#include<stdio.h>

#include<time.h>

#include<stdlib.h>

#include <windows.h>

#include<string.h>

typedef struct node

{

int name;

int number;

int time;

int value;

int Attack[1600][4];

int important;

int hadcreate;

int shuangchong;

int ifdefense;

struct node \*on;

struct node \*next[20];

}Node;

int zaima(int a,int b,int c[]) //要查找的数，目前数组中已经有的数的数量，数组

{

int d=0,e=0;

for(d=0;d<b;d++)

if(c[d]==a)

{

e=1;

break;

}

if(e==1)

return 0; //等于0表示在里面了，等于1表示不在里面

else

return 1;

}

//为主机和路由器随机分配ip地址

//为主机和路由器随机分配ip地址

void dis(int a,int b,Node link\_a[],Node link\_b[])

{

int i,j;

int num[50]; // 装随机数的数组 50个整数

int ret=0; // 临时存放产生的一个随机数

int flag=1;// 定义标志位,flag=false代表数组中无重复数,true代表有重复数

int h=0,e=0;

int c=a+b;

srand((unsigned)time(NULL));

for(i=0;i<a;)

{

do{

ret=rand()%253+1;

for(h=0;h<50;h++)

{

if(ret==num[h])

{

flag=1;

break;

}

else{

flag=0;

}

}

}while(flag==1);

if(flag==0)

{

link\_a[i].number=ret;

num[i]=ret;

link\_a[i].name=i+1;

i++;

}

}

flag=1;

for(j=0;j<b;)

{

do

{

ret=rand()%253+1;

for(h=0;h<50;h++)

{

if(ret==num[h])

{

flag=1;

break;

}

else{

flag=0;

}

}

}while(flag==1);

if(flag==0)

{

link\_b[j].number=ret;

link\_b[j].name=a+j+1;

num[i+j]=ret;

j++;

}

}

for(c=0;c<20;c++)

{

for(e=0;e<20;e++)

link\_a[c].next[e]=NULL;

link\_a[c].on=NULL;

link\_a[c].ifdefense=0;

link\_a[c].hadcreate=0;

link\_a[c].important=0;

link\_a[c].time=5\*(rand()%100+20);

link\_a[c].shuangchong=0;

}

for(c=0;c<20;c++)

{

for(e=0;e<10;e++)

link\_b[c].next[e]=NULL;

link\_b[c].hadcreate=0;

link\_b[c].on=NULL;

link\_b[c].shuangchong=0;

link\_b[c].important=0;

link\_b[c].time=5\*(rand()%100+20);

}

for(c=0;c<20;c++)

for(i=0;i<1600;i++)

link\_a[c].Attack[i][0]=0;

}

void Show\_Interface(int a,int b,Node list\_a[],Node list\_b[])

{

int i,j;

HANDLE hdl = GetStdHandle(STD\_OUTPUT\_HANDLE);

if(a>=b)

{

printf("The host number \t The IP address \t The router\t The IP address\n");

for(i=0;i<b;i++)

{

printf(" %d \t",list\_a[i].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("\t\t 192.168.1.%d\t",list\_a[i].number);

SetConsoleTextAttribute(hdl, 7);

printf("\t %d\t",list\_b[i].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf(" \t 192.168.1.%d \n",list\_b[i].number);

SetConsoleTextAttribute(hdl, 7);

}

j=i;

for(j;j<a;j++)

{

printf(" %d \t",list\_a[j].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("\t\t 192.168.1.%d \n",list\_a[j].number);

SetConsoleTextAttribute(hdl, 7);

}

}

else

{

printf(" The router\t The IP address \t The host number \t The IP address\n");

for(i=0;i<a;i++)

{

printf(" %d \t",list\_b[i].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("\t 192.168.1.%d\t",list\_b[i].number);

SetConsoleTextAttribute(hdl, 7);

printf(" \t\t%d\t",list\_a[i].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf(" \t 192.168.1.%d \n",list\_a[i].number);

SetConsoleTextAttribute(hdl, 7);

}

j=i;

for(j;j<b;j++)

{

printf(" %d \t",list\_b[j].name);

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("\t 192.168.1.%d \n",list\_b[j].number);

SetConsoleTextAttribute(hdl, 7);

}

}

}

void printdot(int a,int b)

{

int i,m,n;

srand((unsigned)time(NULL));

if(a>=b)

m=rand()%a+b;

else

m=rand()%b+a;

for(i=0;i<m;i++)

{

n=rand()%(200+50)\*5;

Sleep(n);

printf(".");

}

printf("\n");

}

void Random\_number(int \*x,int \*y)

{

srand(time(NULL));

\*x=rand()%18+2;

\*y=rand()%19+1;

}

void Assignment(int a,int b,Node link\_a[],Node link\_b[])

{

int i,j;

for(i=0;i<a;i++)

{

link\_a[i].value=0;

}

for(j=0;j<b;j++)

{

link\_b[j].value=1;

}

printf("----------------------------------------------------------------------------------------------------------");

}

void routerconnect(int b,Node link\_b[])

{

int x,temp[20],e=0,x1,n,m=0;

srand(time(NULL));

if(b==2)

{

x=rand()%2;

temp[e++]=x;

do{ //一直到生成一个不在里面的数x1

x1=rand()%b;

}while(zaima(x1,e,temp)==0); //表示在里面

link\_b[x].next[link\_b[x].hadcreate]=&link\_b[x1];

link\_b[x].hadcreate++;

link\_b[x1].next[link\_b[x1].hadcreate]=&link\_b[x];

link\_b[x1].hadcreate++;

}

else if(b>2)

{

x=rand()%b+1;

temp[e++]=x;

do{ //一直到生成一个不在里面的数x1

x1=rand()%b+1;

}while(zaima(x1,e,temp)==0); //表示在里面

temp[e++]=x1;

link\_b[x-1].next[link\_b[x-1].hadcreate++]=&link\_b[x1-1];

link\_b[x1-1].next[link\_b[x1-1].hadcreate++]=&link\_b[x-1];

for(m=0;m<b-2;m++)

{

do{ //一直到生成一个不在里面的数x1

x1=rand()%b+1;

}while(zaima(x1,e,temp)==0); //表示在里面

n=rand()%e;

link\_b[temp[n]-1].next[link\_b[temp[n]-1].hadcreate++]=&link\_b[x1-1];

link\_b[x1-1].next[link\_b[x1-1].hadcreate++]=&link\_b[temp[n]-1];

temp[e++]=x1;

}

}

}

void hostconnect(int a,int b,Node link\_a[],Node link\_b[])

{

int i,x;

srand(time(NULL));

for(i=0;i<a;i++)

{

x=rand()%b;

link\_b[x].next[link\_b[x].hadcreate++]=&link\_a[i];

link\_a[i].next[link\_a[i].hadcreate++]=&link\_b[x];

}

}

void Show(int a,int b,Node link\_a[],Node link\_b[])

{

int i,j,t,k;

HANDLE hdl = GetStdHandle(STD\_OUTPUT\_HANDLE);

printf("\n");

printf("Network connection of host:\n");

for(i=0;i<a;i++)

{

printf("%d\t",i+1);

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d",link\_a[i].number);

SetConsoleTextAttribute(hdl, 7);

printf("->");

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("192.168.1.%d\n",link\_a[i].next[0]->number);

SetConsoleTextAttribute(hdl, 7);

}

printf("\n");

Sleep(1000);

printf("Network connection of router:\n");

for(i=0;i<b;i++)

{

printf("%d",i+a+1);

for(j=0;j<link\_b[i].hadcreate;j++)

{

for(t=0;t<a;t++)

{

if(link\_b[i].next[j]->number==link\_a[t].number)

{

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("\t192.168.1.%d",link\_b[i].number);

SetConsoleTextAttribute(hdl, 7);

printf("->");

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d\n",link\_b[i].next[j]->number);

SetConsoleTextAttribute(hdl, 7);

}

}

for(k=0;k<b;k++)

{

if(link\_b[i].next[j]->number==link\_b[k].number)

{

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("\t192.168.1.%d",link\_b[i].number);

SetConsoleTextAttribute(hdl, 7);

printf("->");

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("192.168.1.%d\n",link\_b[i].next[j]->number);

SetConsoleTextAttribute(hdl, 7);

}

}

}

printf("\n");

}

printf("\n");

}

void show\_main\_menu()

{

printf("\nPlease type the instruction:\n");

printf("1 ipconfig Enter the serial number to view the host or router information.\n2 ping \tEnter the IP addresses of the two hosts to view the ping process\n");

printf("3 newcreate Build a new host or router and connect to the Internet\n4 synattack Enter the IP address of attacker and target and start flooding attack\n");

printf("5 syndefense Type the IP address and specify the host to open the flooding attack defense system\n6 main menu Display command operation main interface\n");

printf("7 show View current Internet connection status\n8 exit End and close the program\n\n");

}

void ipconfig(int a,int b,Node link\_a[],Node link\_b[])

{

int i,m=1,ip,x;

HANDLE hdl = GetStdHandle(STD\_OUTPUT\_HANDLE);

printf("Please type in the IP address of the existing emulator, such as: '192.168.1. X'\n");

scanf("192.168.1.%d",&ip);

for(i=0;i<a;i++)

if(link\_a[i].number==ip)

{ m=0;

break;}

if(m==1)

for(i=0;i<b;i++)

if(link\_b[i].number==ip)

{ m=1;

break;}

printf("\n\n");

if(m==0)

{

printf("The host number ");

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d",ip);

SetConsoleTextAttribute(hdl, 7);

printf(" configuration:\n\n");

printf("Connected router IP address. . . . . . . . . . . . .:192.168.1.%d\n\n",link\_a[i].next[0]->number);

printf("Time to communicate with connection route. . . . . .:%dms\n\n",link\_a[i].time);

if(link\_a[i].ifdefense==1)

printf("Flood prevention state:. . . . . . . . . . . . . . .:OPEN\n");

else

printf("Flood prevention state:. . . . . . . . . . . . . . .:CLOSE\n");

}

if(m==1)

{

printf("The router number ");

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("192.168.1.%d",ip);

SetConsoleTextAttribute(hdl, 7);

printf(" configuration:\n\n");

printf("Connected router IP address. . . . . . . . . . . . .:192.168.1.%d\n",link\_b[i].next[0]->number);

for(x=1;x<link\_b[i].hadcreate;x++)

printf(" 192.168.1.%d\n",link\_b[i].next[x]->number);

printf("\nTime to communicate with connection node. . . . . . :%dms\n\n",link\_b[i].time);

printf("Number of connected nodes . . . . . . . . . . . . . :%d\n\n",link\_b[i].hadcreate);

}

}

void findway(int a,int b,Node link\_a[],Node link\_b[],int m,int n)

{

int i,j,k;

int timesum=0;

int TTL=12;

HANDLE hdl = GetStdHandle(STD\_OUTPUT\_HANDLE);

Node \*p;

if(link\_a[a].next[0]->number==link\_a[b].next[0]->number)

{

link\_a[a].shuangchong=1;

link\_a[a].next[0]->shuangchong=1;

link\_a[b].shuangchong=1;

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d->",link\_a[a].number,link\_a[b].number);

SetConsoleTextAttribute(hdl, 7);

Sleep(link\_a[a].time);

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("192.168.1.%d->",link\_a[a].next[0]->number);

SetConsoleTextAttribute(hdl, 7);

Sleep(link\_a[a].next[0]->time);

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d\n",link\_a[b].number);

SetConsoleTextAttribute(hdl, 7);

timesum=link\_a[a].time+link\_a[a].next[0]->time;

TTL--;

printf("Packet transfer time：%dms TTL------%d\n",timesum,TTL);

}

else

{

p=link\_a[b].next[0];

(\*p).shuangchong=1;

link\_a[a].next[0]->important=1;

link\_a[a].shuangchong=1;

link\_a[a].next[0]->on=&link\_a[a];

link\_a[b].on=link\_a[b].next[0];

link\_a[b].shuangchong=1;

for(k=0;k<n;k++)

for(i=0;i<n;i++)

for(j=0;j<link\_b[i].hadcreate;j++)

if(link\_b[i].next[j]->important==1&&link\_b[i].important!=1)

{

link\_b[i].on=link\_b[i].next[j];

link\_b[i].important=1;

}

do{

(\*p).on->shuangchong=1;

p=(\*p).on;

}while((\*p).number!=link\_a[a].next[0]->number);

p=link\_a[a].next[0];

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d->",link\_a[a].number);

SetConsoleTextAttribute(hdl, 7);

Sleep(link\_a[a].time);

for(;(\*p).number!=link\_a[b].number;)

{

SetConsoleTextAttribute(hdl, FOREGROUND\_RED);

printf("192.168.1.%d->",(\*p).number);

SetConsoleTextAttribute(hdl, 7);

Sleep((\*p).time);

for(j=0;j<(\*p).hadcreate;j++)

if((\*p).next[j]->shuangchong==1&&(\*p).next[j]!=(\*p).on)

{

p=(\*p).next[j];

break;

}

}

SetConsoleTextAttribute(hdl, FOREGROUND\_GREEN);

printf("192.168.1.%d\n",link\_a[b].number);

SetConsoleTextAttribute(hdl, 7);

for(i=0;i<n;i++)

if(link\_b[i].shuangchong==1)

{

timesum+=link\_b[i].time;

TTL--;

}

printf("Packet transfer time：%dms TTL------%d\n",timesum,TTL);

}

}

void guiling(int a,int b,Node link\_a[],Node link\_b[])

{

int i=0;

for(i=0;i<a;i++)

{

link\_a[i].important=0;

link\_a[i].on=NULL;

link\_a[i].shuangchong=0;

}

for(i=0;i<b;i++)

{

link\_b[i].important=0;

link\_b[i].on=NULL;

link\_b[i].shuangchong=0;

}

}

void newcreate(int x,int p,Node link\_a[],Node link\_b[],int a,int b)

{

int i,h,j,number;

srand((unsigned)time(NULL));

for(i=0;i<b;i++)

if(link\_b[i].name==p)

break;

if(x==0)

{

link\_a[a].hadcreate=1;

link\_a[a].next[0]=&link\_b[i];

link\_b[i].next[link\_b[i].hadcreate++]=&link\_a[a];

link\_a[a].name=a+1;

for(i=0;i<b;i++)

link\_b[i].name++;

while(1) //随机生成新的.ip好重新写成一个函数，方便调用

{

srand(time(NULL));

x=rand()%254+1;

for(h=0;h<a;h++)

{

if(x!=link\_a[h].number)

continue;

else

x=rand()%254+1;

}

number=x;

for(j=0;j<b;j++)

{

if(number!=link\_b[j].number)

continue;

else

number=rand()%254+1;

}

break;

}

link\_a[a].number=number;

}

else

{

link\_b[b].hadcreate=1;

link\_b[b].next[0]=&link\_b[i];

link\_b[i].next[link\_b[i].hadcreate++]=&link\_b[b];

link\_b[b].name=a+b+1;

while(1) //随机生成新的.ip好重新写成一个函数，方便调用

{

srand(time(NULL));

x=rand()%254+1;

for(h=0;h<a;h++)

{

if(x!=link\_a[h].number)

continue;

else

x=rand()%254+1;

}

number=x;

for(j=0;j<b;j++)

{

if(number!=link\_b[j].number)

continue;

else

number=rand()%254+1;

}

break;

}

link\_b[b].number=number;

}

}

int gailv(int b)

{

int a;

a=rand()%b;

if(a==0)

return 1;

else

return 0;

}

void attack(int p,int q,Node link\_a[],Node link\_b[],int a,int b,int e)

{

int i=0,w=0,yifadao[100],time=0,j,m,timesum=0,muditime=0;

int shuliang,sumtime;

int yihuifu=0,weihuifu=0,biaozhun1,biaozhun2,biaozhun3;

Node \*u;

biaozhun1=rand()%200+400;

biaozhun2=rand()%100+600;

biaozhun3=rand()%50+700;

findway(p,q,link\_a,link\_b,a,b);

printf("Please input the contract quantity:\n");

scanf("%d",&shuliang);

u=link\_a[p].next[0];

sumtime=link\_a[p].time;

for(i=0;i<b;i++)

if(link\_b[i].shuangchong==1)

{ //w表示主干路节点个数

w++;

sumtime+=link\_b[i].time;

}

timesum=link\_a[p].time;

for(i=0;i<w;i++)

yifadao[i]=0;

for(i=0;i<shuliang+sumtime/100;i++)

{

if(i<shuliang)

{

u=link\_a[p].next[0];

link\_a[p].Attack[i][0]=rand()%254+1;

link\_a[p].Attack[i][1]=rand()%254+1;

link\_a[p].Attack[i][2]=rand()%254+1;

link\_a[p].Attack[i][3]=rand()%254+1;

printf("\t%d.%d.%d.%d已生成\t",link\_a[p].Attack[i][0],link\_a[p].Attack[i][1],link\_a[p].Attack[i][2],link\_a[p].Attack[i][3]);

}

else

printf("\t\t空闲\t\t");

time+=100;

timesum=link\_a[p].time;

for(j=0;j<w;j++)

{

if(timesum<time&&link\_a[p].Attack[yifadao[j]][0]!=0)

{

printf("\t%d.%d.%d.%d已转发\t",link\_a[p].Attack[yifadao[j]][0],link\_a[p].Attack[yifadao[j]][1],link\_a[p].Attack[yifadao[j]][2],link\_a[p].Attack[yifadao[j]][3]);

yifadao[j]++;

}

else

printf("\t\t空闲\t\t");

for(m=0;m<u->hadcreate;m++)

if(u->next[m]->shuangchong==1)

break;

timesum+=u->time;

u=(\*u).next[m];

}

if(timesum<time)

if(gailv(3)==1)

yihuifu++;

else

weihuifu++;

if(link\_a[q].ifdefense==1&&gailv(2)==0&&yihuifu>0)

{

weihuifu--;

yihuifu++;

}

printf("靶机已回复%d\t未回复%d \t",yihuifu,weihuifu);

if(weihuifu<biaozhun1)

printf("流畅");

else if(biaozhun1<weihuifu&&weihuifu<biaozhun2||weihuifu==biaozhun1||weihuifu==biaozhun2)

printf("卡顿");

else if(biaozhun2<weihuifu&&weihuifu<biaozhun3||weihuifu==biaozhun2||weihuifu==biaozhun3)

printf("严重卡顿");

else if(weihuifu>biaozhun3)

{

printf("瘫痪\n");

break;

}

Sleep(e\*5);

printf("\n");

}

}

int main()

{

int a,b,m,n,p,q,w=0,e;

Node link\_a[20],link\_b[20];

char c[20],d[10];

HANDLE hdl = GetStdHandle(STD\_OUTPUT\_HANDLE);

printf("Environment deployment in progress"); //环境部署正在进行

printdot(5,1);

printf("----------------------------------------------------------\n");

printf("Random hosts and random routers are being generated"); //随机主机数与路由器数正在生成

printdot(5,1);

Random\_number(&a,&b);

printf("Finished: %d random hosts and %d random routers were generated\n",a,b); //已生成%d个随机主机与%d个随机路由器

printf("Please wait for the host and router to assign IP address");

printdot(6,2);

dis(a,b,link\_a,link\_b); //分配IP地址并且完成路由器随机链接

printf("Finished:Host and router have assigned IP address!\n"); //主机与路由器已分配ip地址！

printf("----------------------------------------------------------\n");

Show\_Interface(a,b,link\_a,link\_b);

Assignment(a,b,link\_a,link\_b);

routerconnect(b,link\_b);//完成主机随机连接路由器

hostconnect(a,b,link\_a,link\_b);

Sleep(1000);

show\_main\_menu();

do

{

gets(c);

if(strcmp(c,"ipconfig")==0)

{

ipconfig(a,b,link\_a,link\_b);

printf("\n");

}

else if(strcmp(c,"ping")==0)

{

printf("Type the IP addresses of both parties, such as '192.168.1. Xping192.168.1. X'\n");

scanf("192.168.1.%dping192.168.1.%d",&m,&n);

for(p=0;p<20;p++)

if(link\_a[p].number==m)

break;

for(q=0;q<20;q++)

if(link\_a[q].number==n)

break;

findway(p,q,link\_a,link\_b,a,b);

guiling(a,b,link\_a,link\_b);

printf("\n\n");

}

else if(strcmp(c,"newcreate")==0)

{

printf("Please enter host or router and select the serial number of the router to connect to\n");

scanf("%s%d",d,&p);

if(strcmp(d,"host")==0)

{

newcreate(0,p,link\_a,link\_b,a,b);

a++;}

else

{

newcreate(1,p,link\_a,link\_b,a,b);

b++;}

printf("Establishment completed\n\n");

}

else if(strcmp(c,"synattack")==0)

{

printf("Select the host you want to attack\nAnd the extent of data analysis interval\n0 will print the whole paper, and the larger the value, the slower the analysis\nsuch as '192.168.1. Xattack192.168.1. X 10'\n");

scanf("192.168.1.%dattack192.168.1.%d%d",&m,&n,&e);

for(p=0;p<20;p++)

if(link\_a[p].number==m)

break;

for(q=0;q<20;q++)

if(link\_a[q].number==n)

break;

attack(p,q,link\_a,link\_b,a,b,e);

guiling(a,b,link\_a,link\_b);

printf("The attack is over\n\n");

}

else if(strcmp(c,"syndefense")==0)

{

printf("Please specify the host to enable flood prevention,such as '192.168.1. X'\n");

SetConsoleTextAttribute(hdl, FOREGROUND\_BLUE);

scanf("192.168.1.%d",&m);

SetConsoleTextAttribute(hdl, 7);

for(p=0;p<20;p++)

if(link\_a[p].number==m)

break;

link\_a[p].ifdefense=1;

printf("The specified host flooding prevention system is turned on\n\n");

}

else if(strcmp(c,"main menu")==0)

{

show\_main\_menu();

}

else if(strcmp(c,"show")==0)

{

Show(a,b,link\_a,link\_b);

}

else if(strcmp(c,"exit")==0)

exit(1);

}while(1);

return 0;

}