实验四

汤添凝 1170300728

实验目的

通过本实验了解 Wireshark 进行被动数据包捕获后的文件还原功能。

辅助工具

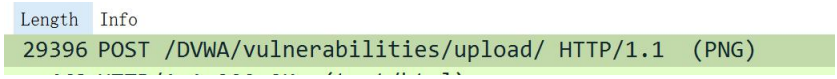
Wireshark，十六进制编辑器

实验目标

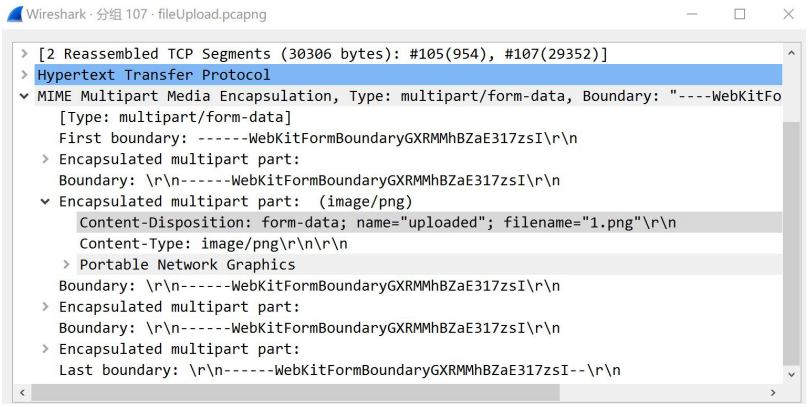
通过 wireshark 还原用户向网站上传的文件。对抓到的包进行显示过滤，找到关键信息。 对信息进行跟踪，确定上传文件的 TCP 流，并保存为二进制原始文件。 对文件中上传文件的信息进行处理，去掉多余的包头和包尾，得到原始文件。

实验步骤

1、使用 wireshark 导入监听数据包，对数据进行显示过滤，提取出来关键信息。
（1）用 wireshark 打开 fileUpload.pcapng。会发现多条数据记录。
（2）利用 Wireshark 提供的过滤显示功能。在 filter 中可以定义显示的数据包类型。此处上传时访问的是网站，因此在 filter 中输入_http_进行协议过滤。
（3）上传文件提交可以使用 post 一个表单的形式，所以可以利用包过滤显示，选出所有使用 post 方法提交的数据包。在一条数据记录中的 info 中看到_upload_这个词，这条可能就是涉及到上传的数据包，截图如下：



2.确定 POST 这条数据包是否上传了文件，若存在则将数据 dump 出来。
（1）双击该条记录。弹出协议分析框。点击+号，将子栏展开。可以看到，上传的文件名是_1.png__，上传的是一张图片。截图如下：



（2）可以看到由于文件比较大，TCP 协议对其进行了切片，一共切了 2 个片。给出实验截图：

```
[2 Reassembled TCP Segments (30306 bytes): #105(954), #107(29352)]
    [Frame: 105, payload: 0-953 (954 bytes)]
    [Frame: 107, payload: 954-30305 (29352 bytes)]
    [Segment count: 2]
    [Reassembled TCP length: 30306]
    [Reassembled TCP Data: 504f5354202f445657412f76756c6e65726162696c697469657
```

（3）将这几个切片还原成一个流式会话。右键 POST 包，点击 Follow TCP Stream 这时候我们会看到整个会话都被还原了出来。能够得到文件的原始信息。继续往下拉，会看到有关蓝色的显示，这是服务器给的回应。文件信息保存在请求部分，因此可以过滤掉响应部分。选择请求部分（更大的那个数据包），选择以 raw 类型显示，保存为任意格式的文件。

3.使用十六进制文件编辑器对文件进行最终处理，并保存文件。

（1）将刚才保存的文件用十六进制编辑器打开。会看到文中包含请求信息和文件信息，以及文件结尾的尾部信息。 对照 wireshark 中刚才的 tcp stream 流，确定图片文件的原始信息头和尾，去掉多余部分。可以看到原始信息头部结尾的四个字节为 0d0a0d0a ，给出实验截图。

```
0D 0A 0D 0A  _filename="1.png"..Content-Type: image/png....
```

原始信息尾部以换行和“------”开始，后者的十六进制为 2D2D2D2D2D2D ,给出实验截图。

```
000066C0  4E 20 20 4C 49 42 77 3F  02 20 10 8C 50 03 20 58  N  LIBw?   ŒP  X
000066D0  23 6C 3F 20 20 89 88 3F  20 3F 3F 20 3F 3F 20 3F  #l?  ‰ˆ? ?? ?? ?
000066E0  FF 0F 72 C8 99 74 2B 38  3F 20 20 20 20 49 45 4E  ÿ rÈ™t+8?    IEN
000066F0  44 AE 42 60 3F 0A 2D 2D  2D 2D 2D 2D 57 65 62 4B  D®B`?------WebK
00006700  69 74 46 6F 72 6D 42 6F  75 6E 64 61 72 79 47 58  itFormBoundaryGX
00006710  52 4D 4D 68 42 5A 61 45  33 31 37 7A 73 49 0D 0A  RMMhBZaE317zsI
```

（2）delete 去掉多余首位，得到原始图片内容（注：如出现系统找不到指定路径的提示，可以按照提示创建指定文件夹路径），Ctrl+S 保存。

（3）将文件后缀改为.png。打开可见原始图片，图片内容如下：

```
void url_decode(char *dst, const char *src)
{
    for (;;)
    {
        if (src[0] == '%' && src[1] && src[2])
        {
            char hexbuf[3];
            hexbuf[0] = src[1];
            hexbuf[1] = src[2];
            hexbuf[2] = '\0';

            *dst = strtol(&hexbuf[0], 0, 16);
            src += 3;
        }
        else if (src[0] == '+')
        {
            *dst = ' ';
            src++;
        }
        else
        {
            *dst = *src;
            src++;

            if (*dst == '\0')
                break;
        }

        dst++;
    }
```