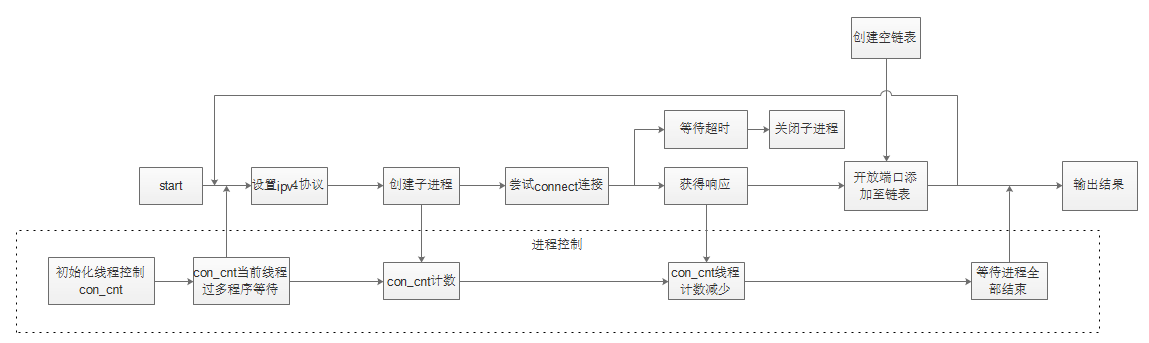
由于获取到的ip中可能有些已经不再存活所以需要对获取到的数据再做一次判断。

1. 检测主机是否存活

可以通过实验1中的ping来判断，也可以通过shell端的指令进行判断，准确率和速率都会有一定提高。通过指令

ping –c 1 [dst\_ip] | grep 'packets' | awk -F ',' '{print $3}' | cut -d ' ' -f2

的结果来判断主机是否为存活状态。

1. 多进程扫描模块

void\* tcp\_con\_scan\_sonthread(void \*argv)

子进程控制函数，使用socket内置connect函数尝试与目标主机所扫描的端口发起tcp连接请求。如果连接成功则将当前端口添加至结果链表中，如果连接不成功或因为防火墙导致无法连接，则不处理。函数最后释放使用的各变量空间，并且结束当前进程，进程计数减一并关闭connect连接。

void\* tcp\_con\_scan(void \*argv)

connnect扫描主函数，扫描所有给定端口，每个端口创建一个子进程，并且计算当前运行的进程数，进程数如果过多则在扫描下一个端口前等待一定时间使进程数减少至规定值以下。结束扫描前等待所有进程结束。