Wireshark之文件还原

## 实验简介

* 实验所属系列： 数据安全
* 实验对象： 本科/专科信息安全专业
* 相关课程及专业： 计算机网络
* 实验时数（学分）：2学时
* 实验类别： 实践实验类

## 实验目的

通过该实验了解wireshark的使用方法，能够通过分析还原网络数据发送现场，并将发送的信息通过wireshark和winhex还原成原文件。

## 预备知识

关于wireshark的一些基础知识

首先，wireshark不会像burp suite那样可以对数据包进行修改，它能做的只是监听网络流量信息并完整的记录下来。可以说，wireshark起到的是一个还原现场的作用。请记住，wireshark只是一个帮助你审计的工具，至于怎么去审计，只能考自己去分析。

其次，说说wireshark在安全方面的作用。Wiki上是这样描述的：对于安全人员而言，网络安全工程师可以使用wireshark来检查讯息安全相关的问题。对于网络安全而言，现场很重要，黑客攻击的过程和服务器被攻击的过程都必须使用网络，wireshark完全可以记录下来，这也正是wireshark强大的地方。

对本节课使用到的winhex的简单描述

Winhex是一款非常优秀的16进制编辑器，事实上，使用winhex单纯进行16进制编译在某些程度上有些大材小用，市场上很难找到像winhex一样功能强大的16进制编辑器。当然，它是收费的。本节课我们使用的是winhex的评估版本，可以满足需要。如果你想对winhex有更深层次的了解，建议您去官方网站上进行咨询。

## 实验环境



服务器：Windows XP SP3 ，IP地址：随机分配

辅助工具：winhex，Wireshark

## 实验步骤

黑客A通过ARP欺骗，使用wireshark获取了整个局域网内的网络流量信息。无意之中，他发现有人在某个网站上上传了一份文件。但是他不知道怎么样通过wireshark去还原这份文件，没办法，他将监听到的数据包保存为了一份wireshark的监听记录，打算去向你请教。你能帮助他找到那份上传的文件吗？

我们的任务分为3个部分：

1. 对抓到的包进行显示过滤，找到关键信息。
2. 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。
3. 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。

### 实验任务一

任务描述：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。

打开catchme.pcapng，双击即可。会发现数据记录一共有148条。如果单纯的从开始到结尾去一条一条的审计，是非常费力的事情。而且实际操作过程中，148条记录，已经算是很少的了。

 

好在wireshark为我们提供了强大的过滤显示功能。我们在filter中可以定义显示出来什么样的数据包。



从题目我们可以明确，上传时访问的是个网站，因此我们需要进行协议过滤。在filter中输入http，表示我们要显示所有使用http协议的数据包。输入回车，或者点击旁边的APPLY按钮，就可以进行显示过滤。



从图上下方我们可以看到，数据包由原来的148个变成了32个。这样就很容易帮我们分析了。

仔细分析，我们会在末尾左右的第143条数据记录中的info中看到upload这个词，我们怀疑这条就是涉及到上传的数据包。



如果你在此之前有些编写网站的经验，就会知道上传文件提交可以使用post一个表单的形式。所以，你也可以使用包过滤显示，选出所有使用post方法提交的数据包。我们可以输入http.request.method==”POST”进行包过滤。这时候的显示如下：



看到了吧，这时候只显示了唯一一条记录，就是我们刚才找到的序号为143的记录，是不是快了很多啊。因此，掌握数据包过滤，是熟练掌握wireshark的必备技能之一。

#### 练习

关于wireshark过滤功能，以下说法错误的是？【单选题】

【A】wireshark使用显示过滤，我们看到的数据包数目会变小，因此意味着我们的数据包文件也会跟着变小。

【B】wireshark的过滤功能不局限于显示过滤，它也可以做到在抓包的时候做到抓包过滤。

【C】在filter中填写过滤语法时，如果错误，会显示红色

【D】抓包过滤和显示过滤使用的不是相同的语法规则

答案：A

### 实验任务二

任务描述：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。

虽然我们看到了有upload关键字，有post方法，但是我们不能确定是不是真的就是上传文件的那个请求。我们来分析一下。双击该行。弹出协议分析框。点击+号，将子栏展开。



我们可以看到，确实是上传了文件，而且文件名是bingo.png.原来他上传的是一张图片。在上方红色部分，我们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5个片。我们点击下方的各个Frame，就可以看到每个包中的内容。

问题来了，能不能将这几个切片还原成一个流式会话，这样我们就能看到一个会话过程，而不是需要一个一个的去点击。

Wireshark还真可以做到。

关闭这个界面，回到我们过滤后的那个POST包，右键Follow TCP Stream



这时候我们会看到：



整个会话都被还原了出来。我们看到了png的原始信息。继续往下拉，我们会看到有关蓝色的显示，这是服务器给我们的回应。我们的图片信息保存在请求部分，因此可以过滤掉响应部分。



因为文件肯定比响应大，所以我们选择6010那个。这时候就没有响应部分出现了。

保存原始文件，以便下一步处理。我们已经知道，请求部分中包含了文件的原始信息。因此，我们可以先保存下来，然后处理一下，得到原始文件。

我们选择raw类型进行保存，表示使用二进制形式保存文件。



保存为任意格式的文件，这里我们保存为temp.bin

#### 练习

以下说法不正确的是：【单选题】

【A】我们之所以能抓到原始数据包，是因为POST的时候会将文件上传给服务器

【B】一般的文件头都会显示出来该文件的类型，我们可以通过这个方法去判断文件是什么类型

【C】 将我们保存的raw文件使用记事本打开，删除一些部分，其余部分不会发生改变。

【D】 follow tcp stream只会显示该选中数据包中的请求和返回信息，不会返回整个数据包中的会话操作。

答案：C

### 实验任务三

任务描述：

使用winhex对文件进行最终处理，并保存文件。

将刚才保存的temp.bin用winhex打开。

会看到，文中包含请求信息和我们的图片信息，以及文件结尾的尾部信息。我们需要做的事情是确定图片文件的原始信息头和尾，去掉多余部分。





回到wireshark中，会看到我们刚才的tcp stream流中，关于图片的头部分



在content-type: image/x-png后面有两个换行符，然后开始我们的原始文件。换行符用十六进制表示是 0D 0A.因为有两个，所以，我们在图片附近寻找0D 0A 0D 0A.后面的部分就表示图片的开始。

回到winhex中，我们找到了上述数字



这时候我们需要去掉图片以上的部分。在00000000偏移处点击alt+1，表示选块开始。



在我们找到的0D 0A 0D 0A处的最后一个0A处点击alt+2.表示选块结束。这时候，我们就选中了图片之前的多余部分。



按下delete键，选择yes。



这时候文件中的多余头部已经被删除



4.回到wireshark中，我们看看图片传送完毕之后的尾部部分。



我们可以看到，这次是一个换行符。后面有些文件结束标志-------------，我们同样删除它们。



这时候我们的文件中就仅仅是原始图片的内容了。Ctrl+S保存。



最激动人心的一步来了。将我们的temp.bin改为temp.png.打开看下：



祝贺你，已经完成了我们本节课的实验，拿下神秘的key。

#### 练习

以下说法正确的是：【单选题】

【A】在我们刚打开的时候，文件的内容是这样的：请求头部分+图片原始信息部分+文件传输完确认部分。

【B】在上一步使用wireshark保存文件的时候，保存为ascii形式，按照任务3的操作，也能得到这幅图片。

【C】图片信息开始前有一个换行符

【D】图片信息结束后有两个换行符。

正确答案：A。

## 实验报告要求

参考实验原理与相关介绍，完成实验任务，并对实验结果进行分析，完成思考题目，总结实验的心得体会，并提出实验的改进意见。

## 分析与思考

1）用户在登陆的时候很多网站都不会选择加密处理，因此一旦被窃听，你的账户和密码信息将暴露无遗。假如你是一个网站开发者，你会采取什么措施去解决这个问题。

2）wireshark除了分析数据包，还会对数据情况进行整体分析与处理，您可以在线下进行尝试。

## 配套学习资源

密码分析工具 – Pipal

<http://www.freebuf.com/tools/49592.html>